



## **Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point**

### *Disclaimer*

---

The present document has been produced and approved by the Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.  
It does not necessarily represent the views of the entire ETSI membership.

---

**Reference**

RGS/NFV-SOL005ed431

---

**Keywords**

API, NFV, protocol

**ETSI**

---

650 Route des Lucioles  
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B  
Association à but non lucratif enregistrée à la  
Sous-Préfecture de Grasse (06) N° w061004871

---

**Important notice**

The present document can be downloaded from:

<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at [www.etsi.org/deliver](http://www.etsi.org/deliver).

Users of the present document should be aware that the document may be subject to revision or change of status.

Information on the current status of this and other ETSI documents is available at

<https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:

<https://portal.etsi.org/People/CommitteeSupportStaff.aspx>

If you find a security vulnerability in the present document, please report it through our  
Coordinated Vulnerability Disclosure Program:

<https://www.etsi.org/standards/coordinated-vulnerability-disclosure>

---

**Notice of disclaimer & limitation of liability**

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

---

**Copyright Notification**

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2022.  
All rights reserved.

# Contents

Intellectual Property Rights .....	24
Foreword.....	24
Modal verbs terminology.....	24
1 Scope .....	25
2 References .....	25
2.1 Normative references .....	25
2.2 Informative references.....	27
3 Definition of terms, symbols and abbreviations.....	28
3.1 Terms.....	28
3.2 Symbols.....	28
3.3 Abbreviations .....	28
4 General Aspects.....	30
4.1 Overview .....	30
4.2 Void.....	30
4.3 Void.....	30
4.4 Common data types .....	31
4.4.1 Structured data types.....	31
4.4.1.1 Introduction.....	31
4.4.1.2 Void.....	31
4.4.1.3 Void.....	31
4.4.1.3a Void.....	31
4.4.1.4 Void.....	31
4.4.1.5 Type: NsInstanceSubscriptionFilter .....	31
4.4.1.6 Type: ResourceHandle .....	31
4.4.1.7 Type: AdditionalResourceInfo .....	32
4.4.2 Simple data types and enumerations .....	33
4.4.2.1 Introduction.....	33
4.4.2.2 Simple data types .....	33
4.4.2.3 Enumerations .....	33
4.5 HTTP conditional requests .....	33
4.6 Void.....	34
4.7 Void.....	34
5 NSD Management interface .....	34
5.1 Description .....	34
5.1a API version.....	35
5.2 Resource structure and methods .....	35
5.3 Sequence diagrams (informative).....	38
5.3.1 Flow of the creation of an individual NS descriptor resource.....	38
5.3.2 Flow of the uploading of NSD archive content .....	39
5.3.3 Flow of the fetching of an onboarded NSD archive .....	40
5.3.4 Flow of the update of an individual NS descriptor resource.....	41
5.3.5 Flow of the deletion of an individual NS descriptor resource.....	42
5.3.6 Flow of the querying/reading of NS descriptor information.....	43
5.3.6a Flow of reading the NSD .....	44
5.3.6b Flow of fetching the manifest file of an onboarded NSD archive.....	44
5.3.6c Flow of fetching an NSD archive artifact .....	45
5.3.7 Flow of the creation of an individual PNF descriptor resource .....	46
5.3.8 Flow of the uploading of PNFD archive .....	46
5.3.9 Flow of the fetching of an onboarded PNFD archive .....	47
5.3.9a Flow of the update of an individual PNF descriptor resource.....	48
5.3.10 Flow of the deletion of an individual PNF descriptor resource .....	49
5.3.11 Flow of the querying/reading of PNF descriptor information.....	50
5.3.11a Flow of reading the PNFD.....	51
5.3.11b Flow of fetching the manifest file of an onboarded PNFD archive .....	51

5.3.11c	Flow of fetching a PNFD archive artifact.....	52
5.3.12	Flow of managing subscriptions.....	52
5.3.13	Flow of sending notifications.....	54
5.4	Resources.....	55
5.4.1	Introduction.....	55
5.4.1a	Resource: API versions.....	55
5.4.2	Resource: NS Descriptors.....	55
5.4.2.1	Description.....	55
5.4.2.2	Resource definition.....	55
5.4.2.3	Resource methods.....	56
5.4.2.3.1	POST.....	56
5.4.2.3.2	GET.....	56
5.4.2.3.3	PUT.....	57
5.4.2.3.4	PATCH.....	58
5.4.2.3.5	DELETE.....	58
5.4.3	Resource: Individual NS Descriptor.....	58
5.4.3.1	Description.....	58
5.4.3.2	Resource definition.....	58
5.4.3.3	Resource methods.....	58
5.4.3.3.1	POST.....	58
5.4.3.3.2	GET.....	58
5.4.3.3.3	PUT.....	59
5.4.3.3.4	PATCH.....	59
5.4.3.3.5	DELETE.....	60
5.4.4	Resource: NSD archive content.....	61
5.4.4.1	Description.....	61
5.4.4.2	Resource definition.....	61
5.4.4.3	Resource methods.....	62
5.4.4.3.1	POST.....	62
5.4.4.3.2	GET.....	62
5.4.4.3.3	PUT.....	63
5.4.4.3.4	PATCH.....	64
5.4.4.3.5	DELETE.....	64
5.4.4a	Resource: NSD.....	64
5.4.4a.1	Description.....	64
5.4.4a.2	Resource definition.....	64
5.4.4a.3	Resource methods.....	65
5.4.4a.3.1	POST.....	65
5.4.4a.3.2	GET.....	65
5.4.4a.3.3	PUT.....	66
5.4.4a.3.4	PATCH.....	67
5.4.4a.3.5	DELETE.....	67
5.4.4b	Resource: NSD archive manifest.....	67
5.4.4b.1	Description.....	67
5.4.4b.2	Resource definition.....	67
5.4.4b.3	Resource methods.....	67
5.4.4b.3.1	POST.....	67
5.4.4b.3.2	GET.....	67
5.4.4b.3.3	PUT.....	69
5.4.4b.3.4	PATCH.....	69
5.4.4b.3.5	DELETE.....	69
5.4.4c	Resource: Individual NSD archive artifact.....	69
5.4.4c.1	Description.....	69
5.4.4c.2	Resource definition.....	69
5.4.4c.3	Resource methods.....	69
5.4.4c.3.1	POST.....	69
5.4.4c.3.2	GET.....	70
5.4.4c.3.3	PUT.....	71
5.4.4c.3.4	PATCH.....	71
5.4.4c.3.5	DELETE.....	71
5.4.5	Resource: PNF Descriptors.....	72
5.4.5.1	Description.....	72

5.4.5.2	Resource definition .....	72
5.4.5.3	Resource methods .....	72
5.4.5.3.1	POST .....	72
5.4.5.3.2	GET .....	73
5.4.5.3.3	PUT .....	74
5.4.5.3.4	PATCH .....	74
5.4.5.3.5	DELETE .....	74
5.4.6	Resource: Individual PNF Descriptor .....	75
5.4.6.1	Description .....	75
5.4.6.2	Resource definition .....	75
5.4.6.3	Resource methods .....	75
5.4.6.3.1	POST .....	75
5.4.6.3.2	GET .....	75
5.4.6.3.3	PUT .....	76
5.4.6.3.4	PATCH .....	76
5.4.6.3.5	DELETE .....	76
5.4.7	Resource: PNFD archive content .....	77
5.4.7.1	Description .....	77
5.4.7.2	Resource definition .....	77
5.4.7.3	Resource methods .....	77
5.4.7.3.1	POST .....	77
5.4.7.3.2	GET .....	78
5.4.7.3.3	PUT .....	79
5.4.7.3.4	PATCH .....	80
5.4.7.3.5	DELETE .....	80
5.4.7a	Resource: PNFD .....	80
5.4.7a.1	Description .....	80
5.4.7a.2	Resource definition .....	80
5.4.7a.3	Resource methods .....	80
5.4.7a.3.1	POST .....	80
5.4.7a.3.2	GET .....	80
5.4.7a.3.3	PUT .....	82
5.4.7a.3.4	PATCH .....	82
5.4.7a.3.5	DELETE .....	82
5.4.7b	Resource: PNFD archive manifest .....	82
5.4.7b.1	Description .....	82
5.4.7b.2	Resource definition .....	83
5.4.7b.3	Resource methods .....	83
5.4.7b.3.1	POST .....	83
5.4.7b.3.2	GET .....	83
5.4.7b.3.3	PUT .....	84
5.4.7b.3.4	PATCH .....	84
5.4.7b.3.5	DELETE .....	84
5.4.7c	Resource: Individual PNFD archive artifact .....	85
5.4.7c.1	Description .....	85
5.4.7c.2	Resource definition .....	85
5.4.7c.3	Resource methods .....	85
5.4.7c.3.1	POST .....	85
5.4.7c.3.2	GET .....	85
5.4.7c.3.3	PUT .....	87
5.4.7c.3.4	PATCH .....	87
5.4.7c.3.5	DELETE .....	87
5.4.8	Resource: Subscriptions .....	87
5.4.8.1	Description .....	87
5.4.8.2	Resource definition .....	87
5.4.8.3	Resource methods .....	88
5.4.8.3.1	POST .....	88
5.4.8.3.2	GET .....	89
5.4.8.3.3	PUT .....	90
5.4.8.3.4	PATCH .....	90
5.4.8.3.5	DELETE .....	90
5.4.9	Resource: Individual subscription .....	91

5.4.9.1	Description .....	91
5.4.9.2	Resource definition .....	91
5.4.9.3	Resource methods .....	91
5.4.9.3.1	POST .....	91
5.4.9.3.2	GET .....	91
5.4.9.3.3	PUT .....	92
5.4.9.3.4	PATCH .....	92
5.4.9.3.5	DELETE .....	92
5.4.10	Resource: Notification endpoint .....	92
5.4.10.1	Description .....	92
5.4.10.2	Resource definition .....	92
5.4.10.3	Resource methods .....	93
5.4.10.3.1	POST .....	93
5.4.10.3.2	GET .....	93
5.4.10.3.3	PUT .....	94
5.4.10.3.4	PATCH .....	94
5.4.10.3.5	DELETE .....	94
5.5	Data model .....	94
5.5.1	Introduction .....	94
5.5.2	Resource and notification data types .....	94
5.5.2.1	Type: NsdInfoModifications .....	94
5.5.2.2	Type: NsdInfo .....	95
5.5.2.3	Type: CreateNsdInfoRequest .....	96
5.5.2.4	Type: PnfdInfoModifications .....	97
5.5.2.5	Type: PnfdInfo .....	97
5.5.2.6	Type: CreatePnfdInfoRequest .....	98
5.5.2.7	Type: NsdmSubscriptionRequest .....	98
5.5.2.8	Type: NsdmSubscription .....	99
5.5.2.9	Type: NsdOnboardingNotification .....	99
5.5.2.10	Type: NsdOnboardingFailureNotification .....	99
5.5.2.11	Type: NsdChangeNotification .....	100
5.5.2.12	Type: NsdDeletionNotification .....	100
5.5.2.13	Type: PnfdOnboardingNotification .....	101
5.5.2.14	Type: PnfdOnboardingFailureNotification .....	101
5.5.2.15	Type: PnfdDeletionNotification .....	102
5.5.3	Referenced structured data types .....	102
5.5.3.1	Introduction .....	102
5.5.3.2	Type: NsdmNotificationsFilter .....	102
5.5.3.3	Type: NsdmLinks .....	103
5.5.3.4	Type: PnfdmLinks .....	104
5.5.3.5	Type: NsdArchiveArtifactInfo .....	104
5.5.3.6	Type: PnfdArchiveArtifactInfo .....	104
5.5.4	Referenced simple data types and enumerations .....	105
5.5.4.1	Introduction .....	105
5.5.4.2	Simple data types .....	105
5.5.4.3	Enumeration: NsdOperationalStateType .....	105
5.5.4.4	Enumeration: NsdUsageStateType .....	105
5.5.4.5	Enumeration: NsdOnboardingStateType .....	105
5.5.4.6	Enumeration: PnfdOnboardingStateType .....	105
5.5.4.7	Enumeration: PnfdUsageStateType .....	106
6	NS Lifecycle Management interface .....	106
6.1	Description .....	106
6.1a	API version .....	107
6.2	Resource structure and methods .....	107
6.3	Sequence diagrams (informative) .....	109
6.3.1	Flow of the creation of an NS instance resource .....	109
6.3.2	Flow of the deletion of an NS instance resource .....	109
6.3.3	Flow of NS lifecycle management operations triggered by task resources .....	110
6.3.3a	Flows of NS lifecycle management operations triggered by task resources with pre-emption .....	112
6.3.3a.1	Overview .....	112
6.3.3a.2	Steps shared by multiple flows .....	113

6.3.3a.3	Shortage Use Case #1.....	115
6.3.3a.4	Shortage Use Case #2.....	116
6.3.3a.5	Shortage Use Case #3.....	119
6.3.3a.6	Shortage Use Case #4.....	123
6.3.3b	Flow of NS lifecycle management operations triggered by task resources with feasibility check .....	124
6.3.4	Flow of the get operations status operation .....	128
6.3.5	Flow of managing subscriptions .....	129
6.3.6	Flow of sending notifications.....	130
6.3.7	Flow of retrying an NS lifecycle management operation .....	131
6.3.8	Flow of rolling back an NS lifecycle management operation.....	133
6.3.9	Flow of continuing an NS lifecycle management operation.....	134
6.3.10	Flow of failing an NS lifecycle management operation.....	135
6.3.11	Flow of cancelling an NS lifecycle management operation.....	136
6.3.12	Flow of the query VNF snapshot operation .....	137
6.3.13	Flow of the deletion of a VNF snapshot resource.....	138
6.3.14	Flow of the Modify an Individual NS LCM Information operation .....	139
6.4	Resources .....	140
6.4.1	Introduction.....	140
6.4.1.1	Overview .....	140
6.4.1.2	Task resources that trigger NS LCM operations .....	140
6.4.1a	Resource: API versions.....	141
6.4.2	Resource: NS Instances .....	141
6.4.2.1	Description .....	141
6.4.2.2	Resource definition .....	141
6.4.2.3	Resource methods .....	142
6.4.2.3.1	POST .....	142
6.4.2.3.2	GET .....	142
6.4.2.3.3	PUT .....	144
6.4.2.3.4	PATCH.....	144
6.4.2.3.5	DELETE.....	145
6.4.3	Resource: Individual NS Instance.....	145
6.4.3.1	Description .....	145
6.4.3.2	Resource definition .....	145
6.4.3.3	Resource methods .....	145
6.4.3.3.1	POST .....	145
6.4.3.3.2	GET .....	145
6.4.3.3.3	PUT .....	146
6.4.3.3.4	PATCH.....	146
6.4.3.3.5	DELETE.....	146
6.4.4	Resource: Instantiate NS task .....	147
6.4.4.1	Description .....	147
6.4.4.2	Resource definition .....	147
6.4.4.3	Resource methods .....	147
6.4.4.3.1	POST .....	147
6.4.4.3.2	GET .....	148
6.4.4.3.3	PUT .....	148
6.4.4.3.4	PATCH.....	148
6.4.4.3.5	DELETE.....	149
6.4.5	Resource: Scale NS task .....	149
6.4.5.1	Description .....	149
6.4.5.2	Resource definition .....	149
6.4.5.3	Resource methods .....	149
6.4.5.3.1	POST .....	149
6.4.5.3.2	GET .....	150
6.4.5.3.3	PUT .....	150
6.4.5.3.4	PATCH.....	150
6.4.5.3.5	DELETE.....	150
6.4.6	Resource: Update NS task .....	151
6.4.6.1	Description .....	151
6.4.6.2	Resource definition .....	151
6.4.6.3	Resource methods .....	151
6.4.6.3.1	POST .....	151

6.4.6.3.2	GET .....	152
6.4.6.3.3	PUT .....	152
6.4.6.3.4	PATCH.....	152
6.4.6.3.5	DELETE.....	152
6.4.7	Resource: Heal NS task .....	152
6.4.7.1	Description .....	152
6.4.7.2	Resource definition .....	153
6.4.7.3	Resource methods .....	153
6.4.7.3.1	POST .....	153
6.4.7.3.2	GET .....	154
6.4.7.3.3	PUT .....	154
6.4.7.3.4	PATCH.....	154
6.4.7.3.5	DELETE.....	154
6.4.8	Resource: Terminate NS task.....	154
6.4.8.1	Description .....	154
6.4.8.2	Resource definition .....	154
6.4.8.3	Resource methods .....	154
6.4.8.3.1	POST .....	154
6.4.8.3.2	GET .....	155
6.4.8.3.3	PUT .....	155
6.4.8.3.4	PATCH.....	155
6.4.8.3.5	DELETE.....	155
6.4.9	Resource: NS LCM operation occurrences.....	156
6.4.9.1	Description .....	156
6.4.9.2	Resource definition .....	156
6.4.9.3	Resource methods .....	156
6.4.9.3.1	POST .....	156
6.4.9.3.2	GET .....	156
6.4.9.3.3	PUT .....	158
6.4.9.3.4	PATCH.....	158
6.4.9.3.5	DELETE.....	158
6.4.10	Resource: Individual NS LCM operation occurrence.....	158
6.4.10.1	Description .....	158
6.4.10.2	Resource definition .....	158
6.4.10.3	Resource methods .....	158
6.4.10.3.1	POST .....	158
6.4.10.3.2	GET .....	159
6.4.10.3.3	PUT .....	159
6.4.10.3.4	PATCH.....	159
6.4.10.3.5	DELETE.....	160
6.4.11	Resource: Retry operation task .....	160
6.4.11.1	Description .....	160
6.4.11.2	Resource definition .....	161
6.4.11.3	Resource methods .....	161
6.4.11.3.1	POST .....	161
6.4.11.3.2	GET .....	162
6.4.11.3.3	PUT .....	162
6.4.11.3.4	PATCH.....	162
6.4.11.3.5	DELETE.....	162
6.4.12	Resource: Rollback operation task.....	162
6.4.12.1	Description .....	162
6.4.12.2	Resource definition .....	162
6.4.12.3	Resource methods .....	163
6.4.12.3.1	POST .....	163
6.4.12.3.2	GET .....	164
6.4.12.3.3	PUT .....	164
6.4.12.3.4	PATCH.....	164
6.4.12.3.5	DELETE.....	164
6.4.13	Resource: Continue operation task .....	164
6.4.13.1	Description .....	164
6.4.13.2	Resource definition .....	164
6.4.13.3	Resource methods .....	165



6.4.13.3.1	POST .....	165
6.4.13.3.2	GET .....	166
6.4.13.3.3	PUT .....	166
6.4.13.3.4	PATCH.....	166
6.4.13.3.5	DELETE.....	166
6.4.14	Resource: Fail operation task.....	166
6.4.14.1	Description .....	166
6.4.14.2	Resource definition .....	166
6.4.14.3	Resource methods .....	167
6.4.14.3.1	POST .....	167
6.4.14.3.2	GET .....	168
6.4.14.3.3	PUT .....	168
6.4.14.3.4	PATCH.....	168
6.4.14.3.5	DELETE.....	169
6.4.15	Resource: Cancel operation task.....	169
6.4.15.1	Description .....	169
6.4.15.2	Resource definition .....	169
6.4.15.3	Resource methods .....	169
6.4.15.3.1	POST .....	169
6.4.15.3.2	GET .....	170
6.4.15.3.3	PUT .....	170
6.4.15.3.4	PATCH.....	171
6.4.15.3.5	DELETE.....	171
6.4.16	Resource: Subscriptions.....	171
6.4.16.1	Description .....	171
6.4.16.2	Resource definition .....	171
6.4.16.3	Resource methods .....	171
6.4.16.3.1	POST .....	171
6.4.16.3.2	GET .....	172
6.4.16.3.3	PUT .....	173
6.4.16.3.4	PATCH.....	174
6.4.16.3.5	DELETE.....	174
6.4.17	Resource: Individual subscription.....	174
6.4.17.1	Description .....	174
6.4.17.2	Resource definition .....	174
6.4.17.3	Resource methods .....	174
6.4.17.3.1	POST .....	174
6.4.17.3.2	GET .....	174
6.4.17.3.3	PUT .....	175
6.4.17.3.4	PATCH.....	175
6.4.17.3.5	DELETE.....	175
6.4.18	Resource: Notification endpoint .....	176
6.4.18.1	Description .....	176
6.4.18.2	Resource definition .....	176
6.4.18.3	Resource methods .....	176
6.4.18.3.1	POST .....	176
6.4.18.3.2	GET .....	177
6.4.18.3.3	PUT .....	177
6.4.18.3.4	PATCH.....	177
6.4.18.3.5	DELETE.....	177
6.4.19	Resource: VNF snapshots.....	177
6.4.19.1	Description .....	177
6.4.19.2	Resource definition .....	177
6.4.19.3	Resource methods .....	178
6.4.19.3.1	POST .....	178
6.4.19.3.2	GET .....	178
6.4.19.3.3	PUT .....	179
6.4.19.3.4	PATCH.....	179
6.4.19.3.5	DELETE.....	179
6.4.20	Resource: Individual VNF snapshot .....	179
6.4.20.1	Description .....	179
6.4.20.2	Resource definition .....	180

6.4.20.3	Resource methods .....	180
6.4.20.3.1	POST .....	180
6.4.20.3.2	GET .....	180
6.4.20.3.3	PUT .....	180
6.4.20.3.4	PATCH .....	181
6.4.20.3.5	DELETE .....	181
6.5	Data model .....	182
6.5.1	Introduction .....	182
6.5.2	Resource and notification data types .....	182
6.5.2.1	Introduction .....	182
6.5.2.2	Type: LccnSubscriptionRequest .....	182
6.5.2.3	Type: NsLcmOpOcc .....	182
6.5.2.4	Type: LccnSubscription .....	185
6.5.2.5	Type: NsLcmOperationOccurrenceNotification .....	185
6.5.2.6	Type: NsIdentifierCreationNotification .....	187
6.5.2.7	Type: NsIdentifierDeletionNotification .....	188
6.5.2.8	Type: NsChangeNotification .....	188
6.5.2.9	Type: CreateNsRequest .....	189
6.5.2.10	Type: NsInstance .....	190
6.5.2.11	Type: InstantiateNsRequest .....	192
6.5.2.12	Type: UpdateNsRequest .....	193
6.5.2.13	Type: HealNsRequest .....	197
6.5.2.14	Type: ScaleNsRequest .....	197
6.5.2.15	Type: TerminateNsRequest .....	197
6.5.2.16	Type: CancelMode .....	198
6.5.2.17	Type: VnfSnapshotInfo .....	198
6.5.2.18	Type: VnfSnapshot .....	199
6.5.2.19	Type: NsLcmCapacityShortageNotification .....	199
6.5.2.20	Type: NsLcmInfoModificationRequest .....	201
6.5.3	Referenced structured data types .....	202
6.5.3.1	Introduction .....	202
6.5.3.2	Type: AffectedVnf .....	202
6.5.3.3	Type: AffectedPnf .....	203
6.5.3.4	Type: AffectedVirtualLink .....	203
6.5.3.5	Type: AffectedVnffg .....	204
6.5.3.6	Type: AffectedNs .....	204
6.5.3.7	Type: AffectedSap .....	205
6.5.3.8	Type: LifecycleChangeNotificationsFilter .....	205
6.5.3.9	Type: LccnLinks .....	207
6.5.3.10	Type: SapData .....	207
6.5.3.11	Type: CpProtocolData .....	207
6.5.3.12	Type: IpOverEthernetAddressData .....	208
6.5.3.12a	Type: VirtualCpAddressData .....	209
6.5.3.13	Type: PnfInfo .....	209
6.5.3.14	Type: AddPnfData .....	210
6.5.3.15	Type: ModifyPnfData .....	210
6.5.3.16	Type: PnfExtCpData .....	210
6.5.3.17	Type: PnfExtCpInfo .....	211
6.5.3.18	Type: IpOverEthernetAddressInfo .....	211
6.5.3.18a	Type: VirtualCpAddressInfo .....	212
6.5.3.19	Type: VnfInstanceData .....	212
6.5.3.19a	Type: NestedNsInstanceData .....	213
6.5.3.20	Type: VnfLocationConstraint .....	214
6.5.3.21	Type: LocationConstraints .....	214
6.5.3.21a	Type: ParamsForNestedNs .....	215
6.5.3.22	Type: ParamsForVnf .....	216
6.5.3.23	Type: AffinityOrAntiAffinityRule .....	217
6.5.3.24	Type: InstantiateVnfData .....	218
6.5.3.25	Type: ChangeVnfFlavourData .....	220
6.5.3.26	Type: ExtVirtualLinkData .....	221
6.5.3.27	Type: ExtManagedVirtualLinkData .....	222
6.5.3.28	Type: ExtLinkPortData .....	222

6.5.3.29	Type: VnfExtCpData .....	223
6.5.3.30	Type: VnfExtCpConfig .....	223
6.5.3.31	Type: OperateVnfData .....	225
6.5.3.32	Type: ModifyVnfInfoData .....	225
6.5.3.33	Type: ChangeExtVnfConnectivityData .....	226
6.5.3.34	Type: AssocNewNsdVersionData .....	226
6.5.3.35	Type: MoveVnfInstanceData .....	227
6.5.3.36	Type: AddVnffgData .....	227
6.5.3.37	Type: UpdateVnffgData .....	227
6.5.3.38	Type: NfpData .....	228
6.5.3.39	Type: ChangeNsFlavourData .....	228
6.5.3.40	Type: NfpRule .....	228
6.5.3.41	Type: Mask .....	229
6.5.3.42	Type: PortRange .....	230
6.5.3.43	Type: HealNsData .....	230
6.5.3.44	Type: HealVnfData .....	230
6.5.3.45	Type: ScaleNsData .....	231
6.5.3.46	Type: ScaleNsByStepsData .....	231
6.5.3.47	Type: ScaleNsToLevelData .....	232
6.5.3.48	Type: NsScaleInfo .....	232
6.5.3.49	Type: ScaleVnfData .....	232
6.5.3.50	Type: ScaleToLevelData .....	233
6.5.3.51	Type: VnfScaleInfo .....	233
6.5.3.52	Type: ScaleByStepData .....	234
6.5.3.53	Type: NsVirtualLinkInfo .....	234
6.5.3.54	Type: ChangeVnfPackageData .....	234
6.5.3.55	Type: NsLinkPortInfo .....	236
6.5.3.56	Type: NsCpHandle .....	236
6.5.3.57	Type: VnfInstance .....	236
6.5.3.58	Type: CpProtocolInfo .....	240
6.5.3.59	Type: ExtManagedVirtualLinkInfo .....	241
6.5.3.60	Type: VnfcResourceInfo .....	241
6.5.3.61	Type: VnfVirtualLinkResourceInfo .....	242
6.5.3.62	Type: ExtVirtualLinkInfo .....	243
6.5.3.63	Type: ExtLinkPortInfo .....	243
6.5.3.64	Type: VnfLinkPortInfo .....	244
6.5.3.65	Type: VnffgInfo .....	245
6.5.3.66	Type: NfpInfo .....	245
6.5.3.67	Type: SapInfo .....	246
6.5.3.68	Type: NsMonitoringParameter .....	246
6.5.3.69	Type: VnfMonitoringParameter .....	246
6.5.3.70	Type: VnfExtCpInfo .....	247
6.5.3.71	Type: CpGroupInfo .....	248
6.5.3.72	Type: CpPairInfo .....	248
6.5.3.73	Type: ForwardingBehaviourInputParameters .....	248
6.5.3.74	Type: DeleteVnfSnapshotData .....	249
6.5.3.75	Type: RevertVnfToSnapshotData .....	249
6.5.3.76	Type: CreateVnfSnapshotData .....	249
6.5.3.77	Type: VnfcSnapshotInfo .....	250
6.5.3.78	Type: VnfLinkPortData .....	250
6.5.3.79	Type: ModificationsTriggeredByVnfPkgChange .....	251
6.5.3.80	Type: WanConnectionData .....	251
6.5.3.81	Type: WanConnectionProtocolData .....	252
6.5.3.82	Type: MscsInfo .....	252
6.5.3.83	Type: MscsEndpointInfo .....	253
6.5.3.84	Type: ConnectivityServiceEndpointInfo .....	254
6.5.3.85	Type: SiteToWanLayer2ProtocolData .....	254
6.5.3.86	Type: SiteToWanLayer3ProtocolData .....	256
6.5.3.87	Type: MscsConfigData .....	257
6.5.3.88	Type: WanLayer2ProtocolData .....	258
6.5.3.89	Type: WanLayer3ProtocolData .....	258
6.5.3.90	Type: WanConnectionInfo .....	259

6.5.3.91	Type: WanConnectionProtocolInfo .....	259
6.5.3.92	Type: ModifyWanConnectionInfoData .....	259
6.5.3.93	Type: WanConnectionInfoModification .....	260
6.5.3.94	Type: VirtualStorageResourceInfo .....	261
6.5.3.95	Type: AddNsVirtualLinkData.....	261
6.5.3.96	Type: NestedNsLocationConstraint .....	262
6.5.3.97	Type: VipCpInfo .....	262
6.5.3.98	Type: TerminateNsData .....	262
6.5.3.99	Type: TerminateVnfData .....	263
6.5.3.100	Type: DataFlowMirroringData .....	263
6.5.3.101	Type: DataFlowData .....	263
6.5.3.102	Type: DataFlowDetails .....	264
6.5.3.103	Type: CollectorDetails .....	264
6.5.3.104	Type: CreateDataFlowMirroring.....	264
6.5.3.105	Type: UpdateDataFlowMirroring .....	264
6.5.3.106	Type: MirroringInfo .....	265
6.5.3.107	Type: McioInfo .....	265
6.5.3.108	Type: VirtualCpInfo.....	266
6.5.3.109	Type: AdditionalServiceInfo.....	266
6.5.3.110	Type: ServicePortInfo .....	267
6.5.3.111	Type: OverridingVersionDependency .....	267
6.5.3.112	Type: VersionDependency.....	267
6.5.3.113	Type: VersionDependencyStatement .....	268
6.5.3.114	Type: NetAttDefResourceInfo .....	268
6.5.3.115	Type: NetAttDefResourceData .....	268
6.5.3.116	Type: IntVnfCpData.....	269
6.5.4	Referenced simple data types and enumerations .....	269
6.5.4.1	Introduction .....	269
6.5.4.2	Simple data types .....	269
6.5.4.3	Enumeration: NsLcmOpType .....	269
6.5.4.4	Enumeration: NsLcmOperationStateType .....	270
6.5.4.5	Enumeration: NsComponentType.....	270
6.5.4.6	Enumeration: LcmOpNameForChangeNotificationType .....	270
6.5.4.7	Enumeration: LcmOpOccStatusForChangeNotificationType.....	271
6.5.4.8	Enumeration: OperationalStates.....	271
6.5.4.9	Enumeration: StopType .....	271
6.5.4.10	Enumeration: CancelModeType .....	271
6.5.4.11	Enumeration: LcmOpOccNotificationVerbosityType .....	272
6.5.4.12	Enumeration: DirectionType.....	272
6.5.4.13	Type: McioTypeName .....	272
6.5.4.14	Enumeration: FeasibilityCheckReserveType.....	272
6.5.4.15	Enumeration: FeasibilityCheckResultType.....	273
6.6	Handling of errors during NS lifecycle management operations.....	273
6.6.1	Basic concepts (informative) .....	273
6.6.1.1	Motivation.....	273
6.6.1.2	Failure resolution strategies: Retry, Rollback and Continue.....	273
6.6.1.3	Error handling at NFVO and OSS/BSS .....	274
6.6.2	States and state transitions of an NS lifecycle management operation occurrence.....	275
6.6.2.1	General .....	275
6.6.2.2	States of an NS lifecycle management operation occurrence.....	276
6.6.2.3	Error handling operations that change the state of an NS lifecycle operation.....	278
6.6.3	Detailed flows .....	279
6.6.3.1	Immediate failure .....	279
6.6.3.2	Failure during actual NS LCM operation execution .....	280
6.6.3.3	LCM operation cancellation.....	281
6.7	Handling of security-sensitive attributes .....	282
7	NS Performance Management interface.....	282
7.1	Description .....	282
7.1a	API version.....	282
7.2	Resource structure and methods .....	282
7.3	Sequence diagrams (informative).....	283

7.3.1	Flow of creating a PM job .....	283
7.3.1a	Flow of updating the callback URI of a PM job .....	284
7.3.2	Flow of querying/reading PM jobs .....	285
7.3.3	Flow of deleting a PM job .....	286
7.3.4	Flow of obtaining performance reports.....	287
7.3.5	Flow of creating a threshold .....	288
7.3.5a	Flow of updating the callback URI of a threshold .....	288
7.3.6	Flow of querying/reading thresholds .....	289
7.3.7	Flow of deleting thresholds.....	290
7.3.8	Void .....	290
7.3.9	Flow of sending notifications.....	290
7.4	Resources .....	291
7.4.1	Introduction.....	291
7.4.1a	Resource: API versions.....	291
7.4.2	Resource: PM jobs .....	291
7.4.2.1	Description.....	291
7.4.2.2	Resource definition .....	291
7.4.2.3	Resource methods .....	292
7.4.2.3.1	POST .....	292
7.4.2.3.2	GET .....	293
7.4.2.3.3	PUT .....	294
7.4.2.3.4	PATCH.....	294
7.4.2.3.5	DELETE.....	294
7.4.3	Resource: Individual PM job .....	294
7.4.3.1	Description .....	294
7.4.3.2	Resource definition .....	294
7.4.3.3	Resource methods .....	295
7.4.3.3.1	POST .....	295
7.4.3.3.2	GET .....	295
7.4.3.3.3	PUT .....	295
7.4.3.3.4	PATCH.....	295
7.4.3.3.5	DELETE.....	296
7.4.4	Resource: Individual performance report .....	297
7.4.4.1	Description .....	297
7.4.4.2	Resource definition .....	297
7.4.4.3	Resource methods .....	297
7.4.4.3.1	POST .....	297
7.4.4.3.2	GET .....	297
7.4.4.3.3	PUT .....	298
7.4.4.3.4	PATCH.....	298
7.4.4.3.5	DELETE.....	298
7.4.5	Resource: Thresholds.....	298
7.4.5.1	Description .....	298
7.4.5.2	Resource definition .....	298
7.4.5.3	Resource methods .....	299
7.4.5.3.1	POST .....	299
7.4.5.3.2	GET .....	300
7.4.5.3.3	PUT .....	301
7.4.5.3.4	PATCH.....	301
7.4.5.3.5	DELETE.....	301
7.4.6	Resource: Individual threshold .....	301
7.4.6.1	Description .....	301
7.4.6.2	Resource definition .....	301
7.4.6.3	Resource methods .....	301
7.4.6.3.1	POST .....	301
7.4.6.3.2	GET .....	301
7.4.6.3.3	PUT .....	302
7.4.6.3.4	PATCH.....	302
7.4.6.3.5	DELETE.....	303
7.4.7	Void .....	303
7.4.8	Void .....	303
7.4.9	Resource: Notification endpoint .....	304

7.4.9.1	Description .....	304
7.4.9.2	Resource definition .....	304
7.4.9.3	Resource methods .....	304
7.4.9.3.1	POST .....	304
7.4.9.3.2	GET .....	304
7.4.9.3.3	PUT .....	305
7.4.9.3.4	PATCH .....	305
7.4.9.3.5	DELETE .....	305
7.5	Data Model .....	305
7.5.1	Introduction .....	305
7.5.2	Resource and notification data types .....	305
7.5.2.1	Introduction .....	305
7.5.2.2	Void .....	305
7.5.2.3	Void .....	305
7.5.2.4	Type: ThresholdCrossedNotification .....	306
7.5.2.5	Type: PerformanceInformationAvailableNotification .....	306
7.5.2.6	Type: CreatePmJobRequest .....	308
7.5.2.7	Type: PmJob .....	309
7.5.2.8	Type: CreateThresholdRequest .....	309
7.5.2.9	Type: Threshold .....	310
7.5.2.10	Type: PerformanceReport .....	311
7.5.2.11	Type: ThresholdModifications .....	311
7.5.2.12	Type: PmJobModifications .....	312
7.5.3	Referenced structured data types .....	312
7.5.3.1	Introduction .....	312
7.5.3.2	Void .....	312
7.5.3.3	Type: PmJobCriteria .....	312
7.5.3.4	Type: ThresholdCriteria .....	313
7.5.4	Referenced simple data types and enumerations .....	313
7.5.4.1	Introduction .....	313
7.5.4.2	Simple data types .....	313
7.5.4.3	Enumeration: CrossingDirectionType .....	314
8	NS Fault Management interface .....	314
8.1	Description .....	314
8.1a	API version .....	314
8.2	Resource structure and methods .....	314
8.3	Sequence diagrams (informative) .....	315
8.3.1	Flow of the Get Alarm List operation .....	315
8.3.2	Flow of acknowledging alarm .....	316
8.3.3	Flow of managing subscriptions .....	317
8.3.4	Flow of sending notifications .....	318
8.4	Resources .....	319
8.4.1	Introduction .....	319
8.4.1a	Resource: API versions .....	319
8.4.2	Resource: Alarms .....	319
8.4.2.1	Description .....	319
8.4.2.2	Resource definition .....	319
8.4.2.3	Resource methods .....	319
8.4.2.3.1	POST .....	319
8.4.2.3.2	GET .....	319
8.4.2.3.3	PUT .....	321
8.4.2.3.4	PATCH .....	321
8.4.2.3.5	DELETE .....	321
8.4.3	Resource: Individual alarm .....	321
8.4.3.1	Description .....	321
8.4.3.2	Resource definition .....	321
8.4.3.3	Resource methods .....	321
8.4.3.3.1	POST .....	321
8.4.3.3.2	GET .....	321
8.4.3.3.3	PUT .....	322
8.4.3.3.4	PATCH .....	322

8.4.3.3.5	DELETE .....	323
8.4.4	Resource: Subscriptions.....	323
8.4.4.1	Description .....	323
8.4.4.2	Resource definition .....	323
8.4.4.3	Resource methods .....	324
8.4.4.3.1	POST .....	324
8.4.4.3.2	GET .....	325
8.4.4.3.3	PUT .....	326
8.4.4.3.4	PATCH.....	326
8.4.4.3.5	DELETE.....	326
8.4.5	Resource: Individual subscription.....	327
8.4.5.1	Description .....	327
8.4.5.2	Resource definition .....	327
8.4.5.3	Resource methods .....	327
8.4.5.3.1	POST .....	327
8.4.5.3.2	GET .....	327
8.4.5.3.3	PUT .....	328
8.4.5.3.4	PATCH.....	328
8.4.5.3.5	DELETE.....	328
8.4.6	Resource: Notification endpoint .....	328
8.4.6.1	Description .....	328
8.4.6.2	Resource definition .....	328
8.4.6.3	Resource methods .....	329
8.4.6.3.1	POST .....	329
8.4.6.3.2	GET .....	329
8.4.6.3.3	PUT .....	330
8.4.6.3.4	PATCH.....	330
8.4.6.3.5	DELETE.....	330
8.5	Data Model.....	330
8.5.1	Introduction.....	330
8.5.2	Resource and notification data types .....	330
8.5.2.1	Introduction.....	330
8.5.2.2	Type: FmSubscriptionRequest.....	330
8.5.2.3	Type: FmSubscription.....	331
8.5.2.4	Type: Alarm .....	331
8.5.2.5	Type: AlarmNotification.....	332
8.5.2.6	Type: AlarmClearedNotification.....	332
8.5.2.7	Type: AlarmListRebuiltNotification .....	333
8.5.2.8	Type: AlarmModifications.....	333
8.5.3	Referenced structured data types .....	334
8.5.3.1	Introduction.....	334
8.5.3.2	Type: FmNotificationsFilter.....	334
8.5.3.3	Type: FaultyResourceInfo.....	334
8.5.3.4	Type: FaultyComponentInfo .....	334
8.5.4	Referenced simple data types and enumerations .....	335
8.5.4.1	Introduction.....	335
8.5.4.2	Simple data types .....	335
8.5.4.3	Enumeration: PerceivedSeverityType.....	335
8.5.4.4	Enumeration: EventType .....	335
8.5.4.5	Enumeration: FaultyResourceType.....	336
9	VNF Package Management interface.....	336
9.1	Description .....	336
9.1a	API version.....	337
9.2	Resource structure and methods .....	337
9.3	Sequence diagrams (informative).....	339
9.3.1	Flow of the creation of an individual VNF package resource.....	339
9.3.2	Flow of the uploading of VNF package content .....	340
9.3.2a	Flow of handling errors during upload of VNF package content.....	341
9.3.2b	Flow of fetching the VNF package manifest .....	343
9.3.3	Flow of querying/reading VNF package information .....	344
9.3.4	Flow of reading the VNFD of an on-boarded VNF package .....	344

9.3.4a	Flow of bulk-fetching VNF package artifacts that are not images.....	345
9.3.5	Flow of updating information of a VNF package.....	346
9.3.6	Flow of deleting a VNF package resource.....	347
9.3.7	Flow of fetching an on-boarded VNF package.....	348
9.3.8	Flow of fetching a VNF package artifact.....	349
9.3.9	Flow of managing subscriptions.....	349
9.3.10	Flow of sending notifications.....	351
9.4	Resources.....	352
9.4.1	Introduction.....	352
9.4.1a	Resource: API versions.....	352
9.4.2	Resource: VNF packages.....	352
9.4.2.1	Description.....	352
9.4.2.2	Resource definition.....	352
9.4.2.3	Resource methods.....	352
9.4.2.3.1	POST.....	352
9.4.2.3.2	GET.....	353
9.4.2.3.3	PUT.....	354
9.4.2.3.4	PATCH.....	354
9.4.2.3.5	DELETE.....	355
9.4.3	Resource: Individual VNF package.....	355
9.4.3.1	Description.....	355
9.4.3.2	Resource definition.....	355
9.4.3.3	Resource methods.....	355
9.4.3.3.1	POST.....	355
9.4.3.3.2	GET.....	355
9.4.3.3.3	PUT.....	356
9.4.3.3.4	PATCH.....	356
9.4.3.3.5	DELETE.....	357
9.4.4	Resource: VNFD in an individual VNF package.....	357
9.4.4.1	Description.....	357
9.4.4.2	Resource definition.....	358
9.4.4.3	Resource methods.....	358
9.4.4.3.1	POST.....	358
9.4.4.3.2	GET.....	358
9.4.4.3.3	PUT.....	360
9.4.4.3.4	PATCH.....	360
9.4.4.3.5	DELETE.....	361
9.4.4a	Resource: Access configuration for external artifacts.....	361
9.4.4a.1	Description.....	361
9.4.4a.2	Resource definition.....	361
9.4.4a.3	Resource methods.....	361
9.4.4a.3.1	POST.....	361
9.4.4a.3.2	GET.....	361
9.4.4a.3.3	PUT.....	362
9.4.4a.3.4	PATCH.....	363
9.4.4a.3.5	DELETE.....	363
9.4.4b	Resource: Manifest in an individual VNF package.....	364
9.4.4b.1	Description.....	364
9.4.4b.2	Resource definition.....	364
9.4.4b.3	Resource methods.....	364
9.4.4b.3.1	POST.....	364
9.4.4b.3.2	GET.....	364
9.4.4b.3.3	PUT.....	365
9.4.4b.3.4	PATCH.....	365
9.4.4b.3.5	DELETE.....	365
9.4.5	Resource: VNF package content.....	366
9.4.5.1	Description.....	366
9.4.5.2	Resource definition.....	366
9.4.5.3	Resource methods.....	366
9.4.5.3.1	POST.....	366
9.4.5.3.2	GET.....	366
9.4.5.3.3	PUT.....	367



9.4.5.3.4	PATCH .....	368
9.4.5.3.5	DELETE .....	368
9.4.5a	Resource: VNF package artifacts .....	369
9.4.5a.1	Description .....	369
9.4.5a.2	Resource definition .....	369
9.4.5a.3	Resource methods .....	369
9.4.5a.3.1	POST .....	369
9.4.5a.3.2	GET .....	369
9.4.5a.3.3	PUT .....	372
9.4.5a.3.4	PATCH .....	372
9.4.5a.3.5	DELETE .....	372
9.4.6	Resource: Upload VNF package from URI task .....	372
9.4.6.1	Description .....	372
9.4.6.2	Resource definition .....	372
9.4.6.3	Resource methods .....	372
9.4.6.3.1	POST .....	372
9.4.6.3.2	GET .....	373
9.4.6.3.3	PUT .....	373
9.4.6.3.4	PATCH .....	373
9.4.6.3.5	DELETE .....	373
9.4.7	Resource: Individual VNF package artifact .....	374
9.4.7.1	Description .....	374
9.4.7.2	Resource definition .....	374
9.4.7.3	Resource methods .....	374
9.4.7.3.1	POST .....	374
9.4.7.3.2	GET .....	374
9.4.7.3.3	PUT .....	376
9.4.7.3.4	PATCH .....	376
9.4.7.3.5	DELETE .....	376
9.4.8	Resource: Subscriptions .....	376
9.4.8.1	Description .....	376
9.4.8.2	Resource definition .....	376
9.4.8.3	Resource methods .....	377
9.4.8.3.1	POST .....	377
9.4.8.3.2	GET .....	378
9.4.8.3.3	PUT .....	379
9.4.8.3.4	PATCH .....	379
9.4.8.3.5	DELETE .....	379
9.4.9	Resource: Individual subscription .....	380
9.4.9.1	Description .....	380
9.4.9.2	Resource definition .....	380
9.4.9.3	Resource methods .....	380
9.4.9.3.1	POST .....	380
9.4.9.3.2	GET .....	380
9.4.9.3.3	PUT .....	381
9.4.9.3.4	PATCH .....	381
9.4.9.3.5	DELETE .....	381
9.4.10	Resource: Notification endpoint .....	381
9.4.10.1	Description .....	381
9.4.10.2	Resource definition .....	381
9.4.10.3	Resource methods .....	382
9.4.10.3.1	POST .....	382
9.4.10.3.2	GET .....	382
9.4.10.3.3	PUT .....	383
9.4.10.3.4	PATCH .....	383
9.4.10.3.5	DELETE .....	383
9.5	Data model .....	383
9.5.1	Introduction .....	383
9.5.2	Resource and notification data types .....	383
9.5.2.1	Introduction .....	383
9.5.2.2	Type: CreateVnfPkgInfoRequest .....	383
9.5.2.3	Type: VnfPkgInfoModifications .....	384

9.5.2.4	Type: UploadVnfPackageFromUriRequest .....	384
9.5.2.5	Type: VnfPkgInfo .....	384
9.5.2.6	Type: PkgmSubscriptionRequest .....	387
9.5.2.7	Type: PkgmSubscription .....	387
9.5.2.8	Type: VnfPackageOnboardingNotification .....	387
9.5.2.9	Type: VnfPackageChangeNotification .....	388
9.5.2.10	Type: ExternalArtifactsAccessConfig .....	389
9.5.3	Referenced structured data types .....	389
9.5.3.1	Introduction .....	389
9.5.3.2	Type: VnfPackageSoftwareImageInfo .....	390
9.5.3.3	Type: VnfPackageArtifactInfo .....	391
9.5.3.4	Type: PkgmNotificationsFilter .....	392
9.5.3.5	Type: PkgmLinks .....	393
9.5.3.6	Void .....	393
9.5.4	Referenced simple data types and enumerations .....	393
9.5.4.1	Introduction .....	393
9.5.4.2	Simple data types .....	393
9.5.4.3	Enumeration: PackageOnboardingStateType .....	394
9.5.4.4	Enumeration: PackageOperationalStateType .....	394
9.5.4.5	Enumeration: PackageUsageStateType .....	394
9.5.4.6	Enumeration: PackageChangeType .....	394
10	NFVI Capacity Information interface .....	394
10.1	Description .....	394
10.1a	API version .....	395
10.2	Resource structure and methods .....	395
10.3	Sequence diagrams (informative) .....	396
10.3.1	Flow of querying/reading NFVI capacity information .....	396
10.3.2	Flow of creating a capacity threshold .....	397
10.3.3	Flow of querying/reading capacity thresholds .....	397
10.3.4	Flow of deleting a capacity threshold .....	398
10.3.5	Flow of updating the callback URI of a threshold .....	399
10.3.6	Flow of sending notifications .....	399
10.4	Resources .....	400
10.4.1	Introduction .....	400
10.4.1a	Resource: API versions .....	400
10.4.2	Resource: NFVI capacity information .....	400
10.4.2.1	Description .....	400
10.4.2.2	Resource definition .....	400
10.4.2.3	Resource methods .....	401
10.4.2.3.1	POST .....	401
10.4.2.3.2	GET .....	401
10.4.2.3.3	PUT .....	402
10.4.2.3.4	PATCH .....	402
10.4.2.3.5	DELETE .....	402
10.4.3	Resource: Individual VIM's NFVI capacity information .....	403
10.4.3.1	Description .....	403
10.4.3.2	Resource definition .....	403
10.4.3.3	Resource methods .....	403
10.4.3.3.1	POST .....	403
10.4.3.3.2	GET .....	403
10.4.3.3.3	PUT .....	404
10.4.3.3.4	PATCH .....	404
10.4.3.3.5	DELETE .....	404
10.4.4	Resource: Capacity thresholds .....	404
10.4.4.1	Description .....	404
10.4.4.2	Resource definition .....	404
10.4.4.3	Resource methods .....	405
10.4.4.3.1	POST .....	405
10.4.4.3.2	GET .....	406
10.4.4.3.3	PUT .....	407
10.4.4.3.4	PATCH .....	407

10.4.4.3.5	DELETE .....	407
10.4.5	Resource: Individual capacity threshold .....	407
10.4.5.1	Description .....	407
10.4.5.2	Resource definition .....	407
10.4.5.3	Resource methods .....	407
10.4.5.3.1	POST .....	407
10.4.5.3.2	GET .....	407
10.4.5.3.3	PUT .....	408
10.4.5.3.4	PATCH .....	408
10.4.5.3.5	DELETE .....	409
10.4.6	Resource: Notification endpoint .....	409
10.4.6.1	Description .....	409
10.4.6.2	Resource definition .....	410
10.4.6.3	Resource methods .....	410
10.4.6.3.1	POST .....	410
10.4.6.3.2	GET .....	410
10.4.6.3.3	PUT .....	411
10.4.6.3.4	PATCH .....	411
10.4.6.3.5	DELETE .....	411
10.5	Data Model .....	411
10.5.1	Introduction .....	411
10.5.2	Resource and notification data types .....	411
10.5.2.1	Introduction .....	411
10.5.2.2	Type: CreateCapacityThresholdRequest .....	411
10.5.2.3	Type: NfviCapacityInfo .....	412
10.5.2.4	Type: NfviCapacityInfoPerZone .....	412
10.5.2.5	Type: NfviCapacityMeasurement .....	412
10.5.2.6	Type: TimeInterval .....	413
10.5.2.7	Type: CapacityThreshold .....	413
10.5.2.8	Type: CapacityThresholdModifications .....	414
10.5.2.9	Type: CapacityShortageNotification .....	414
10.5.3	Referenced structured data types .....	415
10.5.3.1	Introduction .....	415
10.5.3.2	Type: CapacityThresholdCriteria .....	415
10.5.4	Referenced simple data types and enumerations .....	416
10.5.4.1	Introduction .....	416
10.5.4.2	Simple data types .....	416
10.5.4.3	Enumeration: NfviCapacityResourceTypeEnumeration .....	416
11	VNF Snapshot Package Management interface .....	416
11.1	Description .....	416
11.1a	API version .....	417
11.2	Resource structure and methods .....	417
11.3	Sequence diagrams (informative) .....	418
11.3.1	Flow of the creation of an "Individual VNF snapshot package" resource .....	418
11.3.2	Flow of querying/reading VNF snapshot package information .....	419
11.3.3	Flow of updating information of a VNF snapshot package .....	420
11.3.4	Flow of deleting a VNF snapshot package resource .....	421
11.3.5	Flow of the uploading of VNF snapshot package content .....	421
11.3.6	Flow of fetching a VNF snapshot package .....	424
11.3.7	Flow of fetching a VNF snapshot package artifact .....	424
11.3.8	Flow of building the content of VNF snapshot package .....	425
11.3.9	Flow of extracting the content of VNF snapshot package .....	427
11.3.10	Flow of cancelling the operation with the VNF snapshot package content .....	427
11.4	Resources .....	428
11.4.1	Introduction .....	428
11.4.1a	Resource: API versions .....	428
11.4.2	Resource: VNF snapshot packages .....	429
11.4.2.1	Description .....	429
11.4.2.2	Resource definition .....	429
11.4.2.3	Resource methods .....	429
11.4.2.3.1	POST .....	429

11.4.2.3.2	GET .....	430
11.4.2.3.3	PUT .....	431
11.4.2.3.4	PATCH.....	431
11.4.2.3.5	DELETE.....	432
11.4.3	Resource: Individual VNF snapshot package .....	432
11.4.3.1	Description .....	432
11.4.3.2	Resource definition .....	432
11.4.3.3	Resource methods .....	432
11.4.3.3.1	POST .....	432
11.4.3.3.2	GET .....	432
11.4.3.3.3	PUT .....	433
11.4.3.3.4	PATCH.....	433
11.4.3.3.5	DELETE.....	434
11.4.4	Resource: VNF snapshot package content.....	434
11.4.4.1	Description .....	434
11.4.4.2	Resource definition .....	434
11.4.4.3	Resource methods .....	435
11.4.4.3.1	POST .....	435
11.4.4.3.2	GET .....	435
11.4.4.3.3	PUT .....	436
11.4.4.3.4	PATCH.....	437
11.4.4.3.5	DELETE.....	437
11.4.5	Resource: Upload VNF snapshot package from URI task.....	437
11.4.5.1	Description .....	437
11.4.5.2	Resource definition .....	437
11.4.5.3	Resource methods .....	438
11.4.5.3.1	POST .....	438
11.4.5.3.2	GET .....	439
11.4.5.3.3	PUT .....	439
11.4.5.3.4	PATCH.....	439
11.4.5.3.5	DELETE.....	439
11.4.6	Resource: Build VNF snapshot package content task.....	439
11.4.6.1	Description .....	439
11.4.6.2	Resource definition .....	439
11.4.6.3	Resource methods .....	439
11.4.6.3.1	POST .....	439
11.4.6.3.2	GET .....	441
11.4.6.3.3	PUT .....	441
11.4.6.3.4	PATCH.....	441
11.4.6.3.5	DELETE.....	441
11.4.7	Resource: Extract VNF snapshot package content task.....	441
11.4.7.1	Description .....	441
11.4.7.2	Resource definition .....	441
11.4.7.3	Resource methods .....	442
11.4.7.3.1	POST .....	442
11.4.7.3.2	GET .....	443
11.4.7.3.3	PUT .....	443
11.4.7.3.4	PATCH.....	443
11.4.7.3.5	DELETE.....	443
11.4.8	Resource: Cancel VNF snapshot package operation task.....	443
11.4.8.1	Description .....	443
11.4.8.2	Resource definition .....	443
11.4.8.3	Resource methods .....	444
11.4.8.3.1	POST .....	444
11.4.8.3.2	GET .....	445
11.4.8.3.3	PUT .....	445
11.4.8.3.4	PATCH.....	445
11.4.8.3.5	DELETE.....	445
11.4.9	Resource: Access configuration for external artifacts.....	445
11.4.9.1	Description .....	445
11.4.9.2	Resource definition .....	445
11.4.9.3	Resource methods .....	446

11.4.9.3.1	POST .....	446
11.4.9.3.2	GET .....	446
11.4.9.3.3	PUT .....	446
11.4.9.3.4	PATCH .....	447
11.4.9.3.5	DELETE .....	447
11.4.10	Resource: Individual VNF snapshot package artifact .....	447
11.4.10.1	Description .....	447
11.4.10.2	Resource definition .....	448
11.4.10.3	Resource methods .....	448
11.4.10.3.1	POST .....	448
11.4.10.3.2	GET .....	448
11.4.10.3.3	PUT .....	450
11.4.10.3.4	PATCH .....	450
11.4.10.3.5	DELETE .....	450
11.5	Data model .....	450
11.5.1	Introduction .....	450
11.5.2	Resource and notification data types .....	450
11.5.2.1	Introduction .....	450
11.5.2.2	Type: CreateVnfSnapshotPkgInfoRequest .....	450
11.5.2.3	Type: VnfSnapshotPkgInfo .....	451
11.5.2.4	Type: VnfSnapshotPkgInfoModifications .....	453
11.5.2.5	Type: UploadVnfSnapshotPkgFromUriRequest .....	453
11.5.2.6	Type: BuildVnfSnapshotPkgRequest .....	454
11.5.2.7	Type: ExtractVnfSnapshotPkgRequest .....	455
11.5.2.8	Type: CancelVnfSnapshotPkgOperationRequest .....	455
11.5.2.9	Type: VnfSnapshotPkgExtArtifactsAccessConfig .....	456
11.5.3	Referenced structured data types .....	456
11.5.3.1	Introduction .....	456
11.5.3.2	Type: VnfcSnapshotImageInfo .....	457
11.5.3.3	Type: SnapshotPkgArtifactInfo .....	458
11.5.3.4	Type: VnfdInfo .....	459
11.5.3.5	Type: VnfSnapshotRecord .....	460
11.5.4	Referenced simple data types and enumerations .....	460
11.6	VNF snapshot package state model and error handling .....	460
11.6.1	Introduction .....	460
11.6.2	States and state transitions .....	460
11.6.3	Handling of errors .....	463
11.6.3.1	Introduction .....	463
11.6.3.2	Failure during uploading of the VNF snapshot package .....	463
11.6.3.3	Failure during building of the VNF snapshot package .....	463
11.6.3.4	Failure during processing of the VNF snapshot package .....	464
11.6.3.5	Failure during extraction of the VNF snapshot package .....	464
12	NS LCM Coordination interface .....	464
12.1	Description .....	464
12.1a	API version .....	465
12.2	Resource structure and methods .....	465
12.3	Sequence diagrams (informative) .....	466
12.3.1	Flow of LCM coordination .....	466
12.4	Resources .....	469
12.4.1	Introduction .....	469
12.4.1a	Resource: API versions .....	469
12.4.2	Resource: Coordinations .....	469
12.4.2.1	Description .....	469
12.4.2.2	Resource definition .....	469
12.4.2.3	Resource methods .....	470
12.4.2.3.1	POST .....	470
12.4.2.3.2	GET .....	471
12.4.2.3.3	PUT .....	472
12.4.2.3.4	PATCH .....	472
12.4.2.3.5	DELETE .....	472
12.4.3	Resource: Individual coordination action .....	472

12.4.3.1	Description .....	472
12.4.3.2	Resource definition .....	472
12.4.3.3	Resource methods .....	472
12.4.3.3.1	POST .....	472
12.4.3.3.2	GET .....	472
12.4.3.3.3	PUT .....	473
12.4.3.3.4	PATCH .....	473
12.4.3.3.5	DELETE .....	473
12.4.4	Resource: Cancel coordination action task .....	473
12.4.4.1	Description .....	473
12.4.4.2	Resource definition .....	473
12.4.4.3	Resource methods .....	474
12.4.4.3.1	POST .....	474
12.4.4.3.2	GET .....	474
12.4.4.3.3	PUT .....	474
12.4.4.3.4	PATCH .....	475
12.4.4.3.5	DELETE .....	475
12.5	Data model .....	475
12.5.1	Introduction .....	475
12.5.2	Resource and notification data types .....	475
12.5.2.1	Introduction .....	475
12.5.2.2	Type: LcmCoordRequest .....	475
12.5.2.3	Type: LcmCoord .....	475
12.5.3	Referenced structured data types .....	476
12.5.3.1	Introduction .....	476
12.5.4	Referenced simple data types and enumerations .....	476
12.5.4.1	Introduction .....	476
12.5.4.2	Simple data types .....	477
12.5.4.3	Enumeration: LcmOperationForCoordType .....	477
12.5.4.4	Enumeration: LcmCoordResultType .....	477
<b>Annex A (informative): Mapping operations to protocol elements.....</b>		<b>478</b>
A.1	Overview .....	478
A.2	NSD Management interface .....	478
A.3	NS Lifecycle Management interface .....	479
A.4	NS Performance Management interface.....	479
A.5	NS Fault Management interface.....	480
A.6	VNF Package Management interface.....	480
A.7	NFVI Capacity Information interface .....	480
A.8	VNF Snapshot Package Management interface .....	481
A.9	NS LCM Coordination interface .....	481
<b>Annex B (informative): State models.....</b>		<b>482</b>
B.1	NSD state model.....	482
B.1.1	Introduction .....	482
B.1.2	State model.....	482
B.2	VNF package state model.....	483
B.2.1	Introduction .....	483
B.2.2	State model.....	483
<b>Annex C (informative): Complementary material for API utilization .....</b>		<b>486</b>
<b>Annex D (informative): History of features added to the present document.....</b>		<b>487</b>
D.1	Overview .....	487

D.2	Features added in Release 3 .....	487
D.2.1	FEAT02: VNF Software modification .....	487
D.2.2	FEAT04: Host reservation.....	487
D.2.3	FEAT10: Multi-site connectivity services.....	487
D.2.4	FEAT15: VNF snapshotting.....	488
D.2.5	Additional new functionality outside the "NFV features" scheme .....	488
D.2.5.1	Trunking support .....	488
D.2.5.2	Verbosity of VNF LCM operation occurrence notifications .....	489
D.2.5.3	Support for virtual IP connection points .....	489
D.2.5.4	NS LCM coordination .....	489
D.2.6	FEAT05: Network slicing .....	490
D.3	Features added in Release 4 .....	490
D.3.1	FEAT17: Cloud-Native VNFs and Container Infrastructure management .....	490
D.3.2	FEAT21: NFV support for 5G .....	491
D.3.3	ENH02: Special technical enhancements .....	492
D.3.3.1	ENH02.02: NFV-MANO enhancement for NS feasibility check.....	492
D.3.3.2	ENH02.03: NFV-MANO enhancement with Data flow mirroring.....	493
D.3.3.3	ENH02.04: Invariant identification of NSD constituents .....	493
D.3.3.4	ENH02.05: Flexibility with scalable VNF/NS instantiation.....	494
<b>Annex E (informative): Guidelines and examples for the use of ETSI GS NFV SOL 005 APIs for multi-site service connectivity.....</b>		<b>495</b>
E.1	Overview and concepts description.....	495
E.1.1	Introduction .....	495
E.1.2	NFVI-PoP network gateway management models.....	495
E.1.3	Types of connectivity service endpoint .....	498
E.2	Description and usage guidelines of API attributes .....	500
E.2.1	Description of API attributes.....	500
E.2.1.1	WanConnectionData.....	500
E.2.1.2	SiteToWanLayer2ProtocolData .....	501
E.2.1.3	SiteToWanLayer3ProtocolData .....	502
E.2.2	Use of API attributes based on different management models.....	502
E.2.2.1	Introduction.....	502
E.2.2.2	Specific usage for externally managed WAN and NFVI-PoP network provider-managed NFVI-PoP network gateway (case wim.2+gw.2) .....	502
E.2.2.3	Specific usage for externally managed WAN and WAN provider-managed NFVI-PoP network gateway (case wim.2+gw.1) .....	503
E.2.2.4	Specific usage for NFV-MANO managed WAN and NFVI-PoP network provider-managed NFVI-PoP network gateway (case wim.1+gw.2).....	504
E.2.2.5	Specific usage for NFV-MANO managed WAN and WAN provider-managed NFVI-PoP network gateway (case wim.1+gw.1) .....	505
E.2.2.6	Specific usage for externally managed WAN and co-managed NFVI-PoP network gateway (case wim.2+gw.3).....	505
E.2.2.7	Specific usage for NFV-MANO managed WAN and co-managed NFVI-PoP network gateway (case wim.1+gw.3).....	506
<b>Annex F (informative): Bibliography.....</b>		<b>508</b>
<b>Annex G (informative): Change History .....</b>		<b>509</b>
History .....		522

---

## Intellectual Property Rights

### Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<https://ipr.etsi.org/>).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

### Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

---

## Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV).

---

## Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.



---

# 1 Scope

The present document specifies a set of RESTful protocol specifications and data models fulfilling the requirements specified in ETSI GS NFV-IFA 013 [3] for the interfaces used over the Os-Ma-nfvo reference point.

---

## 2 References

### 2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <https://docbox.etsi.org/Reference/>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

[1] IANA: "Assigned Internet Protocol Numbers".

NOTE: Available at <https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>.

[2] ETSI GS NFV-IFA 010: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Functional requirements specification".

[3] ETSI GS NFV-IFA 013: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Os-Ma-nfvo reference point - Interface and Information Model Specification".

[4] ETSI GS NFV-SOL 003: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; RESTful protocols specification for the Or-Vnfm Reference Point".

[5] ETSI GS NFV-SOL 004: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; VNF Package and PNFD Archive specification".

[6] IEEE 802.1Q™-2018: "IEEE Standard for Local and Metropolitan Area Networks - Bridges and Bridged Networks".

[7] IETF RFC 791: "Internet Protocol".

NOTE: Available at <https://tools.ietf.org/html/rfc791>.

[8] IETF RFC 4776: "Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information".

NOTE: Available at <https://tools.ietf.org/html/rfc4776>.

[9] IETF RFC 5646: "Tags for Identifying Languages".

NOTE: Available at <https://www.rfc-editor.org/info/rfc5646>.

[10] IETF RFC 7233: "Hypertext Transfer Protocol (HTTP/1.1): Range Requests".

NOTE: Available at <https://tools.ietf.org/html/rfc7233>.

[11] IETF RFC 7396: "JSON Merge Patch".

NOTE: Available at <https://tools.ietf.org/html/rfc7396>.

- [12] IETF RFC 8200: "Internet Protocol, Version 6 (IPv6) Specification".  
NOTE: Available at <https://tools.ietf.org/html/rfc8200>.
- [13] ISO 3166 (all parts): "Codes for the representation of names of countries and their subdivisions".
- [14] Recommendation ITU-T X.733: "Information technology - Open Systems Interconnection - Systems Management: Alarm reporting function".
- [15] ETSI GS NFV-IFA 027: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Performance Measurements Specification".
- [16] ETSI GS NFV-SOL 013: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; Specification of common aspects for RESTful NFV MANO APIs".
- [17] IETF RFC 7193: "The application/cms Media Type".  
NOTE: Available at <https://tools.ietf.org/html/rfc7193>.
- [18] ETSI GS NFV-SOL 007: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; Network Service Descriptor File Structure Specification".
- [19] ETSI GS NFV-IFA 007: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification".
- [20] IETF RFC 7946: "The GeoJSON Format".  
NOTE: Available at <https://tools.ietf.org/html/rfc7946>.
- [21] IETF RFC 7432: "BGP MPLS-Based Ethernet VPN".  
NOTE: Available at <https://tools.ietf.org/html/rfc7432>.
- [22] IETF RFC 4761: "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling".  
NOTE: Available at <https://tools.ietf.org/html/rfc4761>.
- [23] IETF RFC 6624: "Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling".  
NOTE: Available at <https://tools.ietf.org/html/rfc6624>.
- [24] IETF RFC 4762: "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling".  
NOTE: Available at <https://tools.ietf.org/html/rfc4762>.
- [25] IETF RFC 6074: "Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)".  
NOTE: Available at <https://tools.ietf.org/html/rfc6074>.
- [26] IETF RFC 4364: "BGP/MPLS IP Virtual Private Networks (VPNs)".  
NOTE: Available at <https://tools.ietf.org/html/rfc4364>.
- [27] IETF RFC 7232: "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests".  
NOTE: Available at <https://tools.ietf.org/html/rfc7232>.
- [28] Void.
- [29] ETSI GS NFV-IFA 011: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; VNF Descriptor and Packaging Specification".

[30] IETF RFC 8214: "Virtual Private Wire Service Support in Ethernet VPN".

NOTE: Available at <https://tools.ietf.org/html/rfc8214>.

## 2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GS NFV-SOL 001: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; NFV descriptors based on TOSCA specification".

[i.2] OpenStack®: "Disk and container formats for images".

NOTE 1: Available from <https://docs.openstack.org/glance/pike/user/formats.html>.

NOTE 2: The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. ETSI is not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

[i.3] OpenAPI™ Specification.

NOTE: Available at <https://github.com/OAI/OpenAPI-Specification>.

[i.4] Void.

[i.5] Void.

[i.6] Void.

[i.7] ETSI GS NFV-SOL 015: "Network Functions Virtualisation (NFV); Protocols and Data Models; Specification of Patterns and Conventions for RESTful NFV-MANO APIs".

[i.8] ETSI GS NFV-SOL 010: "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; VNF Snapshot Package specification".

[i.9] ETSI GS NFV-IFA 005: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification".

[i.10] ETSI GS NFV-IFA 032: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Interface and Information Model Specification for Multi-Site Connectivity Services".

[i.11] ETSI GR NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[i.12] Kubernetes® Production-grade Container Orchestration.

NOTE 1: Available at <https://kubernetes.io/>.

NOTE 2: Kubernetes® is a registered trademark of the Linux Foundation in the United States and other countries, and is used pursuant to a license from the Linux Foundation.

[i.13] ETSI GS NFV-SOL 018: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; Profiling specification of protocol and data model solutions for OS Container management and orchestration".

## 3 Definition of terms, symbols and abbreviations

### 3.1 Terms

For the purposes of the present document, the terms given in ETSI GR NFV 003 [i.11] and the following apply:

**Compute MCIO:** MCIO which declarative descriptor specifies compute infrastructure resource requests

**Network MCIO:** MCIO which declarative descriptor specifies network infrastructure resource requests

**Storage MCIO:** MCIO which declarative descriptor specifies storage infrastructure resource requests

### 3.2 Symbols

Void.

### 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GR NFV 003 [i.11] and the following apply:

API	Application Programming Interface
AS	Autonomous System
BFD	Bidirectional Forwarding Detection
BGP	Border Gateway Protocol
BSS	Business Support System
CE	Customer Edge
CIDR	Classless Inter-Domain Routing
CP	Connection Point
CPD	CP Descriptor
CSAR	Cloud Service Archive
CSE	Connectivity Service Endpoint
DC	Data Centre
DF	Deployment Flavour
DSCP	Differentiated Services Code Point
EM	Element Management
ETH	Ethernet
ETSI	European Telecommunications Standards Institute
FM	Fault Management
GS	Group Specification
GUI	Graphical User Interface
GW	Gateway
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IFA	Interfaces and Architecture
IP	Internet Protocol
IPv4	Internet Protocol (version 4)
IPv6	Internet Protocol (version 6)
JSON	JavaScript Object Notation
L2VPN	Level 2 Virtual Private Network
L3VPN	Level 3 Virtual Private Network
LACP	Link Aggregation Control Protocol
LAG	Link Aggregation Group
LB	Load Balancer
LCCN	Lifecycle Change Notification

LCM	Lifecycle Management
MAC	Medium Access Control
MANO	Management and Orchestration
MSCS	Multi-Site Connectivity Service
MTU	Maximum Transmission Unit
NAD	Network Attachment Definition
NFP	Network Forwarding Path
NFPD	NFP Descriptor
NFV	Network Functions Virtualisation
NFVI	Network Function Virtualisation Infrastructure
NFVO	NFV Orchestrator
NS	Network Service
NSD	Network Service Descriptor
NsDf	Network service Deployment flavour
OSPF	Open Shortest Path First
OSS	Operation Support System
PDU	Packet Data Unit
PE	Provider Edge
PM	Performance Management
PNF	Physical Network Function
PNFD	Physical Network Function Descriptor
RAM	Random-Access Memory
RFC	Request For Comments
RIP	Routing Information Protocol
SAP	Service Access Point
SAPD	Service Access Point Descriptor
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOSCA	Topology and Orchestration Specification for Cloud Applications
UC	Use Case
UDP	User Datagram Protocol
UNI	User Network Interface
URI	Uniform Resource Identifier
VDU	Virtualisation Deployment Unit
VIM	Virtualised Infrastructure Manager
VIP	Virtual IP
VL	Virtual Link
VLAN	Virtual Local Area Network
VLD	VL Descriptor
VN	Virtual Network
VNF	Virtualised Network Function
VNFC	VNF Component
VNFD	VNF Descriptor
VNFFG	VNF Forwarding Graph
VNFFGD	VNFFG Descriptor
VNFM	VNF Manager
VNFSR	VNF Snapshot Record
VNI	VxLAN Network Identifier
VPN	Virtual Private Network
VRF	Virtual Routing and Forwarding
VRRP	Virtual Router Redundancy Protocol
VTEP	VxLAN Tunnel Endpoint
VXLAN	Virtual eXtensible LAN
WAN	Wide Area Network
WIM	WAN Infrastructure Manager
YAML	YAML Ain't Markup Language
YANG	Yet Another Next Generation

## 4 General Aspects

### 4.1 Overview

The present document defines the protocol and data model for the following interfaces, in the form of RESTful Application Programming Interface (API) specifications:

- NSD Management interface (as produced by the NFVO towards the OSS/BSS)
- NS Lifecycle Management interface (as produced by the NFVO towards the OSS/BSS)
- NS Performance Management interface (as produced by the NFVO towards the OSS/BSS)
- NS Fault Management interface (as produced by the NFVO towards the OSS/BSS)
- VNF Package Management interface (as produced by the NFVO towards the OSS/BSS)
- NFVI Capacity Information interface (as produced by the NFVO towards the OSS/BSS)
- VNF Snapshot Package Management interface (as produced by the NFVO towards the OSS/BSS)
- NS LCM coordination interface (as produced by the OSS/BSS towards the NFVO)

Table 4.1-1 lists the versions of the APIs defined in the present document.

**Table 4.1-1: Versions of the APIs specified in the present document**

API	API version
NSD Management interface	2.10.0
NS Lifecycle Management interface	2.10.0
NS Performance Management interface	2.10.0
NS Fault Management interface	1.10.0
VNF Package Management interface	2.10.0
NFVI Capacity Information interface	1.10.0
VNF Snapshot Package Management interface	1.10.0
NS LCM Coordination interface	1.10.0

The design of the protocol and data model for the above interfaces is based on the information model and requirements defined in ETSI GS NFV-IFA 013 [3]. In clause 4, general aspects are specified that apply to multiple APIs on the Os-Ma-nfvo reference point. In addition, the provisions in clauses 4, 5, 6, 8 and 9 of ETSI GS NFV-SOL 013 [16] define common aspects of RESTful NFV MANO APIs, and shall apply for all APIs defined in the present document.

In the subsequent clauses, the protocol and data model for the individual interfaces are specified. Per interface, the resource structure with associated HTTP methods is defined and applicable flows are provided. Further, the resources and the data model are specified in detail.

Annex A provides the mapping of the combination of resources and methods defined in the present document to the operations defined in ETSI GS NFV-IFA 013 [3].

Even though the various interfaces defined in the present document are related, implementations shall not assume a particular order of messages that arrive via different interfaces.

### 4.2 Void

### 4.3 Void

## 4.4 Common data types

### 4.4.1 Structured data types

#### 4.4.1.1 Introduction

This clause defines data structures that are referenced from data structures in multiple interfaces. In addition, the structured data types defined in clause 7.1 of ETSI GS NFV-SOL 013 [16] shall apply.

#### 4.4.1.2 Void

#### 4.4.1.3 Void

#### 4.4.1.3a Void

#### 4.4.1.4 Void

#### 4.4.1.5 Type: NsInstanceSubscriptionFilter

This type represents subscription filter criteria to match NS instances. It shall comply with the provisions defined in table 4.4.1.5-1.

**Table 4.4.1.5-1: Definition of the NsInstanceSubscriptionFilter data type**

Attribute name	Data type	Cardinality	Description
nsdIds	Identifier	0..N	If present, match NS instances that were created based on an NSD identified by one of the nsdId values listed in this attribute. See note 1.
vnfdIds	Identifier	0..N	If present, match NS instances that contain VNF instances that were created based on a VNFD identified by one of the vnfdId values listed in this attribute. See note 1.
pnfdIds	Identifier	0..N	If present, match NS instances that contain PNFs that are represented by a PNFD identified by one of the pnfdId values listed in this attribute. See note 1.
nsInstanceIds	Identifier	0..N	If present, match NS instances with an instance identifier listed in this attribute. See note 2.
nsInstanceNames	String	0..N	If present, match NS instances with an NS Instance Name listed in this attribute. See note 2.
<p>NOTE 1: The attributes "nsdIds", "vnfdIds" and "pnfdIds" are alternatives to reference to NS instances that are created based on certain NSDs, or contain VNF instances that are based on certain VNFDs, or contain PNFs that are based on certain PNFDs in a filter. They should not be used together in the same filter instance, but one alternative should be chosen.</p> <p>NOTE 2: The attributes "nsInstanceIds" and "nsInstanceNames" are alternatives to reference to particular NS Instances in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.</p>			

#### 4.4.1.6 Type: ResourceHandle

This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM. The ResourceHandle type shall comply with the provisions defined in table 4.4.1.6-1.

Table 4.4.1.6-1: Definition of the ResourceHandle data type

Attribute name	Data type	Cardinality	Description
vimId	Identifier	0..1	Identifier of the VIM or CISM or WIM under whose control this resource is placed. This attribute shall be supported when the resource is managed by a CISM. When the resource is managed by a VIM or WIM, this attribute shall be present if VNF-related resource management in direct mode is applicable. It shall also be present for resources that are part of an NS instance such as virtual link resources.
resourceProviderId	Identifier	0..1	Identifier of the entity responsible for the management of the resource. This attribute shall only be supported and present when VNF-related resource management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
resourceId	IdentifierInVim	1	Identifier of the resource in the scope of the VIM, the CISM, the WIM or the resource provider. See note 2.
vimLevelResourceType	String	0..1	Type of the resource in the scope of the VIM, the CISM, the WIM or the resource provider. See note 1.
vimLevelAdditionalResourceInfo	AdditionalResourceInfo	0..1	Additional resource information which is specific to this resource and its type, and which is available from the VIM or the CISM or the resource provider.
containerNamespace	String	0..1	The value of the namespace in which the MCIO corresponding to the resource is deployed.  This attribute shall be present if the resource is managed by a CISM and it shall be absent otherwise.
NOTE 1: The value set of the "vimLevelResourceType" attribute is within the scope of the VIM, the CISM, the WIM or the resource provider and can be used as information that complements the ResourceHandle. When the container infrastructure service is a Kubernetes® instance the vimLevelResourceType is the type of resource, as would correspond to the 'kind' field if the resource is declared in its own Kubernetes® manifest, e.g. Pod, PersistentVolumeClaim, NetworkAttachmentDefinition.			
NOTE 2: When the container infrastructure service is a Kubernetes® instance the resourceId shall be populated in the following way: <ul style="list-style-type: none"> <li>– For a compute MCIO, it is the instance identifier that Kubernetes® assigns, which is unique cluster wide per resource type.</li> <li>– For a storage MCIO modelled as a persistent volume claim, it is the name of the persistent volume claim, i.e. the value of the 'claimName' field in the Kubernetes® manifest, or a compound name built by Kubernetes® if the persistent volume claim is defined inline in another template instead of in its own manifest.</li> <li>– For a network MCIO representing a NetworkAttachmentDefinition, a Service or an Ingress, it is the value of the 'metadata.name' field in Kubernetes® manifest.</li> </ul>			

#### 4.4.1.7 Type: AdditionalResourceInfo

This type represents additional resource information which resource and resource type specific, and which is available from the VIM or the CISM or the resource provider. It shall comply with the provisions defined in table 4.4.1.7-1.

Table 4.4.1.7-1: Definition of the AdditionalResourceInfo data type

Attribute name	Data type	Cardinality	Description
hostName	String	0..1	Name of the host where the resource is allocated. It shall be present for compute resources in the scope of the CISM and shall be absent otherwise. See note.
persistentVolume	String	0..1	Name of the persistent volume to which the persistent volume claim representing the storage resource is bound. It may be present for storage resources in the scope of the CISM and shall be absent otherwise. See note.
additionalInfo	KeyValuePairs	0..1	Information related to other properties directly owned by the resource and available from the VIM or CISM or WIM or the resource provider. See note.
NOTE: At least one attribute shall be present.			



## 4.4.2 Simple data types and enumerations

### 4.4.2.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in multiple interfaces.

### 4.4.2.2 Simple data types

Table 4.4.2.2-1 defines simple data types for reference from data type definitions in the present document. In addition, the simple data types defined in clause 7.2.2 of ETSI GS NFV-SOL 013 [16] shall apply.

**Table 4.4.2.2-1: Simple data types**

Type name	Description
IdentifierInNs	An identifier that is unique with respect to an NS. Representation: string of variable length.
IdentifierInNsd	An identifier that is unique within an NS descriptor. Representation: string of variable length.
IdentifierInPnf	An Identifier that is unique within respect to a PNF. Representation: string of variable length.
IdentifierInVim	An identifier maintained by the VIM or the CISM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.

### 4.4.2.3 Enumerations

The enumerations defined in clause 7.2.3 of ETSI GS NFV-SOL 013 [16] shall apply to be available for referencing from data type definitions in the present document.

## 4.5 HTTP conditional requests

Conditional requests are HTTP (POST, PUT, PATCH and DELETE) requests that include one or more header fields indicating a precondition to be tested before applying the method semantics to the target resource. They are supported by meta-information about the resource that was provided in earlier HTTP responses. The set of HTTP header fields to compose a conditional request allowed in the present version of the present document is defined in clause 4.2 of ETSI GS NFV-SOL 013 [16] with the following provisions applicable to the HTTP requests and responses:

- POST:** For resources that also support the PUT or PATCH method, the API producer should provide the "ETag" and the "Last-Modified" HTTP headers in the POST response when the HTTP response codes are "200 OK", "201 Created" or "204 No Content".
- GET:** For resources that also support the PUT or PATCH method, the API producer should provide the "ETag" and the "Last-Modified" HTTP headers in the GET response when the response code is "200 OK".
- PATCH:** The API producer should provide the "ETag" and the "Last-Modified" HTTP headers in the PATCH response when the response code are "200 OK" or "204 No Content". In case the related "Last-Modified" or "ETag" headers have been received in previous responses related to the target resource, the API consumer should provide the "If-Unmodified-Since" or the "If-Match" header fields as conditions (see sections 3.1 and 3.4 of IETF RFC 7232 [27]) in order to prevent conflicts with other changes to the resource.

When receiving the request, the API producer shall test if there is an "If-Unmodified-Since" or "If-Match" header included. If such header is included, the API producer shall further process the request according to sections 3.1, 3.4 and 4.2 of IETF RFC 7232 [27]. This includes to check if there is a mismatch between the content of that header and the last modification of the target resource and to return the HTTP response code "412 Precondition failed" in such a case, indicating that the resource was modified by another entity since the API consumer has obtained the representation of the resource.

**PUT:** The same provisions as for PATCH apply.

**DELETE:** The API producer shall not provide the "ETag" and the "Last-Modified" HTTP headers in the DELETE response as these headers are not applicable when the resource has ceased to exist.

4.6 Void

4.7 Void

---

## 5 NSD Management interface

### 5.1 Description

This interface allows the OSS/BSS to invoke management operations of NSDs towards the NFVO and to subscribe to notifications related to NSD management changes.

The operations provided through this interface are as follows:

- Create NSD Info
- Upload NSD archive
- Fetch NSD archive
- Update NSD Info
- Delete NSD
- Query NSD Info
- Read NSD
- Fetch NSD archive manifest
- Fetch NSD archive artifact
- Create PNFD Info
- Upload PNFD archive
- Fetch PNFD archive
- Update PNFD Info
- Delete PNFD
- Query PNFD Info
- Read PNFD
- Fetch PNFD archive manifest
- Fetch PNFD archive artifact
- Subscribe
- Terminate Subscription
- Query Subscription Information

- Notify

This interface also enables to invoke error handling procedures (i.e. Retry, Rollback, Continue, Cancel, and Fail) on the actual NS lifecycle management operation occurrences, and API version information retrieval.

The state changes of an NSD are illustrated in clause B.2.

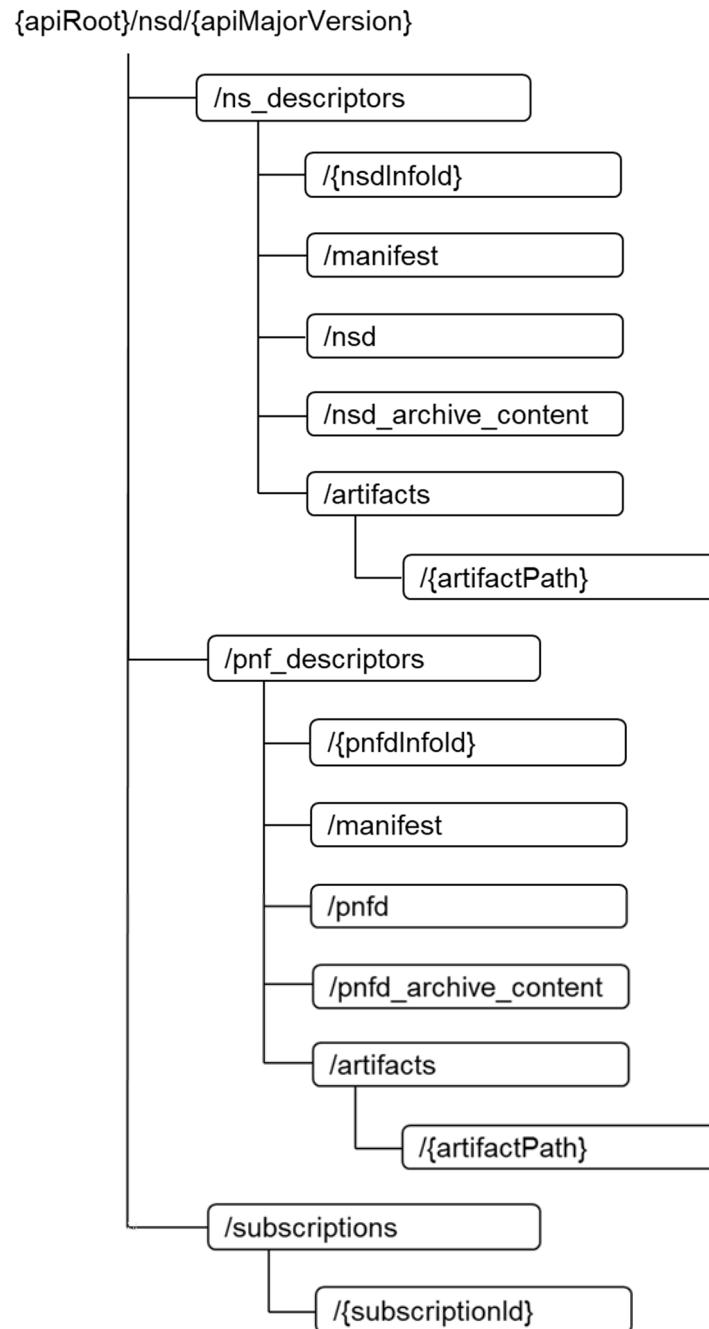
## 5.1a API version

For the NSD management interface version as specified in the present document, the MAJOR version field shall be 2, the MINOR version field shall be 10, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [16] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v2".

## 5.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [16]. The string "nsd" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 5.2-1 shows the overall resource URI structure defined for the NSD management interface.



**Figure 5.2-1: Resource URI structure of NSD Management Interface**

Table 5.2-1 lists the individual resources defined, and the applicable HTTP methods.

The NFVO shall support responding to requests for all HTTP methods on the resources in table 5.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [16].

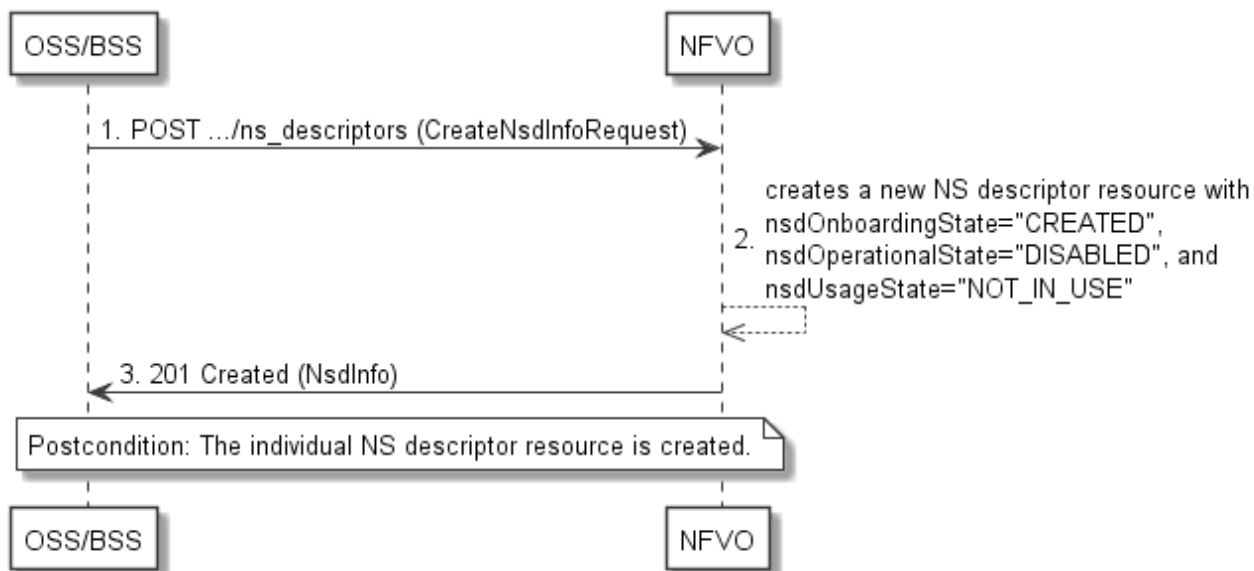
Table 5.2-1: Resources and methods overview of the NSD Management interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
NS descriptors	/ns_descriptors	GET	M	Query information about multiple NS descriptor resources.
		POST	M	Create a new NS descriptor resource.
Individual NS descriptor	/ns_descriptors/{nsdInfold}	GET	M	Read information about an individual NS descriptor resource.
		PATCH	M	Modify the operational state and/or the user defined data of an individual NS descriptor resource.
		DELETE	M	Delete an individual NS descriptor resource.
NSD archive content	/ns_descriptors/{nsdInfold}/nsd_archive_content	GET	M	Fetch the content of an NS descriptor archive.
		PUT	M	Upload the content of an NS descriptor archive.
NSD of an individual NSD archive	/ns_descriptors/{nsdInfold}/nsd	GET	M	Read an onboarded NS descriptor.
Manifest of an individual NSD archive	/ns_descriptors/{nsdInfold}/manifest	GET	M	Fetch the manifest file of an onboarded NSD archive.
Individual NSD archive artifact	/ns_descriptors/{nsdInfold}/artifacts/{artifactPath}	GET	M	Fetch individual NSD archive artifact.
PNF descriptors	/pnf_descriptors	GET	M	Query information about multiple PNF descriptor resources.
		POST	M	Create a new PNF descriptor resource.
Individual PNF descriptor	/pnf_descriptors/{pnfdInfold}	GET	M	Read information about an individual PNF descriptor resource.
		PATCH	M	Modify the user defined data of an individual PNF descriptor resource.
		DELETE	M	Delete an individual PNF descriptor resource.
PNFD archive content	/pnf_descriptors/{pnfdInfold}/pnfd_archive_content	GET	M	Fetch the content of a PNF descriptor archive.
		PUT	M	Upload the content of a PNF descriptor archive.
PNFD of an individual PNFD archive	/pnf_descriptors/{pnfdInfold}/pnfd	GET	M	Read an onboarded PNF descriptor.
Manifest of an individual PNFD archive	/pnf_descriptors/{pnfdInfold}/manifest	GET	M	Fetch the manifest file of an onboarded PNFD archive.
Individual PNFD archive artifact	/pnf_descriptors/{pnfdInfold}/artifacts/{artifactPath}	GET	M	Fetch individual PNFD archive artifact.
Subscriptions	/subscriptions	POST	M	Subscribe to NS descriptor and PNF descriptor change notifications.
		GET	M	Query multiple subscriptions.
Individual subscription	/subscriptions/{subscriptionId}	GET	M	Read an individual subscription resource.
		DELETE	M	Terminate a subscription.
Notification endpoint	(provided by API consumer)	POST	See note	Notify about NS descriptor and PNF descriptor changes. See note.
		GET	See note	Test the notification endpoint. See note.
NOTE:	The NFVO shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the OSS/BSS. If the OSS/BSS supports invoking the POST method on the "Subscriptions" resource towards the NFVO, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.			

## 5.3 Sequence diagrams (informative)

### 5.3.1 Flow of the creation of an individual NS descriptor resource

This clause describes the procedure for creating an individual NS descriptor resource.



**Figure 5.3.1-1: Flow of the creation of an individual NS descriptor resource**

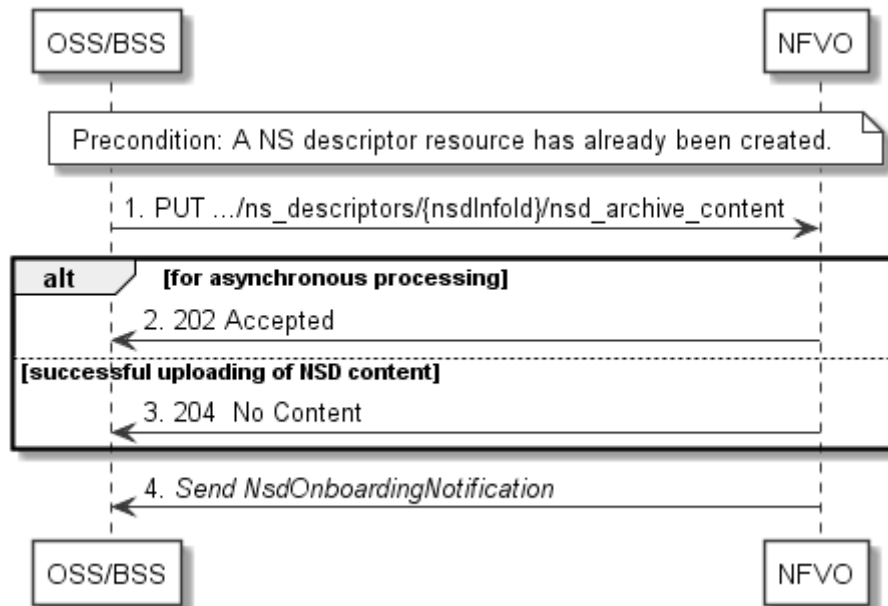
The procedure consists of the following steps as illustrated in figure 5.3.1-1:

- 1) The OSS/BSS sends a POST request to the "NS descriptors" resource including in the payload body a data structure of type "CreateNsdInfoRequest".
- 2) The NFVO creates a new NS descriptor resource with nsdOnboardingState="CREATED", nsdOperationalState="DISABLED" and nsdUsageState="NOT\_IN\_USE".
- 3) The NFVO returns a 201 Created response containing a representation of the "Individual NS descriptor" resource just created by the NFVO.

**Postcondition:** Upon successful completion, the "Individual NS descriptor" resource has been created with nsdOnboardingState="CREATED", nsdOperationalState="DISABLED", and nsdUsageState="NOT\_IN\_USE".

### 5.3.2 Flow of the uploading of NSD archive content

This clause describes the procedure for the uploading of an NSD archive.



**Figure 5.3.2-1: Flow of the uploading of NSD archive content**

**NOTE:** Due to possible race conditions, the 202 or 204 response and the NsdOnBoardingNotification can arrive in any order at the OSS/BSS.

**Precondition:** An NS descriptor resource has been created and the value of "nsdOnboardingState" attribute equals to "CREATED", or the value of "nsdOnboardingState" equals to "ERROR" due to a previous failure of uploading or processing.

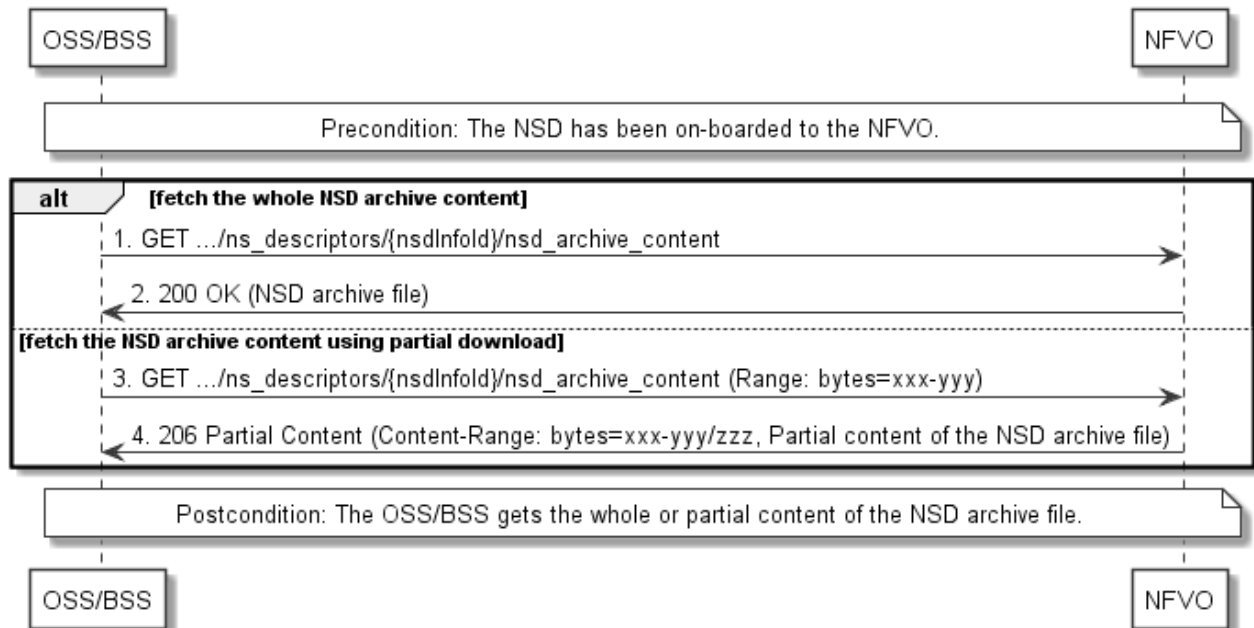
The procedure consists of the following steps as illustrated in figure 5.3.2-1:

- 1) The OSS/BSS sends a PUT request to an "NSD archive content" resource using a "Content-Type" HTTP header as defined in clause 5.4.4.3.3 of the present document.
- 2) For asynchronous processing, the NFVO returns "202 Accepted".
- 3) Otherwise, the NFVO returns a "204 No Content" response to the OSS/BSS with an empty payload body for successful uploading of the NSD content.
- 4) After successful uploading and processing, the NFVO sends an NsdOnboardingNotification to the OSS/BSS.

**Error handling:** In case of failure, appropriate error information is provided in the response, and/or the "nsdOnboardingState" is set to "ERROR".

### 5.3.3 Flow of the fetching of an onboarded NSD archive

This clause describes the procedure for fetching the content of an onboarded NSD archive.



**Figure 5.3.3-1: Flow of the fetching of an NSD archive**

**Precondition:** The NSD is on-boarded to the NFVO.

Fetching an on-boarded NSD, as illustrated in figure 5.3.3-1, consists of the following steps:

- 1) If fetching the whole NSD content, the OSS/BSS sends a GET request to the "NSD archive content" resource.
- 2) The NFVO returns a "200 OK" response, and includes a copy of the complete NSD archive in the payload body.

NOTE: The complete NSD archive is the zip file as onboarded in the Upload NSD archive operation.

- 3) If fetching the NSD content using partial download, the OSS/BSS sends a GET request to the "NSD archive content" resource, and includes a "Range" HTTP header indicating the partition of the NSD content that needs to be transferred.
- 4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the NSD, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the NSD.

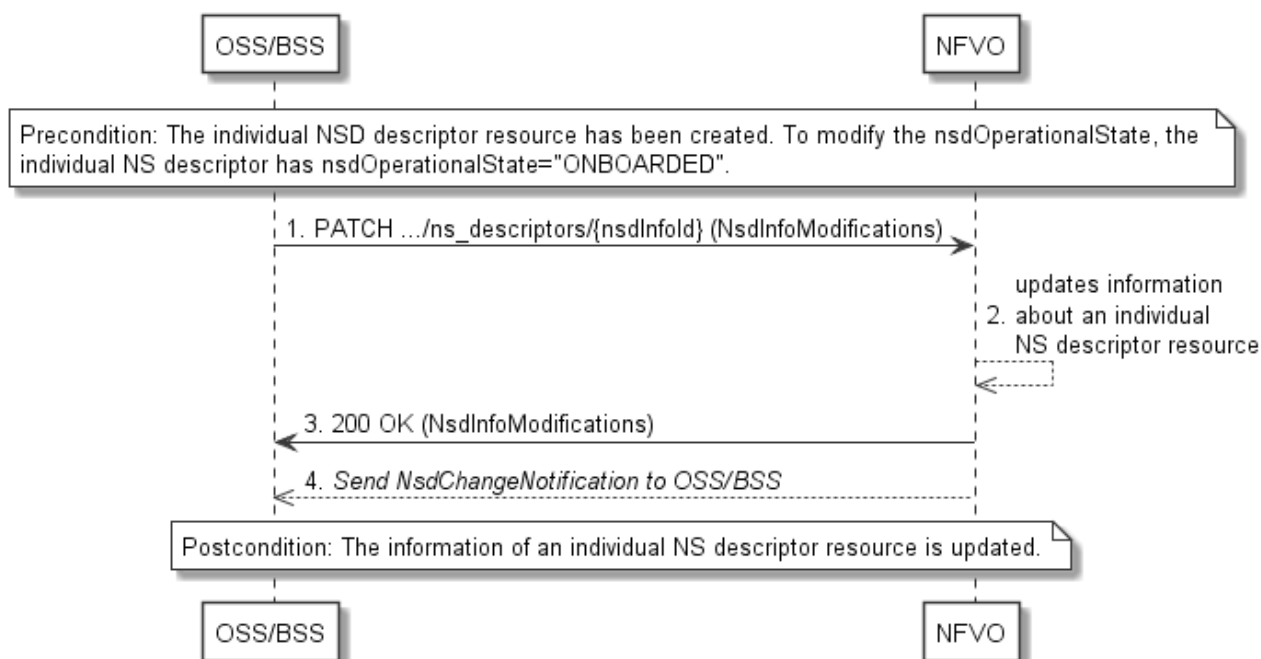
**Postcondition:** Upon successful completion, the OSS/BSS gets the whole or partial content of the NSD.

**Error handling:** In case of failure, appropriate error information is provided in the response.



### 5.3.4 Flow of the update of an individual NS descriptor resource

This clause describes the procedure for the update of an NS descriptor resource. The Update NSD Info operation allows for the modification of the operational state and/or user defined data of an individual NS descriptor resource.



**Figure 5.3.4-1: Flow of the update of an individual NS descriptor resource**

**NOTE:** Due to possible race conditions, the 200 response and the NsdChangeNotification can arrive in any order at the OSS/BSS.

**Precondition:** The individual NS descriptor resource has been created. To modify the nsdOperationalState from "ENABLED" to "DISABLED" or vice-versa, the individual NS descriptor resource has nsdOnboardingState="ONBOARDED".

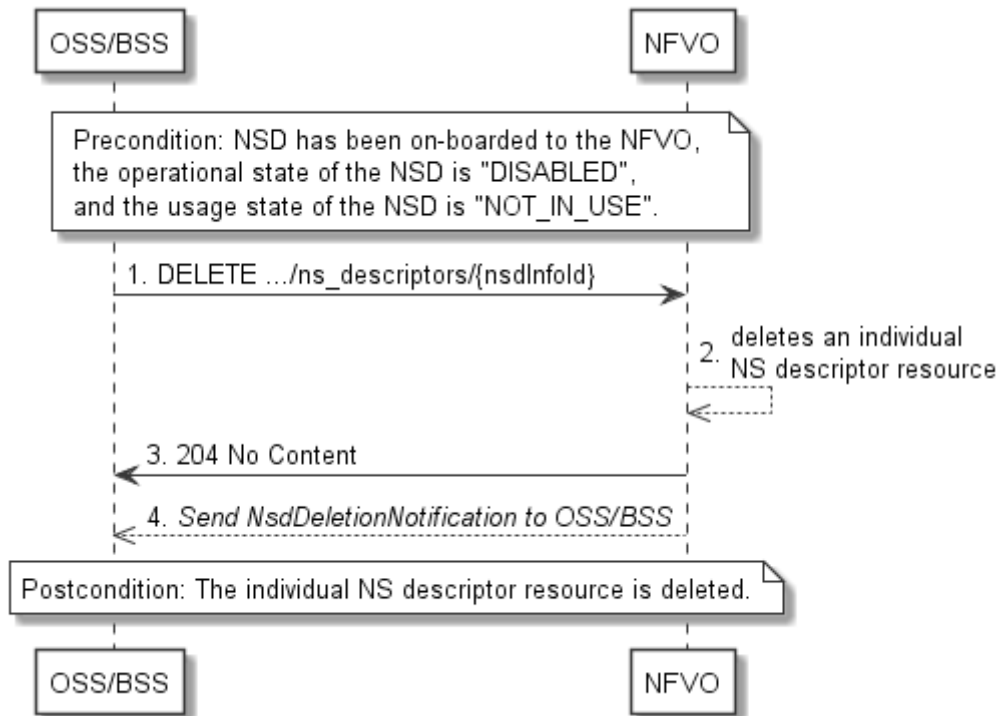
The procedure consists of the following steps as illustrated in figure 5.3.4-1:

- 1) The OSS/BSS sends a PATCH request to the "Individual NS descriptor" resource.
- 2) The NFVO modifies the information associated with the "Individual NS descriptor" resource.
- 3) The NFVO returns a "200 OK" response including the data structure of type "NsdInfoModifications" in the payload body.
- 4) When modifying the nsdOperationalState attribute, the NFVO sends to the OSS/BSS an NsdChangeNotification to indicate the state change of the "Individual NS descriptor" resource.

**Postcondition:** Upon successful completion, the information about an "Individual NS descriptor" resource has been updated.

### 5.3.5 Flow of the deletion of an individual NS descriptor resource

This clause describes the procedure for the deletion of an individual NS descriptor resource.



**Figure 5.3.5-1: Flow of the deletion of an individual NS descriptor resource**

**NOTE:** Due to possible race conditions, the 204 response and the NsdDeletionNotification can arrive in any order at the OSS/BSS.

**Precondition:** NSD has been on-boarded to the NFVO, the operational state of the NSD is equal to "DISABLED", and the usage state of the NSD is equal to "NOT\_IN\_USE".

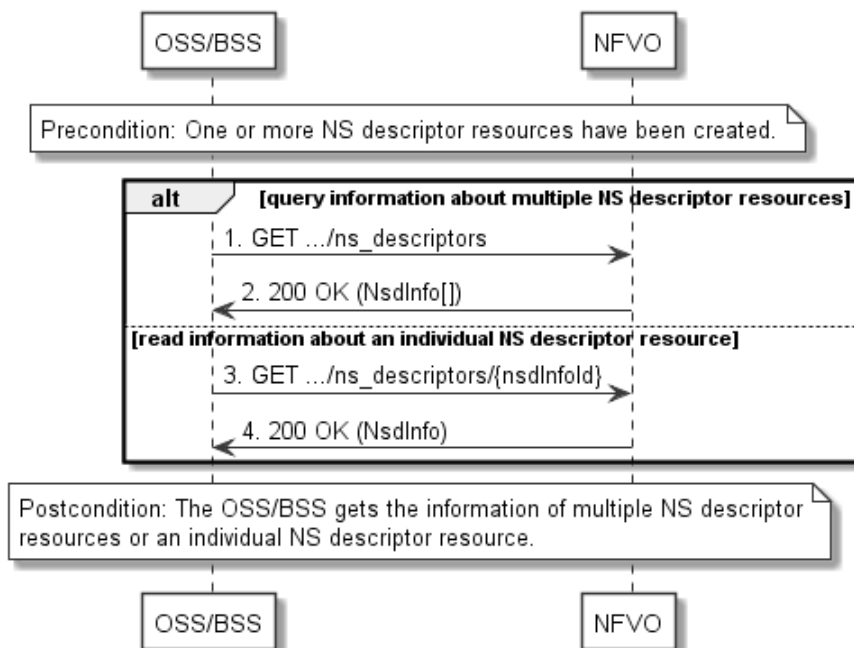
The procedure consists of the following steps as illustrated in figure 5.3.5-1:

- 1) The OSS/BSS sends a DELETE request to an "Individual NS descriptor" resource.
- 2) The NFVO deletes the "Individual NS descriptor" resource.
- 3) The NFVO returns a "204 No Content" response to the OSS/BSS with an empty payload body.
- 4) The NFVO sends to the OSS/BSS an NsdDeletionNotification to indicate the deletion of the "Individual NS descriptor" resource.

**Postcondition:** Upon successful completion, the "Individual NS descriptor" resource has been deleted.

### 5.3.6 Flow of the querying/reading of NS descriptor information

This clause describes the procedure for querying information about multiple NS descriptor resources and reading information about an individual NS descriptor resource.



**Figure 5.3.6-1: Flow of the querying/reading of NS descriptor resources**

**Precondition:** One or more NS descriptor resources have been created.

The procedure consists of the following steps as illustrated in figure 5.3.6-1:

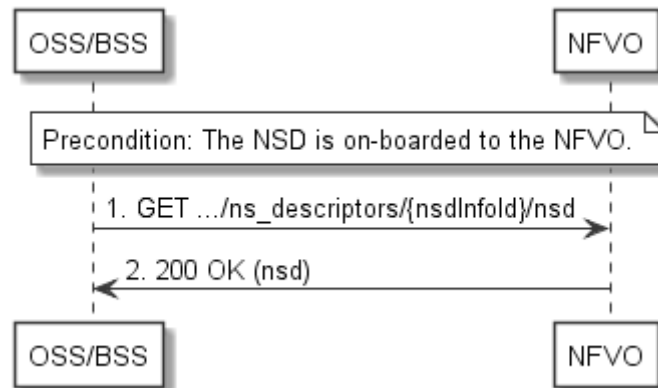
- 1) If the OSS/BSS intends to query information about multiple NS descriptor resources, it sends a GET request to the "NS descriptors" resource.
- 2) The NFVO returns a "200 OK" response and includes in the payload body zero or more data structures of type "NsdInfo".
- 3) If the OSS/BSS intends to read information about an "Individual NS descriptor" resource, the OSS/BSS sends a GET request to the "Individual NS descriptor" resource, addressed by the appropriate NsdInfo identifier in its resource URI.
- 4) The NFVO returns a "200 OK" response, and includes in the payload body a data structure of type "NsdInfo".

**Postcondition:** Upon successful completion, the OSS/BSS gets the information of multiple (i.e. zero or more) NS descriptor resources or an "Individual NS descriptor" resource.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 5.3.6a Flow of reading the NSD

This clause describes the procedure for reading an onboarded NSD.



**Figure 5.3.6a-1: Flow of reading an NSD**

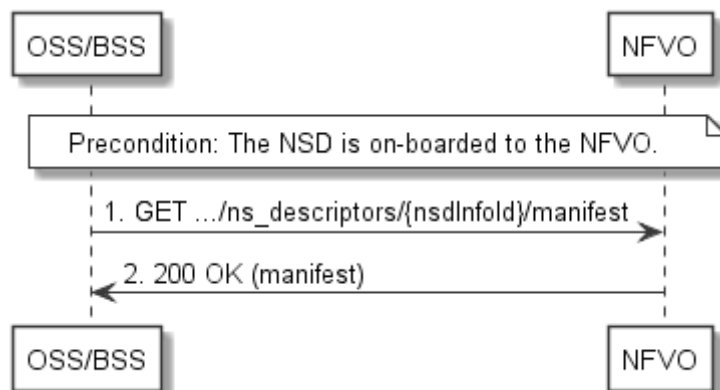
**Precondition:** The NSD is onboarded to the NFVO.

The procedure consists of the following steps as illustrated in figure 5.3.6a-1:

- 1) The OSS/BSS sends a GET request to the "NSD of an individual NSD archive" resource.
- 2) The NFVO returns a "200 OK" response and includes a copy of the NSD in the payload body.

### 5.3.6b Flow of fetching the manifest file of an onboarded NSD archive

This clause describes the procedure for fetching the manifest file of an onboarded NSD.



**Figure 5.3.6b-1: Flow of fetching the manifest file of an onboarded NSD archive**

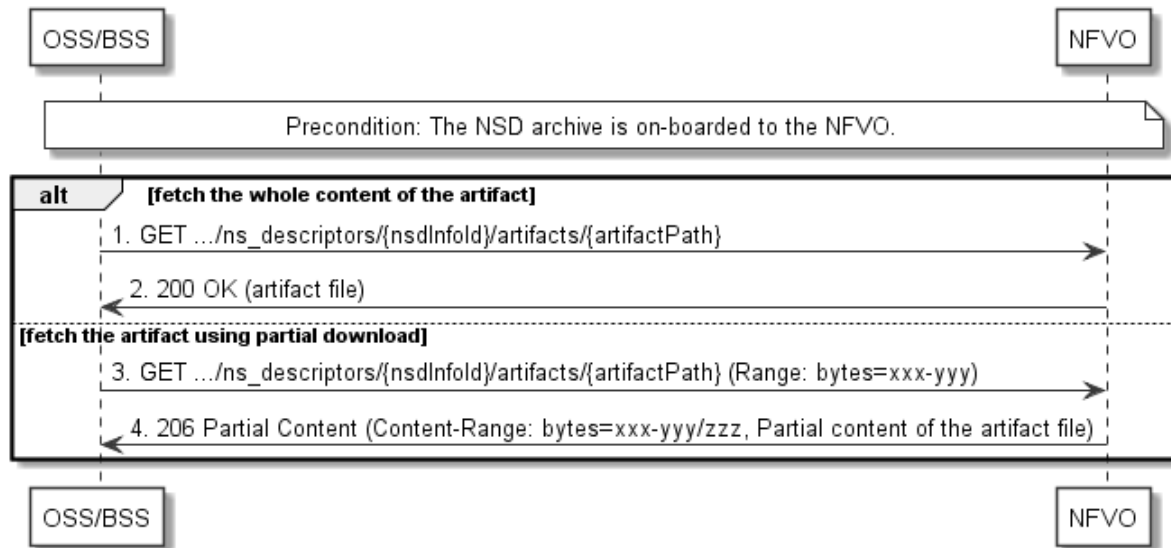
**Precondition:** The NSD is onboarded to the NFVO.

The procedure consists of the following steps as illustrated in figure 5.3.6b-1:

- 1) The OSS/BSS sends a GET request to the "Manifest of an individual NSD archive" resource.
- 2) The NFVO returns a "200 OK" response and includes a copy of the manifest file in the payload body.

### 5.3.6c Flow of fetching an NSD archive artifact

This clause describes the procedure for fetching an individual artifact contained in an on-boarded NSD archive.



**Figure 5.3.6c-1: Flow of fetching an NSD archive artifact**

**Precondition:** The NSD archive is on-boarded to the NFVO.

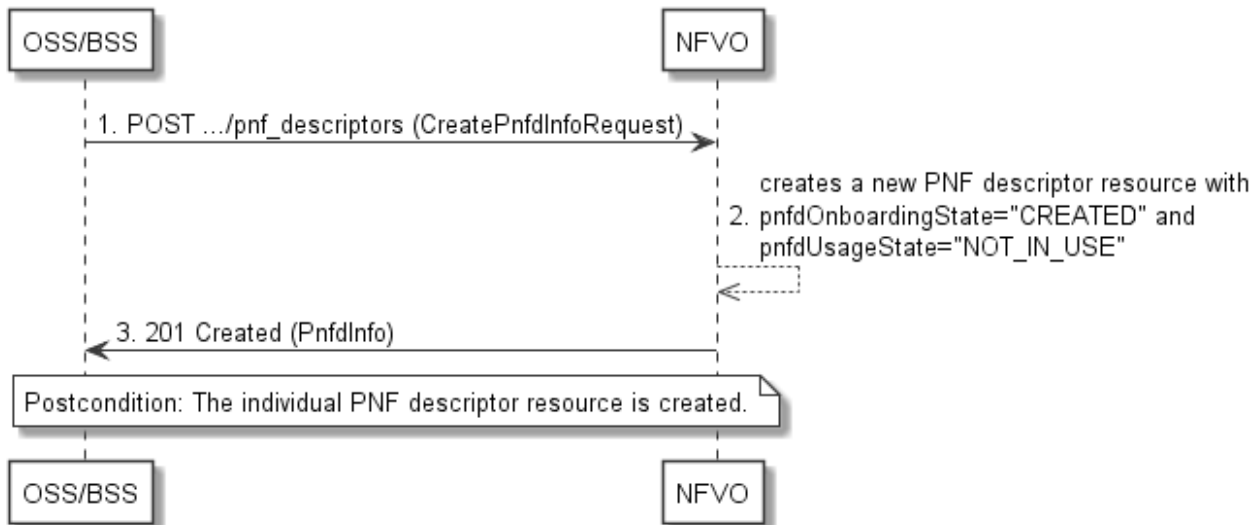
Fetching an individual artifact contained in an on-boarded NSD archive, as illustrated in figure 5.3.6c-1, consists of the following steps:

- 1) If fetching the whole content of the artifact, the OSS/BSS sends a GET request to the "Individual NSD archive artifact" resource.
- 2) The NFVO returns a "200 OK" response, and includes a copy of the applicable artifact file from the NSD archive in the payload body.
- 3) If fetching the artifact using partial download, the OSS/BSS sends a GET request to the "Individual NSD archive artifact" resource and includes a "Range" HTTP header indicating the partition of the artifact needs to be transferred.
- 4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the artifact file, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the artifact file.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 5.3.7 Flow of the creation of an individual PNF descriptor resource

This clause describes the procedure for creating an individual PNF descriptor resource.



**Figure 5.3.7-1: Flow of the creation of an individual PNF descriptor resource**

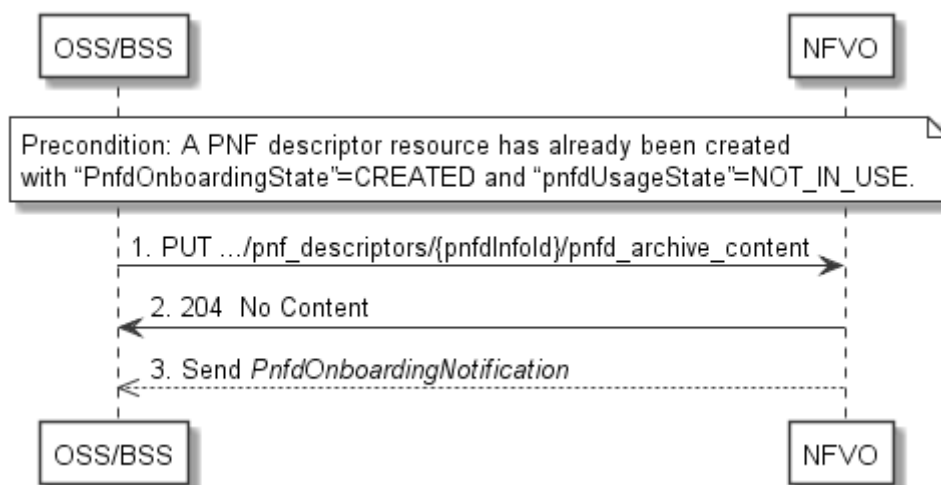
The procedure consists of the following steps as illustrated in figure 5.3.7-1:

- 1) The OSS/BSS sends a POST request to the "PNF descriptors" resource including in the payload body a data structure of type "CreatePnfdInfoRequest".
- 2) The NFVO creates a new PNF descriptor resource with `pnfdOnboardingState="CREATED"` and `pnfdUsageState="NOT_IN_USE"`.
- 3) The NFVO returns a 201 Created response containing a representation of the "Individual PNF descriptor" resource just created by the NFVO.

**Postcondition:** Upon successful completion, the "Individual PNF descriptor" resource has been created with `pnfdOnboardingState="CREATED"` and `pnfdUsageState="NOT_IN_USE"`.

### 5.3.8 Flow of the uploading of PNFD archive

This clause describes the procedure for the uploading of a PNFD archive.



**Figure 5.3.8-1: Flow of the uploading of PNFD archive content**

**Precondition:** A PNF descriptor resource has been created and the value of the "pnfdOnboardingState" attribute equals to "CREATED" or the value of the "pnfdOnboardingState" attribute equals to "ERROR" due to a previous failure of uploading or processing.

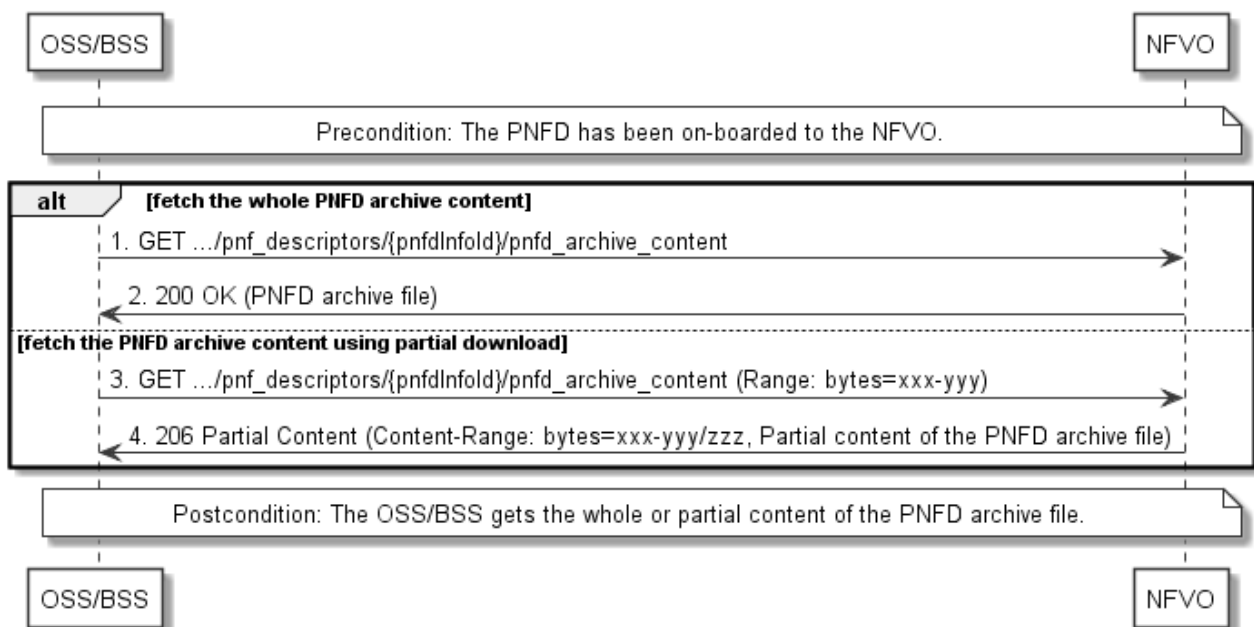
The procedure consists of the following steps as illustrated in figure 5.3.8-1:

- 1) The OSS/BSS sends a PUT request to a "PNFD archive content" resource using a "Content-Type" HTTP header as defined in clause 5.4.4.3.3 of the present document.
- 2) The NFVO returns a "204 No Content" response to the OSS/BSS with an empty payload body for successful uploading of the PNFD content.
- 3) After successful uploading and processing, the NFVO sends a PnfdOnboardingNotification to the OSS/BSS.

**Error handling:** In case of failure, appropriate error information is provided in the response, and/or the "pnfdOnboardingState" is set to "ERROR".

### 5.3.9 Flow of the fetching of an onboarded PNFD archive

This clause describes the procedure for fetching the content of an onboarded PNFD archive.



**Figure 5.3.9-1: Flow of the fetching of PNFD archive content**

**Precondition:** The PNFD has been on-boarded to the NFVO.

Fetching an on-boarded PNFD, as illustrated in figure 5.3.9-1, consists of the following steps:

- 1) The OSS/BSS sends a GET request to the "PNFD archive content" resource.
- 2) The NFVO returns a "200 OK" response and includes a copy of the complete PNFD archive in the payload body.

**NOTE:** The complete PNFD archive is the zip file as onboarded in the Upload PNFD archive operation.

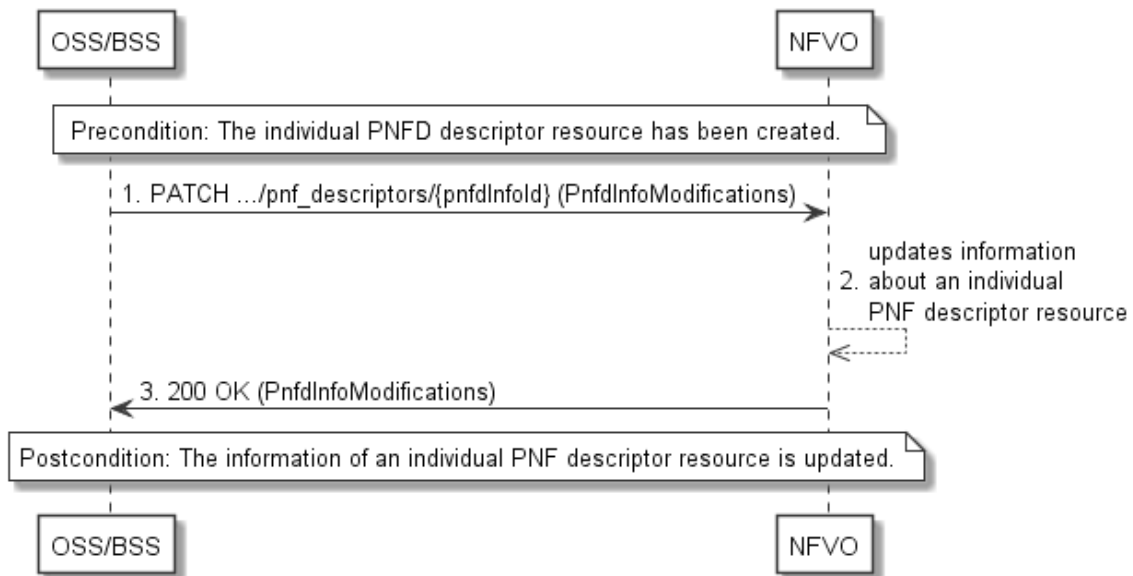
- 3) If fetching the PNFD archive using partial download, the OSS/BSS sends a GET request to the "PNFD archive content" resource and includes a "Range" HTTP header indicating the partition of the PNFD archive content that needs to be transferred.
- 4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the PNFD archive, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the PNFD archive content.

**Postcondition:** Upon successful completion, the OSS/BSS gets the content of the PNF.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 5.3.9a Flow of the update of an individual PNF descriptor resource

This clause describes the procedure for the update of a PNF descriptor resource. The Update PNF Info operation allows for the modification of the operational state and/or user defined data of an individual PNF descriptor resource.



**Figure 5.3.9a-1: Flow of the update of an individual PNF descriptor resource**

**NOTE:** Due to possible race conditions, the 200 response and the PnfdChangeNotification can arrive in any order at the OSS/BSS.

**Precondition:** The individual PNF descriptor resource has been created. To modify the `pnfdOperationalState` from "ENABLED" to "DISABLED" or vice-versa, the individual PNF descriptor resource has `pnfdOnboardingState="ONBOARDED"`.

The procedure consists of the following steps as illustrated in figure 5.3.9a-1:

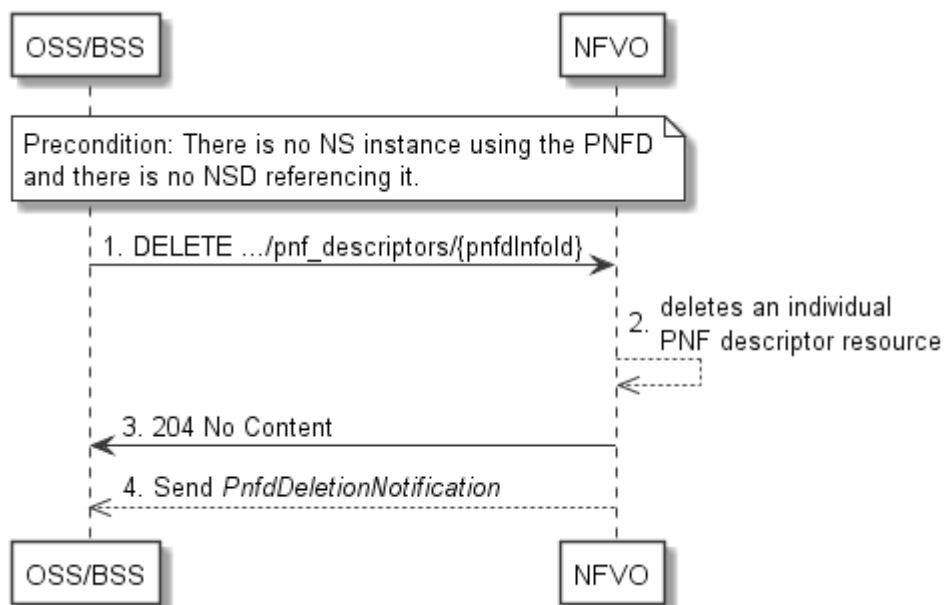
- 1) The OSS/BSS sends a PATCH request to the "Individual PNF descriptor" resource.
- 2) The NFVO modifies the information associated with the "Individual PNF descriptor" resource.
- 3) The NFVO returns a "200 OK" response including the data structure of type "PnfdInfoModifications" in the payload body.

**Postcondition:** Upon successful completion, the information about an "Individual PNF descriptor" resource has been updated.



### 5.3.10 Flow of the deletion of an individual PNF descriptor resource

This clause describes the procedure for the deletion of an individual PNF descriptor resource.



**Figure 5.3.10-1: Flow of the deletion of an individual PNF descriptor resource**

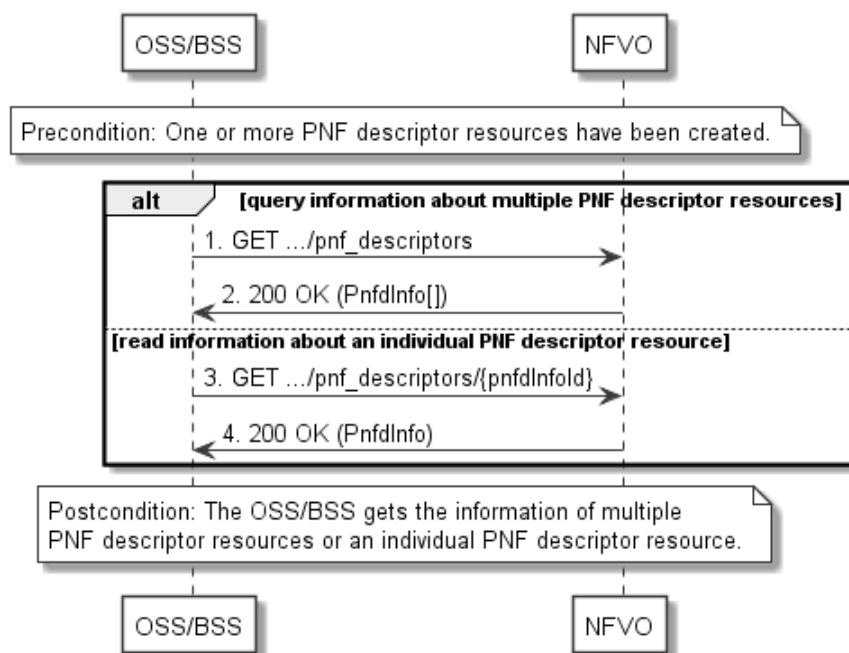
**Precondition:** NSD has been on-boarded to the NFVO and in the Disabled state.

The procedure consists of the following steps as illustrated in figure 5.3.10-1:

- 1) The OSS/BSS sends a DELETE request to an "Individual PNF descriptor" resource.
- 2) The NFVO deletes the "Individual PNF descriptor" resource.
- 3) The NFVO returns a "204 No Content" response to the OSS/BSS with an empty payload body.
- 4) The NFVO sends to the OSS/BSS a `PnfdDeletionNotification` to indicate the deletion of the "Individual PNF descriptor" resource.

### 5.3.11 Flow of the querying/reading of PNF descriptor information

This clause describes the procedure for querying information about multiple PNF descriptor resources and reading information about an individual PNF descriptor resource.



**Figure 5.3.11-1: Flow of the querying/reading of PNF descriptor resources**

**Precondition:** One or more PNF descriptor resources have been created.

The procedure consists of the following steps as illustrated in figure 5.3.11-1:

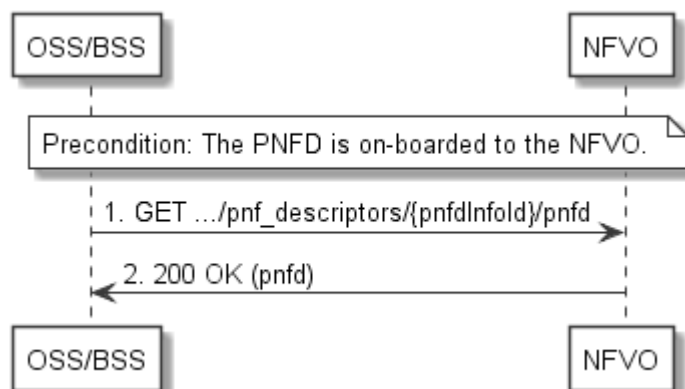
- 1) If the OSS/BSS intends to query information about multiple PNF descriptor resources, it sends a GET request to the "PNF descriptors" resource.
- 2) The NFVO returns a "200 OK" response, and includes in the payload body zero or more data structures of type "PnfdInfo".
- 3) If the OSS/BSS intends to read information about an "Individual PNF descriptor" resource, the OSS/BSS sends a GET request to the "Individual PNF descriptor" resource, addressed by the appropriate PnfdInfo identifier in its resource URI.
- 4) The NFVO returns a "200 OK" response, and includes in the payload body a data structure of type "PnfdInfo".

**Postcondition:** Upon successful completion, the OSS/BSS gets the information of multiple (i.e. zero or more) PNF descriptor resources or an "Individual PNF descriptor" resource.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 5.3.11a Flow of reading the PNFD

This clause describes the procedure for reading an onboarded PNFD.



**Figure 5.3.11a-1: Flow of reading a PNFD**

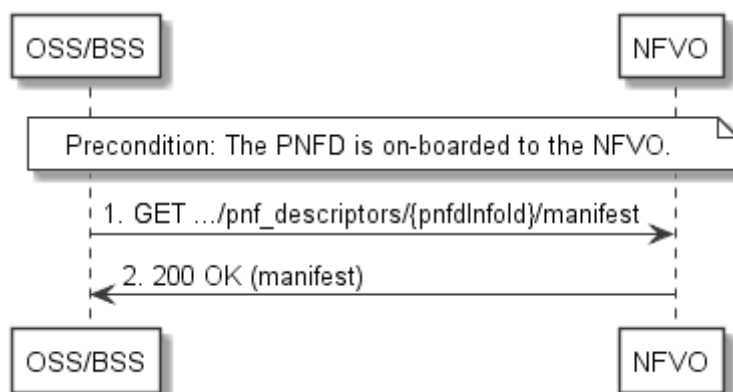
**Precondition:** The PNFD is onboarded to the NFVO.

The procedure consists of the following steps as illustrated in figure 5.3.11a-1:

- 1) The OSS/BSS sends a GET request to the "PNFD of an individual PNFD archive" resource.
- 2) The NFVO returns a "200 OK" response and includes a copy of the PNFD in the payload body.

### 5.3.11b Flow of fetching the manifest file of an onboarded PNFD archive

This clause describes the procedure for fetching the manifest file of an onboarded PNFD archive.



**Figure 5.3.11b-1: Flow of fetching the manifest file of an onboarded PNFD archive**

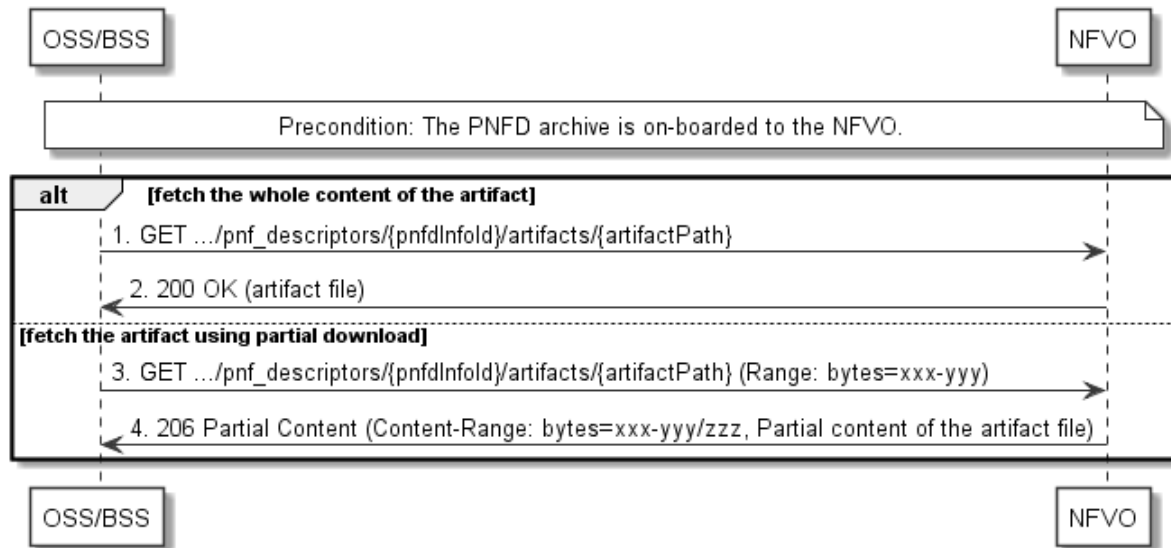
**Precondition:** The PNFD is onboarded to the NFVO.

The procedure consists of the following steps as illustrated in figure 5.3.11b-1:

- 1) The OSS/BSS sends a GET request to the "Manifest of an individual PNFD archive" resource.
- 2) The NFVO returns a "200 OK" response and includes the manifest file in the payload body.

### 5.3.11c Flow of fetching a PNFD archive artifact

This clause describes the procedure for fetching an individual artifact contained in an on-boarded PNFD archive.



**Figure 5.3.11c-1: Flow of fetching a PNFD archive artifact**

**Precondition:** The PNFD archive is on-boarded to the NFVO.

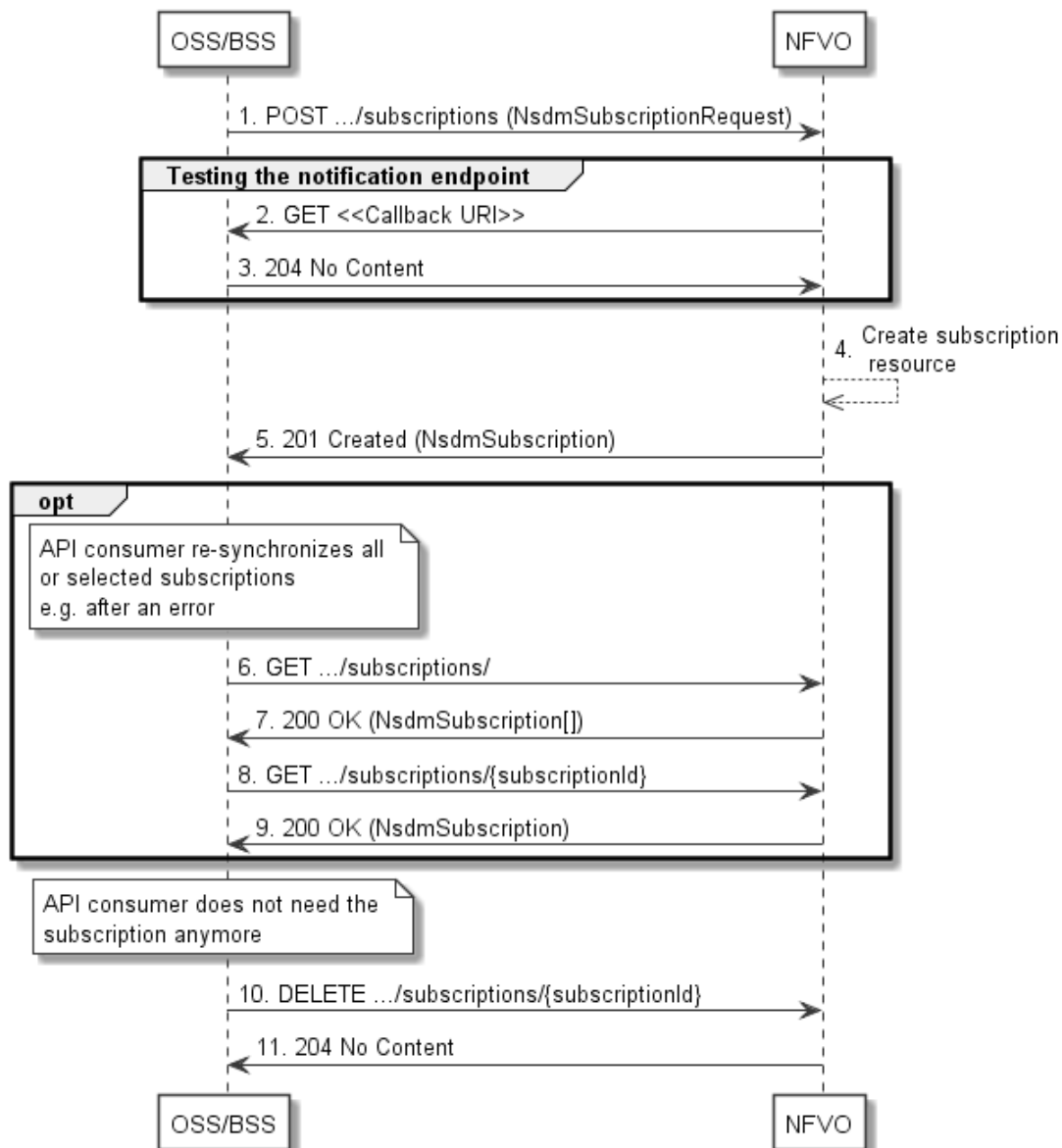
Fetching an individual artifact contained in an on-boarded PNFD archive, as illustrated in figure 5.3.11c-1, consists of the following steps:

- 1) If fetching the whole content of the artifact, the OSS/BSS sends a GET request to the "Individual PNFD archive artifact" resource.
- 2) The NFVO returns a "200 OK" response, and includes a copy of the applicable artifact file from the PNFD archive in the payload body.
- 3) If fetching the artifact using partial download, the OSS/BSS sends a GET request to the "Individual PNFD archive artifact" resource, and includes a "Range" HTTP header indicating the partition of the artifact needs to be transferred.
- 4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the artifact file, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the artifact file.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 5.3.12 Flow of managing subscriptions

This clause describes the procedure for creating, reading and terminating subscriptions to notifications related to NSD management.



**Figure 5.3.12-1: Flow of managing subscriptions**

The procedure consists of the following steps as illustrated in figure 5.3.12-1:

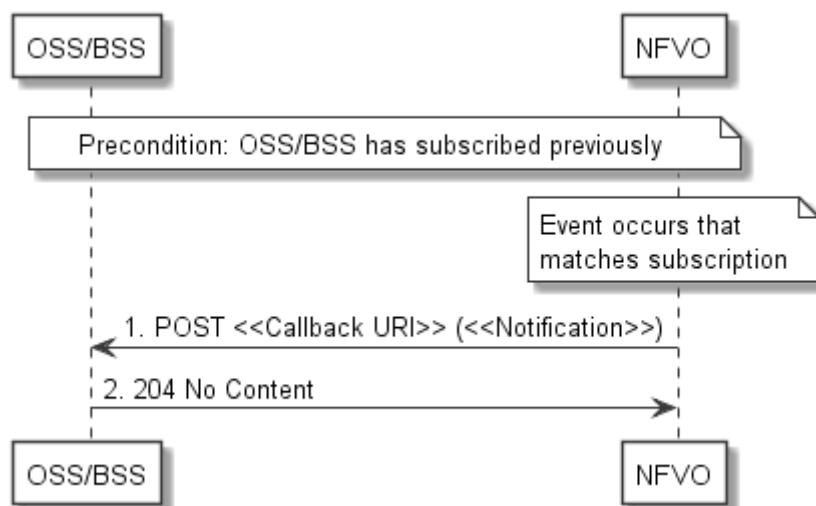
- 1) The OSS/BSS sends a POST request to the "Subscriptions" resource including in the payload body a data structure of type "NsdmSubscriptionRequest". This data structure contains filtering criteria and a callback URI to which the NFVO will subsequently send notifications about events that match the filter.
- 2) To test the notification endpoint that has been registered by the OSS/BSS as part of the subscription, the NFVO sends a GET request to the notification endpoint URI.
- 3) The OSS/BSS returns a "204 No Content" response to indicate success.
- 4) The NFVO creates a new subscription for notifications related to NS fault management, and a resource that represents this subscription.
- 5) The NFVO returns a "201 Created" response containing a data structure of type "NsdmSubscription", representing the subscription resource just created by the NFVO, and provides the URI of the newly-created resource in the "Location" HTTP header.
- 6) If desired, e.g. to recover from an error situation, the OSS/BSS can query information about its subscriptions by sending a GET request to the "Subscriptions" resource.

- 7) In that case, the NFVO returns a "200 OK" response that contains the list of representations of all existing subscriptions that were created by the NFVO.
- 8) If desired, e.g. to recover from an error situation, the OSS/BSS can read information about a particular subscription by sending a GET request to the resource representing that individual subscription.
- 9) In that case, the NFVO returns a "200 OK" response that contains a representation of that individual subscription.
- 10) When the OSS/BSS does not need the subscription anymore, it terminates the subscription by sending a DELETE request to the resource that represents the individual subscription.
- 11) The NFVO acknowledges the successful termination of the subscription by returning a "204 No Content" response.

**Error handling:** The NFVO rejects a subscription if the subscription information is not valid: endpoint cannot be reached, subscription information is malformed, etc.

### 5.3.13 Flow of sending notifications

This clause describes the procedure for sending notifications related to NSD management.



**Figure 5.3.13-1: Flow of sending notifications**

**Precondition:** The OSS/BSS has subscribed previously for notifications related to NSD management.

The procedure consists of the following steps as illustrated in figure 5.3.13-1:

- 1) If an event occurs that matches the filtering criteria which are part of the subscription, the NFVO generates a notification that includes information about the event, and sends it in the body of a POST request to the URI which the NFVO has registered as part of the subscription request. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API (see clauses 5.5.2.9, 5.5.2.10, 5.5.2.11, 5.5.2.12, 5.5.2.13, 5.5.2.14 and 5.5.2.15).
- 2) The OSS/BSS acknowledges the successful delivery of the notification by returning a "204 No Content" response.

**Error handling:** If the NFVO does not receive the "204 No Content" response from the OSS/BSS, it can retry sending the notification.

## 5.4 Resources

### 5.4.1 Introduction

This clause defines all the resource and methods provided by the NSD management interface.

The on-boarding of an NSD includes:

- 1) Creation of an individual NS descriptor resource.
- 2) Uploading the NSD content.
- 3) Validation of the NSD inside the NFVO.

In the present document, the NSD is referred to as "on-boarded" only after these three procedures are successfully accomplished.

NOTE: Annex B describes the state model of NSD in the NFVO. It includes the state models for two phases, i.e. onboarding phase of NSD and operational phase of NSD.

Further, the on-boarding of a PNFD includes:

- 1) Creation of an individual PNF descriptor resource.
- 2) Uploading the PNFD.
- 3) Processing the PNFD, including validation, inside the NFVO.

A PNFD is referred as "on-boarded" only after these three procedures are successfully accomplished.

#### 5.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [16] are part of the NSD management interface.

### 5.4.2 Resource: NS Descriptors

#### 5.4.2.1 Description

This resource represents NS descriptors. It can be used to create an individual NS descriptor resource, and to query multiple NS descriptor resources.

#### 5.4.2.2 Resource definition

The resource URI is:

**{apiRoot}/nsd/{apiMajorVersion}/ns\_descriptors**

This resource shall support the resource URI variables defined in table 5.4.2.2-1.

**Table 5.4.2.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.

### 5.4.2.3 Resource methods

#### 5.4.2.3.1 POST

The POST method is used to create a new NS descriptor resource.

This method shall follow the provisions specified in the tables 5.4.2.3.1-1 and 5.4.2.3.1-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.2.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 5.4.2.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	CreateNsdInfoRequest	1	Parameters of creating an NS descriptor resource, as defined in clause 5.5.2.3.	
Response body	Data type	Cardinality	Response Codes	Description
	NsdInfo	1	201 Created	<p>Shall be returned when a new "Individual NS descriptor" resource and the associated NS descriptor identifier has been created successfully.</p> <p>The response body shall contain a representation of the created NS descriptor resource, as defined in clause 5.5.2.2.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created NS descriptor resource.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 5.4.2.3.2 GET

The GET method queries information about multiple NS descriptor resources.

This method shall follow the provisions specified in the tables 5.4.2.3.2-1 and 5.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.2.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this filtering parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>All attribute names that appear in the NsdInfo and in data types referenced from it shall be supported by the NFVO in the filter expression.</p>
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.



Name	Cardinality	Description
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.  The following attributes shall be excluded from the NsdInfo structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: <ul style="list-style-type: none"> <li>- userDefinedData</li> <li>- onboardingFailureDetails</li> </ul>
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 5.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	NsdInfo	0..N	200 OK	Shall be returned when information about zero or more NS descriptors has been queried successfully.  The response body shall contain in an array the representations of zero or more NS descriptors, as defined in clause 5.5.2.2.  If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression.  The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute selector.  The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big.  If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 5.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.3 Resource: Individual NS Descriptor

#### 5.4.3.1 Description

This task resource represents an individual NS descriptor. The API consumer can use this resource to modify, delete and read the information of the individual NS descriptor.

#### 5.4.3.2 Resource definition

The resource URI is:

**{apiRoot}/nsd/{apiMajorVersion}/ns\_descriptors/{nsdInfoId}**

This resource shall support the resource URI variables defined in table 5.4.3.2-1.

**Table 5.4.3.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
nsdInfoId	Identifier of the individual NS descriptor resource. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new NS descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

#### 5.4.3.3 Resource methods

##### 5.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

##### 5.4.3.3.2 GET

The GET method reads information about an individual NS descriptor.

This method shall follow the provisions specified in the tables 5.4.3.3.2-1 and 5.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.3.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
None supported		

**Table 5.4.3.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	NsdInfo	1	200 OK	Shall be returned when information about the individual NS descriptor has been read successfully.  The response body shall contain a representation of the individual NS descriptor, as defined in clause 5.5.2.2.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 5.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.3.3.4 PATCH

The PATCH method modifies the operational state and/or user defined data of an individual NS descriptor resource.

This method can be used to:

- 1) Enable a previously disabled individual NS descriptor resource, allowing again its use for instantiation of new network service with this descriptor. The usage state (i.e. "IN\_USE/NOT\_IN\_USE") shall not change as a result.
- 2) Disable a previously enabled individual NS descriptor resource, preventing any further use for instantiation of new network service(s) with this descriptor. The usage state (i.e. "IN\_USE/NOT\_IN\_USE") shall not change as a result.
- 3) Modify the user defined data of an individual NS descriptor resource.

This method shall follow the provisions specified in the tables 5.4.3.3.4-1 and 5.4.3.3.4-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.3.3.4-1: URI query parameters supported by the PATCH method on this resource**

Name	Cardinality	Description
None supported		

**Table 5.4.3.3.4-2: Details of the PATCH request/response on this resource**

Request body	Data type	Cardinality	Description	
	NsdInfoModifications	1	Parameters for the modification of an individual NS descriptor resource, as defined in clause 5.5.2.1.  The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [11].	
Response body	Data type	Cardinality	Response Codes	Description
	NsdInfoModifications	1	200 OK	Shall be returned when the operation has been accepted and completed successfully.  The response body shall contain attribute modifications for an "Individual NS descriptor" resource (see clause 5.5.2.1).
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the "Individual NS descriptor" resource.  Typically, this is due to an operational state mismatch, i.e. enable an already enabled or disable an already disabled individual NS descriptor resource, or the "nsdOnboardingState" is not ONBOARDED.  The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled.  Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity.  The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

#### 5.4.3.3.5 DELETE

The DELETE method deletes an individual NS descriptor resource.

An individual NS descriptor resource can only be deleted when there is no NS instance using it (i.e. "nsdUsageState" = "NOT\_IN\_USE"), there is no NSD referring to it which have a value of "strict NSD constituent onboarding" set to True and has been disabled already (i.e. "nsdOperationalState" = "DISABLED"). Otherwise, the DELETE method shall fail.

This method shall follow the provisions specified in the tables 5.4.3.3.5-1 and 5.4.3.3.5-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.3.3.5-1: URI query parameters supported by the DELETE method on this resource**

Name	Cardinality	Description
None supported		

Table 5.4.3.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	<p>Shall be returned when the operation has completed successfully.</p> <p>The response body shall be empty.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact the NS descriptor resource is in the enabled operational state (i.e. nsdOperationalState = ENABLED) or there are running NS instances using the concerned individual NS descriptor resource (i.e. nsdUsageState = IN_USE).</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 5.4.4 Resource: NSD archive content

### 5.4.4.1 Description

This resource represents the content of the individual NSD archive. The API consumer can use this resource to upload and download the content of the NSD.

### 5.4.4.2 Resource definition

The resource URI is:

**{apiRoot}/nsd/{apiMajorVersion}/ns\_descriptors/{nsdInfoId}/nsd\_content**

This resource shall support the resource URI variables defined in table 5.4.4.2-1.

Table 5.4.4.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
nsdInfoId	Identifier of the individual NS descriptor. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new NS descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

### 5.4.4.3 Resource methods

#### 5.4.4.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.4.3.2 GET

The GET method fetches the content of the NSD archive.

The NSD archive is implemented as a single zip file.

The content of the NSD archive is provided as onboarded, i.e. depending on the security option used, the CSAR or the CSAR wrapped in a ZIP archive together with an external signature is returned, as defined in clause 5.1 of ETSI GS NFV-SOL 007 [18].

NOTE: Information about the applicable security option can be obtained by evaluating the "archiveSecurityOption" attribute in the "nsdInfo" structure.

This method shall follow the provisions specified in the tables 5.4.4.3.2-1 and 5.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.4.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 5.4.4.3.2-2: Details of the GET request/response on this resource**

	Data type	Cardinality	Description	
	Request body	n/a		<p>The request may contain a "Range" HTTP header to obtain single range of bytes from the NSD archive. This can be used to continue an aborted transmission.</p> <p>If the Range header is present in the request and the NFVO does not support responding to range requests with a 206 response, it shall return a 200 OK response instead as defined below.</p>
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the content of the NSD has been read successfully.</p> <p>The payload body shall contain a copy of the ZIP file that contains the NSD file structure.</p> <p>The "Content-Type" HTTP header shall be set to "application/zip".</p>
	n/a	1	206 Partial Content	<p>If the NFVO supports range requests, this response shall be returned when a single consecutive byte range from the content of the NSD file has been read successfully according to the request.</p> <p>The response body shall contain the requested part of the NSD archive.</p> <p>The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [10].</p> <p>The "Content-Type" HTTP header shall be set as defined above for the "200 OK" response.</p>
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.

				Typically, this is due to the fact "nsdOnboardingState" has a value different from ONBOARDED.  The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	416 Range Not Satisfiable	The byte range passed in the "Range" header did not match any available byte range in the NSD file (e.g. "access after end of file").  The response body may contain a ProblemDetails structure.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 5.4.4.3.3 PUT

The PUT method is used to upload the content of an NSD archive.

The NSD to be uploaded is implemented as a single ZIP file as defined in clause 5.4.4.3.2.

The "Content-Type" HTTP header in the PUT request shall be set to "application/zip".

Whether all descriptors referenced from the NSD: VNFD, PNFD and NSD for nested NSs shall be on-boarded before this operation depends on the value of "strict NSD constituent onboarding" present in the NSD.

The user may use this operation to upload a new NSD version, which can be associated to an NS instance with the Update NS operation (see clause 6.4.6). Different NSD versions have the same "nsdInvariantId" attribute, but different "nsdId" attributes and different NSD information objects. The design of different NSD versions and their business logic is out of scope of the present document.

After a successful result, the NSD and those referenced descriptors that are already on-boarded are known to and validated by the NFVO. The associated NSD information object is updated with the information populated from the validated NSD. The NSD is on-boarded and is in "nsdOperationalState" = "ENABLED" and "nsdUsageState" = "NOT\_IN\_USE" states, allowing its use for NS lifecycle management. See the NSD state model in clause B.1.2. If there are on-boarded NSDs that refer to this NSD as a nested NSD, their respective NsdInfo records are updated to refer to the on-boarded nested NSD.

This method shall follow the provisions specified in the tables 5.4.4.3.3-1 and 5.4.4.3.3-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.4.3.3-1: URI query parameters supported by the PUT method on this resource**

Name	Cardinality	Description
none supported		

Table 5.4.4.3.3-2: Details of the PUT request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a	1	The payload body contains a ZIP file that represents the NSD archive, as specified above.  The request shall set the "Content-Type" HTTP header to "application/zip".	
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	202 Accepted	Shall be returned when the NSD archive has been accepted for uploading, but the processing has not been completed. It is expected to take some time for processing (asynchronous mode).  The response body shall be empty. See note.
	n/a	1	204 No Content	The NSD content successfully uploaded and validated (synchronous mode).  The response body shall be empty.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.  Typically, this is due to the fact that the "nsdOnboardingState" attribute has a value other than "CREATED" or "ERROR".  The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error "response code" as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	
NOTE: The API consumer can track the uploading progress by receiving the "NsdOnBoardingNotification" and "NsdOnBoardingFailureNotification" from the NFVO or by reading the status of the individual NS descriptor resource using the GET method.				

#### 5.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.4a Resource: NSD

#### 5.4.4a.1 Description

This resource represents the NSD contained in an on-boarded NSD archive. The API consumer can use this resource to obtain the content of the NSD.

#### 5.4.4a.2 Resource definition

The resource URI is:

**{apiRoot}/nsd/{apiMajorVersion}/ns\_descriptors/{nsdInfoId}/nsd**

This resource shall support the resource URI variables defined in table 5.4.4a.2-1.



**Table 5.4.4a.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
nsdInfold	Identifier of the individual NS descriptor. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new NS descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

### 5.4.4a.3 Resource methods

#### 5.4.4a.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.4a.3.2 GET

The GET method reads the content of the NSD within an NSD archive.

The NSD can be implemented as a single file or as a collection of multiple files. If the NSD is implemented in the form of multiple files, a ZIP file embedding these files shall be returned. If the NSD is implemented as a single file, either that file or a ZIP file embedding that file shall be returned.

The selection of the format is controlled by the "Accept" HTTP header passed in the GET request:

- If the "Accept" header contains only "text/plain" and the NSD is implemented as a single file, the file shall be returned; otherwise, an error message shall be returned.
- If the "Accept" header contains only "application/zip", the single file or the multiple files that make up the NSD shall be returned embedded in a ZIP file.
- If the "Accept" header contains both "text/plain" and "application/zip", it is up to the NFVO to choose the format to return for a single-file NSD; for a multi-file NSD, a ZIP file shall be returned.

The default format of the ZIP file shall comply with the CSAR format as specified in ETSI GS NFV-SOL 007 [18] where only the YAML files representing the NSD, and information necessary to navigate the ZIP file and to identify the file that is the entry point for parsing the NSD and (if requested) further security information are included, and other artifacts referenced from the YAML files are excluded. This means that the content of the ZIP archive shall contain the following files from the NSD archive:

- TOSCA.meta (if available in the NSD archive).
- The main TOSCA definitions YAML file (either as referenced by the Entry-Definitions keyword from TOSCA.meta or available as a file with the extension ".yaml" or ".yml" from the root of the archive).
- Other TOSCA YAML files, if any, as referenced by the Other-Definitions keyword from TOSCA.meta.
- Every component of the NSD referenced (recursively) from the YAML files as mentioned above.

NOTE 1: For an NSD based on TOSCA, it includes all the imported type definition files as indicated in the top level service template and in any of the lower level service template if it has any as described in ETSI GS NFV-SOL 001 [i.1].

NOTE 2: For an NSD based on YANG, it includes the file as indicated by the "yang\_definitions" keyname in the metadata section of the main yaml file as described in ETSI GS NFV-SOL 007 [18].

- The related security information, if the "include\_signatures" URI parameter is provided, as follows:
  - the manifest file;
  - the singleton certificate file in the root of the NSD archive (if available in the NSD archive);

- the signing certificates of the individual files included in the ZIP archive (if available in the NSD archive);
- the signatures of the individual files (if available in the NSD archive).

This method shall follow the provisions specified in the tables 5.4.4a.3.2-1 and 5.4.4a.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.4a.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
include_signatures	0..1	<p>If this parameter is provided, the NFVO shall include in the ZIP file the security information as specified above.</p> <p>This URI query parameter is a flag, i.e. it shall have no value.</p> <p>The NFVO shall support this parameter.</p>

**Table 5.4.4a.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a		The request shall contain the appropriate entries in the "Accept" HTTP header as defined above.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the content of the NSD has been read successfully.</p> <p>The payload body shall contain a copy of the file representing the NSD or a ZIP file that contains the file or multiple files representing the NSD, as specified above.</p> <p>The "Content-Type" HTTP header shall be set according to the format of the returned file. It shall be set to "text/plain" for a YAML file.</p>
	ProblemDetails	0..1	406 Not Acceptable	<p>If the "Accept" header does not contain at least one name of a content type for which the NFVO can provide a representation of the NSD, the NFVO shall respond with this response code.</p> <p>The "ProblemDetails" structure may be included with the "detail" attribute providing more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "nsdOnboardingState" has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

### 5.4.4a.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.4a.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.4a.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.4b Resource: NSD archive manifest

#### 5.4.4b.1 Description

This resource represents the manifest file contained in an on-boarded NSD archive. The API consumer can use this resource to obtain the content of the manifest file.

#### 5.4.4b.2 Resource definition

The resource URI is:

**{apiRoot}/nsd/{apiMajorVersion}/ns\_descriptors/{nsdInfoId}/manifest**

This resource shall support the resource URI variables defined in table 5.4.4b.2-1.

**Table 5.4.4b.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
nsdInfoId	Identifier of the individual NS descriptor. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new NS descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

#### 5.4.4b.3 Resource methods

##### 5.4.4b.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

##### 5.4.4b.3.2 GET

The GET method reads the content of the manifest file within an NSD archive.

This method shall follow the provisions specified in the tables 5.4.4b.3.2-1 and 5.4.4b.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.4b.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
include_signatures	0..1	<p>If this parameter is provided, the NFVO shall return the manifest and related security information (signature and certificate) either in a single text file if the signature and certificate are included in the manifest file, or in a zip file containing the manifest and the certificate file, if this is provided as a separate file in the NSD archive.</p> <p>If this parameter is not given, the NFVO shall provide only a copy of the manifest file, as onboarded. If the security information is included in the onboarded manifest, it shall also be included in the returned copy.</p> <p>This URI query parameter is a flag, i.e. it shall have no value.</p> <p>The NFVO shall support this parameter.</p>

Table 5.4.4b.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the content of the manifest file has been read successfully.</p> <p>If the "include_signatures" URI query parameter was absent in the request, or if the manifest file has all security-related information embedded (i.e. there is no separate certificate file), the payload body shall contain a copy of the manifest file of the NSD archive, and the "Content-Type" HTTP header shall be set to "text/plain".</p> <p>If the "include_signatures" URI query parameter was present in the related request and the manifest file does not have all the security-related information embedded (i.e. there is a separate certificate file), the "Content-Type" HTTP header shall be set to "application/zip" and the payload body shall contain a ZIP archive which includes:</p> <ul style="list-style-type: none"> <li>- a copy of the manifest file of the NSD archive;</li> <li>- a copy of the related individual certificate file.</li> </ul>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "nsdOnboardingState" has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 5.4.4b.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.4b.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.4b.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 5.4.4c Resource: Individual NSD archive artifact

### 5.4.4c.1 Description

This resource represents an individual artifact contained in an NSD archive. The API consumer can use this resource to fetch the content of the artifact.

### 5.4.4c.2 Resource definition

The resource URI is:

**{apiRoot}/nsd/{apiMajorVersion}/ns\_descriptors/{nsdInfoId}/artifacts/{artifactPath}**

This resource shall support the resource URI variables defined in table 5.4.4c.2-1.

**Table 5.4.4c.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
nsdInfoId	Identifier of the individual NS descriptor. See note 1.
artifactPath	<p>For an artifact contained as a file in the NSD archive, this variable shall contain a sequence of one or path segments representing the path of the artifact within the NSD archive, relative to the root of the package.</p> <p>EXAMPLE:      foo/bar/m%40ster.sh</p> <p>For an external artifact represented as a URI in the NSD archive manifest, this variable shall contain the URI as provided in the NS archive.</p> <p>See note 2 and note 3.</p>
<p>NOTE 1: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new NS descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.</p> <p>NOTE 2: This identifier can be retrieved from the "artifactPath" attribute of the applicable "artifacts" entry in the body of the response to a GET request querying the "Individual NSD" resource.</p> <p>NOTE 3: Since multiple path segments are allowed to be contained in this variable, the "/" character that separates these segments is not percent-encoded. Each individual segment is percent-encoded if necessary as defined in clause 4.1 of ETSI GS NFV-SOL 013 [16].</p>	

### 5.4.4c.3 Resource methods

#### 5.4.4c.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 5.4.4c.3.2 GET

The GET method fetches the content of an individual artifact within an NSD archive.

This method shall follow the provisions specified in the tables 5.4.4c.3.2-1 and 5.4.4c.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.4c.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
include_signatures	0..1	<p>If this parameter is provided, the NFVO shall return the artifact and related security information (such as signature and optional certificate) in a ZIP archive.</p> <p>If this parameter is not given, the NFVO shall provide only a copy of the artifact file. This URI query parameter is a flag, i.e. it shall have no value.</p> <p>The NFVO shall support this parameter.</p>

**Table 5.4.4c.3.2-2: Details of the GET request/response on this resource**

	Data type	Cardinality	Description	
	Request body	n/a		<p>The request may contain a "Range" HTTP header to obtain single range of bytes from an artifact file. This can be used to continue an aborted transmission.</p> <p>If the Range header is present in the request and the NFVO does not support responding to range requests with a 206 response, it shall return a 200 OK response instead as defined below.</p>
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the whole content of the artifact file has been read successfully.</p> <p>If the "include_signatures" request URI parameter was not provided in the related request, the payload body shall contain a copy of the artifact file from the NSD archive, as defined by ETSI GS NFV-SOL 007 [18], and the "Content-Type" HTTP header shall be set according to the content type of the artifact file. If the artifact is encrypted, the header shall be set to the value "application/cms" (IETF RFC 7193 [17]). If the content type cannot be determined, the header shall be set to the value "application/octet-stream".</p> <p>If the "include_signatures" request URI parameter was provided in the related request, the "Content-Type" HTTP header shall be set to "application/zip" and the payload body shall contain a ZIP archive which includes:</p> <ul style="list-style-type: none"> <li>- a copy of the artifact file from the VNF package, as defined by ETSI GS NFV-SOL 007 [18];</li> <li>- the related security information (individual signature file and optional related individual certificate file).</li> </ul>

	n/a	1	206 Partial Content	<p>If the NFVO supports range requests and the "include_signatures" request URI parameter was not present in the related request, this response shall be returned when a single consecutive byte range from the content of the artifact file has been read successfully according to the request.</p> <p>The response body shall contain the requested part of the artifact file from the NSD archive, as defined by ETSI GS NFV-SOL 007 [18].</p> <p>The "Content-Type" HTTP header shall be set according to the content type of the artifact file. If the content type cannot be determined, the header shall be set to the value "application/octet-stream".</p> <p>The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [10].</p>
	ProblemDetails	0..1	406 Not Acceptable	<p>If the related request contained an "Accept" header not compatible with the Content type "application/zip" but the "include_signatures" flag was provided, the NFVO shall respond with this response code.</p> <p>The "ProblemDetails" structure may be included with the "detail" attribute providing more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "nsdOnboardingState" has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	0..1	416 Range Not Satisfiable	<p>The byte range passed in the "Range" header did not match any available byte range in the artifact file (e.g. "access after end of file").</p> <p>The response body may contain a ProblemDetails structure.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>

#### 5.4.4c.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.4c.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.4c.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 5.4.5 Resource: PNF Descriptors

### 5.4.5.1 Description

This resource represents PNF descriptors and it can be used to create an individual PNF descriptor resource, and to query PNF descriptor resources.

### 5.4.5.2 Resource definition

The resource URI is:

**{apiRoot}/nsd/{apiMajorVersion}/pnf\_descriptors**

This resource shall support the resource URI variables defined in table 5.4.5.2-1.

**Table 5.4.5.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.

### 5.4.5.3 Resource methods

#### 5.4.5.3.1 POST

The POST method is used to create a new PNF descriptor resource.

This method shall follow the provisions specified in the tables 5.4.5.3.1-1 and 5.4.5.3.1-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.5.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 5.4.5.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	CreatePnfdInfoRequest	1	Parameters of creating a PNF descriptor resource, as defined in clause 5.5.2.6.	
Response body	Data type	Cardinality	Response Codes	Description
	PnfdInfo	1	201 Created	<p>Shall be returned when a new "Individual PNF descriptor" resource and the associated PNF descriptor identifier has been created successfully.</p> <p>The response body shall contain a representation of the created PNF descriptor resource, as defined in clause 5.5.2.5.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created PNF descriptor resource.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.



## 5.4.5.3.2 GET

The GET method queries information about multiple PNF descriptor resources.

This method shall follow the provisions specified in the tables 5.4.5.3.2-1 and 5.4.5.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.5.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].  The NFVO shall support receiving this filtering parameter as part of the URI query string. The OSS/BSS may supply this parameter.  All attribute names that appear in the PnfdInfo and in data types referenced from it shall be supported by the NFVO in the filter expression.
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.  The following attributes shall be excluded from the PnfdInfo structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: <ul style="list-style-type: none"> <li>- userDefinedData</li> <li>- onboardingFailureDetails</li> </ul>
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 5.4.5.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
n/a				
Response body	Data type	Cardinality	Response Codes	Description
	PnfdInfo	0..N	200 OK	<p>Shall be returned when information about zero or more PNF descriptors has been queried successfully.</p> <p>The response body shall contain in an array the representations of zero or more PNF descriptors, as defined in clause 5.5.2.5.</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute selector.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 5.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 5.4.6 Resource: Individual PNF Descriptor

### 5.4.6.1 Description

This resource represents an individual PNF descriptor. The API consumer can use this resource to modify, delete and read the information of the individual PNF descriptor resource.

### 5.4.6.2 Resource definition

The resource URI is:

**{apiRoot}/nsd/{apiMajorVersion}/pnf\_descriptors/{pnfdInfoId}**

This resource shall support the resource URI variables defined in table 5.4.6.2-1.

**Table 5.4.6.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
pnfdInfoId	Identifier of the individual PNF descriptor resource. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new PNF descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

### 5.4.6.3 Resource methods

#### 5.4.6.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.6.3.2 GET

The GET method reads information about an individual PNF descriptor.

This method shall follow the provisions specified in the tables 5.4.6.3.2-1 and 5.4.6.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.6.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
None supported		

**Table 5.4.6.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	PnfdInfo	1	200 OK	Shall be returned when information about the individual PNF descriptor has been read successfully.  The response body shall contain a representation of the individual PNF descriptor, as defined in clause 5.5.2.5.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 5.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.6.3.4 PATCH

The PATCH method modifies the user defined data of an individual PNF descriptor resource.

This method shall follow the provisions specified in the tables 5.4.6.3.4-1 and 5.4.6.3.4-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.6.3.4-1: URI query parameters supported by the PATCH method on this resource**

Name	Cardinality	Description
None supported		

**Table 5.4.6.3.4-2: Details of the PATCH request/response on this resource**

Request body	Data type	Cardinality	Description	
	PnfdInfoModifications	1	Parameters for the modification of an individual PNF descriptor resource, as defined in clause 5.5.2.4.  The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [11].	
Response body	Data type	Cardinality	Response Codes	Description
	PnfdInfoModifications	1	200 OK	Shall be returned when the operation has been accepted and completed successfully.  The response body shall contain attribute modifications for an 'Individual PNF Descriptor' resource (see clause 5.5.2.4).
	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled.  Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity.  The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
ProblemDetails	See clause 6.4 of [16]		4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 5.4.6.3.5 DELETE

The DELETE method deletes an individual PNF descriptor resource.

An individual PNF descriptor resource can only be deleted when there is no NS instance using it or there is NSD referencing it which have a value of "strict NSD constituent onboarding" set to True.

To delete all PNF versions identified by a particular value of the "pnfdInvariantId" attribute, the procedure is to first use the GET method with filter "pnfdInvariantId" towards the PNF descriptors resource to find all versions of the PNF. Then, the API consumer uses the DELETE method described in this clause to delete each PNF version individually.

This method shall follow the provisions specified in the tables 5.4.6.3.5-1 and 5.4.6.3.5-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.6.3.5-1: URI query parameters supported by the DELETE method on this resource**

Name	Cardinality	Description
none supported		

**Table 5.4.6.3.5-2: Details of the DELETE request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the operation has completed successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 5.4.7 Resource: PNFD archive content

### 5.4.7.1 Description

This resource represents the content of the individual PNF archive, i.e. PNFD content. The API consumer can use this resource to upload and download the content of the PNFD.

### 5.4.7.2 Resource definition

The resource URI is:

**{apiRoot}/nsd/{apiMajorVersion}/pnf\_descriptors/{pnfdInfoId}/pnfd\_content**

This resource shall support the resource URI variables defined in table 5.4.7.2-1.

**Table 5.4.7.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
pnfdInfoId	Identifier of the individual PNF descriptor. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new PNF descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

### 5.4.7.3 Resource methods

#### 5.4.7.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 5.4.7.3.2 GET

The GET method fetches the content of the PNFD archive.

The content of the PNFD archive is provided as onboarded, i.e. depending on the security option used, the CSAR or the CSAR wrapped in a ZIP archive together with an external signature is returned, as defined in clause 5.1 of ETSI GS NFV-SOL 004 [5].

NOTE: Information about the applicable security option can be obtained by evaluating the "archiveSecurityOption" attribute in the "pnfdInfo" structure.

This method shall follow the provisions specified in the tables 5.4.7.3.2-1 and 5.4.7.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.7.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 5.4.7.3.2-2: Details of the GET request/response on this resource**

	Data type	Cardinality	Description	
Request body	n/a		<p>The request may contain a "Range" HTTP header to obtain single range of bytes from the PNFD archive. This can be used to continue an aborted transmission.</p> <p>If the NFVO does not support range requests, the NFVO shall ignore the "Range" header, process the GET request, and return the whole PNFD archive with a 200 OK response (rather than returning a 4xx error status code).</p>	
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the content of the PNFD archive has been read successfully.</p> <p>The payload body shall contain a copy of the PNFD archive.</p> <p>The "Content-Type" HTTP header shall be set to "application/zip".</p>
	n/a	1	206 Partial Content	<p>If the NFVO supports range requests, this response shall be returned when a single consecutive byte range from the content of the PNFD archive has been read successfully according to the request.</p> <p>The response body shall contain the requested part of the PNFD archive.</p> <p>The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [10].</p> <p>The "Content-Type" HTTP header shall be set as defined above for the "200 OK" response.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact pnfdOnboardingState has a value different from ONBOARDED.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>

	ProblemDetails	0..1	416 Range Not Satisfiable	<p>Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the PNFD archive (e.g. "access after end of file").</p> <p>The response body may contain a ProblemDetails structure.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 5.4.7.3.3 PUT

The PUT method is used to upload the content of a PNFD archive.

The user may use this operation to upload a new PNFD version. Different PNFD versions have the same "pnfdInvariantId" attribute, but different "pnfdId" attributes and different PNFD information objects. The design of different PNFD versions and their business logic is out of scope of the present document.

Once on-boarded, the PNFD is known to and validated by the NFVO. The associated PNFD information object is updated with the information populated from the validated PNFD. If there are on-boarded NSDs that refer to this PNFD, their respective NsdInfo records are updated to refer to the on-boarded PNFD.

This method shall follow the provisions specified in the tables 5.4.7.3.3-1 and 5.4.7.3.3-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.7.3.3-1: URI query parameters supported by the PUT method on this resource**

Name	Cardinality	Description
none supported		

**Table 5.4.7.3.3-2: Details of the PUT request/response on this resource**

	Data type	Cardinality	Description	
	Request body	n/a	1	<p>The payload body contains a ZIP file representing the PNFD archive.</p> <p>The request shall set the "Content-Type" HTTP header to "application/zip".</p>
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the PNFD archive has been accepted for uploading, but the processing has not been completed. It is expected to take some time for processing (asynchronous mode).</p> <p>The response body shall be empty. See note.</p>
	n/a	1	204 No Content	<p>Shall be returned when the PNFD archive content has been uploaded and validated successfully.</p> <p>The response body shall be empty.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the "pnfdOnboardingState" attribute has a value other than "CREATED" or "ERROR".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>

	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.
NOTE:	The API consumer can track the uploading progress by receiving the "PnfdOnBoardingNotification" and "PnfdOnBoardingFailureNotification" from the NFVO or by reading the status of the individual PNF descriptor resource using the GET method.			

#### 5.4.7.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.7.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.7a Resource: PNFD

#### 5.4.7a.1 Description

This resource represents the PNFD contained in an on-boarded PNFD archive. The API consumer can use this resource to obtain the content of the PNFD.

#### 5.4.7a.2 Resource definition

The resource URI is:

**{apiRoot}/nsd/{apiMajorVersion}/pnf\_descriptors/{pnfdInfoId}/pnfd**

This resource shall support the resource URI variables defined in table 5.4.7a.2-1.

**Table 5.4.7a.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
pnfdInfoId	Identifier of the individual PNF descriptor. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new PNF descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

#### 5.4.7a.3 Resource methods

##### 5.4.7a.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

##### 5.4.7a.3.2 GET

The GET method reads the content of the PNFD within a PNFD archive.

The PNFD can be implemented as a single file or as a collection of multiple files. If the PNFD is implemented in the form of multiple files, a ZIP file embedding these files shall be returned. If the PNFD is implemented as a single file, either that file or a ZIP file embedding that file shall be returned.



The selection of the format is controlled by the "Accept" HTTP header passed in the GET request:

- If the "Accept" header contains only "text/plain" and the PNFD is implemented as a single file, the file shall be returned; otherwise, an error message shall be returned.
- If the "Accept" header contains only "application/zip", the single file or the multiple files that make up the PNFD shall be returned embedded in a ZIP file.
- If the "Accept" header contains both "text/plain" and "application/zip", it is up to the NFVO to choose the format to return for a single-file PNFD; for a multi-file PNFD, a ZIP file shall be returned.

The default format of the ZIP file shall be the one specified in ETSI GS NFV-SOL 004 [5] where only the YAML files representing the PNFD, and information necessary to navigate the ZIP file and to identify the file that is the entry point for parsing the PNFD and (if requested) further security information are included. This means that the content of the ZIP archive shall contain the following files from the PNFD archive:

- TOSCA.meta (if available in the PNFD archive);
- the main service template (either as referenced from TOSCA.meta or available as a file with the extension ".yaml" or ".yml" from the root of the archive);
- every component of the PNFD referenced (recursively) from the main service template;
- the related security information, if the "include\_signatures" URI parameter is provided, as follows:
  - the manifest file;
  - the singleton certificate file in the root of the PNFD archive (if available in the PNFD archive);
  - the signing certificates of the individual files included in the ZIP archive (if available in the PNFD archive);
  - the signatures of the individual files (if available in the PNFD archive).

This method shall follow the provisions specified in the tables 5.4.7a.3.2-1 and 5.4.7a.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.7a.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
include_signatures	0..1	<p>If this parameter is provided, the NFVO shall include in the ZIP file the security information as specified above.</p> <p>This URI query parameter is a flag, i.e. it shall have no value.</p> <p>The NFVO shall support this parameter.</p>

Table 5.4.7a.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			The request shall contain the appropriate entries in the "Accept" HTTP header as defined above.
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the content of the PNFD has been read successfully.</p> <p>The payload body shall contain a copy of the file representing the PNFD or a ZIP file that contains the file or multiple files representing the PNFD, as specified above.</p> <p>The "Content-Type" HTTP header shall be set according to the format of the returned file. It shall be set to "text/plain" for a YAML file.</p>
	ProblemDetails	0..1	406 Not Acceptable	<p>If the "Accept" header does not contain at least one name of a content type for which the NFVO can provide a representation of the PNFD, the NFVO shall respond with this response code.</p> <p>The "ProblemDetails" structure may be included with the "detail" attribute providing more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "pnfdOnboardingState " has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

### 5.4.7a.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.7a.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.7a.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 5.4.7b Resource: PNFD archive manifest

### 5.4.7b.1 Description

This resource represents the manifest file contained in an on-boarded PNFD archive. The API consumer can use this resource to obtain the content of the manifest file.

### 5.4.7b.2 Resource definition

The resource URI is:

**{apiRoot}/nsd/{apiMajorVersion}/pnf\_descriptors/{pnfdInfoId}/manifest**

This resource shall support the resource URI variables defined in table 5.4.7b.2-1.

**Table 5.4.7b.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
pnfdInfoId	Identifier of the individual PNF descriptor. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new PNF descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

### 5.4.7b.3 Resource methods

#### 5.4.7b.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.7b.3.2 GET

The GET method reads the content of the manifest file within a PNFD archive.

This method shall follow the provisions specified in the tables 5.4.7b.3.2-1 and 5.4.7b.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.7b.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
include_signatures	0..1	<p>If this parameter is provided, the NFVO shall return the manifest and related security information (signature and certificate) either in a single text file if the signature and certificate are included in the manifest file, or in a zip file containing the manifest and the certificate file, if this is provided as a separate file in the PNFD archive.</p> <p>If this parameter is not given, the NFVO shall provide only a copy of the manifest file, as onboarded. If the security information is included in the onboarded manifest, it shall also be included in the returned copy.</p> <p>This URI query parameter is a flag, i.e. it shall have no value.</p> <p>The NFVO shall support this parameter.</p>

Table 5.4.7b.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
n/a				
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the content of the manifest file has been read successfully.</p> <p>If the "include_signatures" URI query parameter was absent in the request, or if the manifest file has all security-related information embedded (i.e. there is no separate certificate file), the payload body shall contain a copy of the manifest file of the PNFD archive, and the "Content-Type" HTTP header shall be set to "text/plain".</p> <p>If the "include_signatures" URI query parameter was present in the related request and the manifest file does not have all the security-related information embedded (i.e. there is a separate certificate file), the "Content-Type" HTTP header shall be set to "application/zip" and the payload body shall contain a ZIP archive which includes:</p> <ul style="list-style-type: none"> <li>- a copy of the manifest file of the PNFD archive;</li> <li>- a copy of the related individual certificate file.</li> </ul>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "pnfdOnboardingState " has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 5.4.7b.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.7b.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.7b.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 5.4.7c Resource: Individual PNFD archive artifact

### 5.4.7c.1 Description

This resource represents an individual artifact contained in a PNFD archive. The API consumer can use this resource to fetch the content of the artifact.

### 5.4.7c.2 Resource definition

The resource URI is:

**{apiRoot}/nsd/{apiMajorVersion}/pnf\_descriptors/{pnfdInfoId}/artifacts/{artifactPath}**

This resource shall support the resource URI variables defined in table 5.4.7c.2-1.

**Table 5.4.7c.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
pnfdInfoId	Identifier of the individual PNF descriptor. See note 1.
artifactPath	<p>For an artifact contained as a file in the PNFD archive, this variable shall contain a sequence of one or path segments representing the path of the artifact within the PNFD archive, relative to the root of the package.</p> <p>EXAMPLE:      foo/bar/m%40ster.sh</p> <p>For an external artifact represented as a URI in the PNFD archive manifest, this variable shall contain the URI as provided in the NS archive.</p> <p>See note 2 and note 3.</p>
NOTE 1:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new PNF descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.
NOTE 2:	This identifier can be retrieved from the "artifactPath" attribute of the applicable "artifacts" entry in the body of the response to a GET request querying the "Individual PNFD" resource.
NOTE 3:	Since multiple path segments are allowed to be contained in this variable, the "/" character that separates these segments is not percent-encoded. Each individual segment is percent-encoded if necessary as defined in clause 4.1 of ETSI GS NFV-SOL 013 [16].

### 5.4.7c.3 Resource methods

#### 5.4.7c.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.7c.3.2 GET

The GET method fetches the content of an individual artifact within a PNFD archive.

This method shall follow the provisions specified in the tables 5.4.7c.3.2-1 and 5.4.7c.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.7c.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
include_signatures	0..1	<p>If this parameter is provided, the NFVO shall return the artifact and related security information (such as signature and optional certificate) in a ZIP archive.</p> <p>If this parameter is not given, the NFVO shall provide only a copy of the artifact file. This URI query parameter is a flag, i.e. it shall have no value.</p> <p>The NFVO shall support this parameter.</p>

Table 5.4.7c.3.2-2: Details of the GET request/response on this resource

	Data type	Cardinality	Description	
	Request body	n/a		<p>The request may contain a "Range" HTTP header to obtain single range of bytes from an artifact file. This can be used to continue an aborted transmission.</p> <p>If the Range header is present in the request and the NFVO does not support responding to range requests with a 206 response, it shall return a 200 OK response instead as defined below.</p>
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the whole content of the artifact file has been read successfully.</p> <p>If the "include_signatures" request URI parameter was not provided in the related request, the payload shall contain a copy of the artifact file from the PNF archive, as defined by ETSI GS NFV-SOL 004 [5], and the "Content-Type" HTTP header shall be set according to the content type of the artifact file. If the artifact is encrypted, the header shall be set to the value "application/cms" (IETF RFC 7193 [17]). If the content type cannot be determined, the header shall be set to the value "application/octet-stream".</p> <p>If the "include_signatures" request URI parameter was provided in the related request, the "Content-Type" HTTP header shall be set to "application/zip" and the payload body shall contain a ZIP archive which includes:</p> <ul style="list-style-type: none"> <li>- a copy of the artifact file from the VNF package, as defined by ETSI GS NFV-SOL 004 [5];</li> <li>- the related security information (individual signature file and optional related individual certificate file).</li> </ul>
	n/a	1	206 Partial Content	<p>If the NFVO supports range requests and the "include_signatures" request URI parameter was not present in the related request, this response shall be returned when a single consecutive byte range from the content of the artifact file has been read successfully according to the request.</p> <p>The response body shall contain the requested part of the artifact file from the PNF archive, as defined by ETSI GS NFV-SOL 004 [5].</p> <p>The "Content-Type" HTTP header shall be set according to the content type of the artifact file. If the content type cannot be determined, the header shall be set to the value "application/octet-stream".</p> <p>The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [10].</p>

	ProblemDetails	0..1	406 Not Acceptable	If the related request contained an "Accept" header not compatible with the Content type "application/zip" but the "include_signatures" flag was provided, the NFVO shall respond with this response code.  The "ProblemDetails" structure may be included with the "detail" attribute providing more information about the error.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.  Typically, this is due to the fact that "pnfdOnboardingState" has a value different from "ONBOARDED".  The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	416 Range Not Satisfiable	The byte range passed in the "Range" header did not match any available byte range in the artifact file (e.g. "access after end of file").  The response body may contain a ProblemDetails structure.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 5.4.7c.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.7c.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.7c.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 5.4.8 Resource: Subscriptions

### 5.4.8.1 Description

This resource represents subscriptions. The API consumer can use this resource to subscribe to notifications related to NSD management and to query its subscriptions.

### 5.4.8.2 Resource definition

The resource URI is:

**{apiRoot}/nsd/{apiMajorVersion}/subscriptions**

This resource shall support the resource URI variables defined in table 5.4.8.2-1.

**Table 5.4.8.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.

### 5.4.8.3 Resource methods

#### 5.4.8.3.1 POST

The POST method creates a new subscription.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 5.4.8.3.1-1 and 5.4.8.3.1-2.

Creation of two subscription resources with the same callback URI and the same filter can result in performance degradation and will provide duplicates of notifications to the OSS, and might make sense only in very rare use cases. Consequently, the NFVO may either allow creating a subscription resource if another subscription resource with the same filter and callback URI already exists (in which case it shall return the "201 Created" response code), or may decide to not create a duplicate subscription resource (in which case it shall return a "303 See Other" response code referencing the existing subscription resource with the same filter and callback URI).

**Table 5.4.8.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Remarks
none supported		

**Table 5.4.8.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Remarks	
	NsdmSubscriptionRequest	1	Details of the subscription to be created, as defined in clause 5.5.2.7.	
Response body	Data type	Cardinality	Response Codes	Remarks
	NsdmSubscription	1	201 Created	<p>Shall be returned when the subscription has been created successfully.</p> <p>A representation of the created "Individual subscription" resource shall be returned in the response body, as defined in clause 5.5.2.8.</p> <p>The HTTP response shall include a "Location:" HTTP header that points to the created "Individual subscription" resource.</p>
	n/a		303 See Other	<p>Shall be returned when a subscription with the same callback URI and the same filter already exists and the policy of the NFVO is to not create redundant subscriptions.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource.</p> <p>The response body shall be empty.</p>



	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 5.4.10.3.2 and the test has failed.</p> <p>In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 5.4.8.3.2 GET

The GET method queries the list of active subscriptions of the functional block that invokes the method. It can be used e.g. for resynchronization after error situations.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 5.4.8.3.2-1 and 5.4.8.3.2-2.

**Table 5.4.8.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Remarks
filter	0..1	<p>Attribute filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>All attribute names that appear in the NsdmSubscription and in data types referenced from it shall be supported by the NFVO in the filter expression.</p>
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 5.4.8.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
n/a				
Response body	Data type	Cardinality	Response Codes	Remarks
	NsdmSubscription	0..N	200 OK	<p>Shall be returned when the list of subscriptions has been queried successfully.</p> <p>The response body shall contain in an array the representations of all active subscriptions of the functional block that invokes the method, i.e. zero or more representations of NSD management subscriptions as defined in clause 5.5.2.8.</p> <p>If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [16].</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails		400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 5.4.8.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.8.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.8.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 5.4.9 Resource: Individual subscription

### 5.4.9.1 Description

This resource represents an individual subscription. It can be used by the API consumer to read and to terminate a subscription to notifications related to NSD management.

### 5.4.9.2 Resource definition

The resource URI is:

**{apiRoot}/nsd/{apiMajorVersion}/subscriptions/{subscriptionId}**

This resource shall support the resource URI variables defined in table 5.4.9.2-1.

**Table 5.4.9.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
subscriptionId	Identifier of this subscription. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new subscription resource. It can also be retrieved from the "id" attribute in the payload body of that response.

### 5.4.9.3 Resource methods

#### 5.4.9.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 5.4.9.3.2 GET

The GET method retrieves information about a subscription by reading an individual subscription resource.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 5.4.9.3.2-1 and 5.4.9.3.2-2.

**Table 5.4.9.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Remarks
none supported		

**Table 5.4.9.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	NsdmSubscription	1	200 OK	Shall be returned when information about an individual subscription has been read successfully.  The response body shall contain a representation of the subscription resource.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 5.4.9.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.9.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.9.3.5 DELETE

The DELETE method terminates an individual subscription.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 5.4.9.3.5-1 and 5.4.9.3.5-2.

**Table 5.4.9.3.5-1: URI query parameters supported by the DELETE method on this resource**

Name	Cardinality	Remarks
none supported		

**Table 5.4.9.3.5-2: Details of the DELETE request/response on this resource**

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		204 No Content	Shall be returned when the "Individual subscription" resource has been deleted successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 5.4.10 Resource: Notification endpoint

### 5.4.10.1 Description

This resource represents a notification endpoint. The API producer can use this resource to send notifications to a subscribed API consumer, which has provided the URI of this resource during the subscription process.

### 5.4.10.2 Resource definition

The resource URI is provided by the API consumer when creating the subscription.

This resource shall support the resource URI variables defined in table 5.4.10.2-1.

**Table 5.4.10.2-1: Resource URI variables for this resource**

Name	Definition
none supported	

### 5.4.10.3 Resource methods

#### 5.4.10.3.1 POST

The POST method delivers a notification from the API producer to the API consumer.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 5.4.10.3.1-1 and 5.4.10.3.1-2.

**Table 5.4.10.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Remarks
none supported		

Each notification request body shall include exactly one of the alternatives defined in table 5.4.10.3.1-2.

**Table 5.4.10.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Remarks	
	NsdOnBoardingNotification	1	A notification about the successful on-boarding of an NSD.	
	NsdOnBoardingFailureNotification	1	A notification about the failure of on-boarding an NSD.	
	NsdChangeNotification	1	A notification about the state change of an on-boarded NSD.	
	NsdDeletionNotification	1	A notification about the deletion of an on-boarded NSD.	
	PnfdOnBoardingNotification	1	A notification about the successful on-boarding of a PNFD.	
	PnfdOnBoardingFailureNotification	1	A notification about the failure of on-boarding a PNFD.	
	PnfdDeletionNotification	1	A notification about the deletion of an on-boarded PNFD.	
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		204 No Content	Shall be returned when the notification has been delivered successfully.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 5.4.10.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during subscription.

This method shall follow the provisions specified in the tables 5.4.10.3.2-1 and 5.4.10.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 5.4.10.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

Table 5.4.10.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the notification endpoint has been tested successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 5.4.10.3.3 PUT

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.10.3.4 PATCH

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 5.4.10.3.5 DELETE

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 5.5 Data model

### 5.5.1 Introduction

This clause defines the request and response data structures of the NSD Lifecycle management interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error and may choose to ignore them.

### 5.5.2 Resource and notification data types

#### 5.5.2.1 Type: NsdInfoModifications

This type represents attribute modifications for an individual NS descriptor resource based on the "NsdInfo" data type. The attributes of "NsdInfo" that can be modified are included in the "NsdInfoModifications" data type.

The "NsdInfoModifications" data type shall comply with the provisions defined in table 5.5.2.1-1.

**Table 5.5.2.1-1: Definition of the NsdInfoModifications data type**

Attribute name	Data type	Cardinality	Description
nsdOperationalState	NsdOperationalStateType	0..1	New value of the "nsdOperationalState" attribute in "NsdInfo" data type. See note.  Permitted values: - ENABLED - DISABLED
userDefinedData	KeyValuePairs	0..1	Modifications of the "userDefinedData" attribute in "NsdInfo" data type. See note. If present, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [11]).
NOTE: At least one of the attributes - nsdOperationalState and userDefinedData - shall be present.			

### 5.5.2.2 Type: NsdInfo

This type represents a response for the query NSD operation. It shall comply with the provisions defined in table 5.5.2.2-1.

**Table 5.5.2.2-1: Definition of the NsdInfo data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the onboarded individual NS descriptor resource. This identifier is allocated by the NFVO.
nsdId	Identifier	0..1	This identifier, which is allocated by the NSD designer, identifies the NSD in a globally unique way. It is copied from the NSD content and shall be present after the NSD content is on-boarded.
nsdExtInvariantId	Identifier	0..1	Identifies an NSD in a version independent manner. This attribute is invariant across versions of the NSD that fulfil certain conditions related to the external connectivity and management of the NS. It shall be present after the NSD content has been on-boarded if it is included in the NSD and shall be absent otherwise. If present it is copied from the NSD of the on-boarded content.
nsdName	String	0..1	Name of the onboarded NSD. This information is copied from the NSD content and shall be present after the NSD content is on-boarded.
nsdVersion	Version	0..1	Version of the on-boarded NSD. This information is copied from the NSD content and shall be present after the NSD content is on-boarded.
nsdDesigner	String	0..1	Designer of the on-boarded NSD. This information is copied from the NSD content and shall be present after the NSD content is on-boarded.
nsdInvariantId	Identifier	0..1	This identifier, which is allocated by the NSD designer, identifies an NSD in a version independent manner. This attribute is invariant across versions of NSD with no constraint on the changes across versions. This information is copied from the NSD of the on-boarded content and shall be present after the NSD content is on-boarded.
vnfPkgIds	Identifier	0..N	Identifies the VNF package for the VNFD referenced by the on-boarded NS descriptor resource. See notes 1 and 5.
pnfdInfolds	Identifier	0..N	Identifies the PnfdInfo element for the PNFD referenced by the on-boarded NS descriptor resource. See note 6.
nestedNsdInfolds	Identifier	0..N	Identifies the NsdInfo element for the nested NSD referenced by the on-boarded NS descriptor resource. See notes 1 and 7.

Attribute name	Data type	Cardinality	Description
archiveSecurityOption	Enum (inlined)	0..1	Signals the security option used by the NSD archive as defined in clause 5.1 of ETSI GS NFV-SOL 007 [18]. It shall be present after the VNF package content has been on-boarded and absent otherwise.  Valid values: - OPTION_1 - OPTION_2
signingCertificate	String	0..1	The singleton signing certificate if it is included as a file in the NSD archive.
artifacts	NsdArchiveArtifactInfo	0..N	Information about NSD archive artifacts contained in the NSD archive.  This attribute shall not be present before the NSD archive content is on-boarded. Otherwise, this attribute shall be present if the NSD archive contains artifacts.
nsdOnboardingState	NsdOnboardingStateType	1	Onboarding state of the individual NS descriptor resource. See note 4.
onboardingFailureDetails	ProblemDetails	0..1	Failure details of current onboarding procedure. See clause 6.3 of ETSI GS NFV-SOL 013 [16] for the details of "ProblemDetails" structure.  It shall be present when the "nsdOnboardingState" attribute is "ERROR", i.e. the uploading or processing has failed in NFVO, unless it has been requested to be excluded via an attribute selector.
nsdOperationalState	NsdOperationalStateType	1	Operational state of the individual NS descriptor resource. This attribute can be modified with the PATCH method. See note 2 and note 4.
nsdUsageState	NsdUsageStateType	1	Usage state of the individual NS descriptor resource. See note 3 and note 4.
userDefinedData	KeyValuePairs	0..1	User defined data for the individual NS descriptor resource. This attribute can be modified with the PATCH method.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>nsd_content	Link	1	Link to the NSD content resource.
<p>NOTE 1: At least one of the attributes - vnfPkgId and nestedNsdInfolist shall be present, after the NSD is on-boarded.</p> <p>NOTE 2: If the value of the nsdOnboardingState attribute is not equal to "ONBOARDED", the value of the nsdOperationalState attribute shall be equal to "DISABLED".</p> <p>NOTE 3: If the value of the nsdOnboardingState attribute is not equal to "ONBOARDED", the value of the nsdUsageState attribute shall be equal to "NOT_IN_USE".</p> <p>NOTE 4: State changes of an NSD are illustrated in clause B.2.</p> <p>NOTE 5: This attribute shall be present after the NSD is on-boarded for those VNF packages that are already on-boarded.</p> <p>NOTE 6: This attribute shall be present after the NSD is on-boarded for those PNFs that are already on-boarded.</p> <p>NOTE 7: This attribute shall be present after the NSD is on-boarded for those nested NSDs that are already on-boarded.</p>			

### 5.5.2.3 Type: CreateNsdInfoRequest

This type creates a completely new NS descriptor resource. It shall comply with the provisions defined in table 5.5.2.3-1.

**Table 5.5.2.3-1: Definition of the CreateNsdInfoRequest data type**

Attribute name	Data type	Cardinality	Description
userDefinedData	KeyValuePairs	0..1	User-defined data for the NS descriptor resource to be created.  It shall be present when the user defined data is set for the individual NS descriptor resource to be created.



### 5.5.2.4 Type: PnfdInfoModifications

This type represents attribute modifications for an individual PNF descriptor resource based on the "PnfdInfo" data type. The attributes of "PnfdInfo" that can be modified are included in the "PnfdInfoModifications" data type.

The "PnfdInfoModifications" data type shall comply with the provisions defined in table 5.5.2.4-1.

**Table 5.5.2.4-1: Definition of the PnfdInfoModifications data type**

Attribute name	Data type	Cardinality	Description
userDefinedData	KeyValuePairs	1	Modifications of the "userDefinedData" attribute in "PnfdInfo" data type.  If present, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [11]).

### 5.5.2.5 Type: PnfdInfo

This type represents a response for the query PNFD operation. It shall comply with the provisions defined in table 5.5.2.5-1.

**Table 5.5.2.5-1: Definition of the PnfdInfo data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the onboarded individual PNF descriptor resource. This identifier is allocated by the NFVO.
pnfdId	Identifier	0..1	This identifier, which is managed by the PNFD designer, identifies the PNFD in a globally unique way. It is copied from the PNFD content and shall be present after the PNFD content is on-boarded.
pnfdName	String	0..1	Name of the onboarded PNFD. This information is copied from the PNFD content and shall be present after the PNFD content is on-boarded.
pnfdVersion	Version	0..1	Version of the onboarded PNFD. This information is copied from the PNFD content and shall be present after the PNFD content is on-boarded.
pnfdProvider	String	0..1	Provider of the onboarded PNFD. This information is copied from the PNFD content and shall be present after the PNFD content is on-boarded.
pnfdInvariantId	Identifier	0..1	Identifies a PNFD in a version independent manner. This attribute is invariant across versions of PNFD. It is copied from the PNFD content and shall be present after the PNFD content is on-boarded.
pnfdExtInvariantId	Identifier	0..1	Identifies a PNFD in a version independent manner. This attribute is invariant across versions of the PNFD that expose the same external connectivity, i.e. same number of pnfExtCpds and same identifiers. It shall be present after the PNFD content has been on-boarded if it is included in the PNFD and shall be absent otherwise. If present it is copied from the PNFD of the on-boarded content.
archiveSecurityOption	Enum (inlined)	0..1	Signals the security option used by the PNFD archive as defined in clause 5.1 of ETSI GS NFV-SOL 004 [5].  Valid values: - OPTION_1 - OPTION_2
signingCertificate	String	0..1	The singleton signing certificate if it is included as a file in the PNFD archive.

Attribute name	Data type	Cardinality	Description
artifacts	PnfdArchiveArtifactInfo	0..N	Information about PNFD archive artifacts contained in the PNFD archive.  This attribute shall not be present before the PNFD archive content is on-boarded. Otherwise, this attribute shall be present if the PNFD archive contains artifacts.
pnfdOnboardingState	PnfdOnboardingStateType	1	Onboarding state of the individual PNF descriptor resource.
onboardingFailureDetails	ProblemDetails	0..1	Failure details of current onboarding procedure. See clause 6.3 of ETSI GS NFV-SOL 013 [16] for the details of "ProblemDetails" structure.  It shall be present when the "pnfdOnboardingState" attribute is "ERROR", i.e. the uploading or processing has failed in the NFVO, unless it has been requested to be excluded via an attribute selector.
pnfdUsageState	PnfdUsageStateType	1	Usage state of the individual PNF descriptor resource.
userDefinedData	KeyValuePairs	0..1	User defined data for the individual PNF descriptor resource. This attribute can be modified with the PATCH method.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>pnfd_content	Link	1	Link to the PNFD Content resource.

### 5.5.2.6 Type: CreatePnfdInfoRequest

This type creates a new PNF descriptor resource. It shall comply with the provisions defined in table 5.5.2.6-1.

**Table 5.5.2.6-1: Definition of the CreatePnfdInfoRequest data type**

Attribute name	Data type	Cardinality	Description
userDefinedData	KeyValuePairs	0..1	User-defined data for the PNF descriptor resource to be created.  It shall be present when the user defined data is set for the individual PNF descriptor resource to be created.

### 5.5.2.7 Type: NsdmSubscriptionRequest

This type represents a subscription request related to notifications about NSD management. It shall comply with the provisions defined in table 5.5.2.7-1.

**Table 5.5.2.7-1: Definition of the NsdmSubscriptionRequest data type**

Attribute name	Data type	Cardinality	Description
filter	NsdmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this subscription, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [16]. This attribute shall only be present if the subscriber requires authorization of notifications.

### 5.5.2.8 Type: NsdmSubscription

This type represents a subscription related to notifications about NSD management. It shall comply with the provisions defined in table 5.5.2.8-1.

**Table 5.5.2.8-1: Definition of the NsdmSubscription data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this subscription resource.
filter	NsdmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.

### 5.5.2.9 Type: NsdOnboardingNotification

This type represents an NSD management notification, which informs the receiver of the successful on-boarding of an NSD. It shall comply with the provisions defined in table 5.5.2.9-1. The support of this notification is mandatory. The notification shall be triggered by the NFVO when the "nsdOnboardingState" attribute of a new NSD has changed to "ONBOARDED".

**Table 5.5.2.9-1: Definition of the NsdOnboardingNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsdOnboardingNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
nsdInfold	Identifier	1	Identifier of the NSD information object. This identifier is allocated by the NFVO.
nsdId	Identifier	1	This identifier, which is managed by the service provider, identifies the NSD in a globally unique way. It is copied from the on-boarded NSD.
_links	NsdmLinks	1	Links to resources related to this notification.

### 5.5.2.10 Type: NsdOnboardingFailureNotification

This type represents an NSD management notification, which informs the receiver of the failure of on-boarding an NSD. It shall comply with the provisions defined in table 5.5.2.10-1. The support of this notification is mandatory. The notification shall be triggered by the NFVO when the on-boarding of an NSD has failed.

**Table 5.5.2.10-1: Definition of the NsdOnboardingFailureNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsdOnboardingFailureNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
nsdInfold	Identifier	1	Identifier of the NSD information object. This identifier is allocated by the NFVO.
nsdId	Identifier	0..1	This identifier, which is managed by the service provider, identifies the NSD in a globally unique way.
onboardingFailureDetails	ProblemDetails	1	Failure details of current onboarding procedure. See clause 6.3 of ETSI GS NFV-SOL 013 [16] for the details of "ProblemDetails" structure.
_links	NsdmLinks	1	Links to resources related to this notification.

### 5.5.2.11 Type: NsdChangeNotification

This type represents an NSD management notification, which informs the receiver of a change of the "nsdOperationalState" attribute of an on-boarded NSD. Changes in the value of the "nsdUsageState" and "nsdOnboardingState" attributes are not reported. The notification shall comply with the provisions defined in table 5.5.2.11-1. The support of this notification is mandatory. The notification shall be triggered by the NFVO when the value of the "nsdOperationalState" attribute has changed, and the "nsdOperationalState" attribute has the value "ONBOARDED".

**Table 5.5.2.11-1: Definition of the NsdChangeNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsdChangeNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
nsdInfold	Identifier	1	Identifier of the NSD information object. This identifier is allocated by the NFVO.
nsdId	Identifier	1	This identifier, which is managed by the service provider, identifies the NSD in a globally unique way. It is copied from the on-boarded NSD.
nsdOperationalState	NsdOperationalState Type	1	New operational state of the on-boarded NSD.
_links	NsdmLinks	1	Links to resources related to this notification.

### 5.5.2.12 Type: NsdDeletionNotification

This type represents an NSD management notification, which informs the receiver of the deletion of an on-boarded NSD. The notification shall comply with the provisions defined in table 5.5.2.12-1. The support of this notification is mandatory. The notification shall be triggered by the NFVO when it has deleted an on-boarded NSD.

**Table 5.5.2.12-1: Definition of the NsdDeletionNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsdDeletionNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
nsdInfold	Identifier	1	Identifier of the NSD information object. This identifier is allocated by the NFVO.
nsdId	Identifier	1	This identifier, which is managed by the service provider, identifies the NSD in a globally unique way. It is copied from the on-boarded NSD.
_links	NsdmLinks	1	Links to resources related to this notification.

### 5.5.2.13 Type: PnfdOnboardingNotification

This type represents a PNFD management notification, which informs the receiver of the successful on-boarding of a PNFD. It shall comply with the provisions defined in table 5.5.2.13-1. The support of this notification is mandatory.

The notification is triggered when a new PNFD is on-boarded.

**Table 5.5.2.13-1: Definition of the PnfdOnboardingNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "PnfdOnboardingNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
pnfdInfold	Identifier	1	Identifier of the PNFD information object. This identifier is allocated by the NFVO.
pnfdId	Identifier	1	This identifier, which is managed by the service provider, identifies the PNFD in a globally unique way. It is copied from the on-boarded PNFD.
_links	PnfdmLinks	1	Links to resources related to this notification.

### 5.5.2.14 Type: PnfdOnboardingFailureNotification

This type represents a PNFD management notification, which informs the receiver of the failure of on-boarding a PNFD. It shall comply with the provisions defined in table 5.5.2.14-1. The support of this notification is mandatory.

The notification is triggered when the on-boarding of a PNFD fails.

**Table 5.5.2.14-1: Definition of the PnfdOnboardingFailureNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "PnfdOnboardingFailureNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
pfnfdInfold	Identifier	1	Identifier of the PNFD information object. This identifier is allocated by the NFVO.
pfnfdId	Identifier	0..1	This identifier, which is managed by the service provider, identifies the PNFD in a globally unique way.
onboardingFailureDetails	ProblemDetails	1	Failure details of current onboarding procedure. See clause 6.3 of ETSI GS NFV-SOL 013 [16] for the details of "ProblemDetails" structure.
_links	PnfdmLinks	1	Links to resources related to this notification.

### 5.5.2.15 Type: PnfdDeletionNotification

This type represents a PNFD management notification, which informs the receiver of the deletion of an on-boarded PNFD. The notification shall comply with the provisions defined in table 5.5.2.15-1. The support of this notification is mandatory.

The notification is triggered when an on-boarded PNFD is deleted.

**Table 5.5.2.15-1: Definition of the PnfdDeletionNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "PnfdDeletionNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
pfnfdInfold	Identifier	1	Identifier of the PNFD information object. This identifier is allocated by the NFVO.
pfnfdId	Identifier	1	This identifier, which is managed by the service provider, identifies the PNFD in a globally unique way. It is copied from the on-boarded PNFD.
_links	PnfdmLinks	1	Links to resources related to this notification.

## 5.5.3 Referenced structured data types

### 5.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but can neither be resource representations nor bound to any subscribe/notify mechanism.

### 5.5.3.2 Type: NsdmNotificationsFilter

This type represents a subscription filter related to notifications about NSD management. It shall comply with the provisions defined in table 5.5.3.2-1.

At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).

**Table 5.5.3.2-1: Definition of the NsdmNotificationsFilter data type**

Attribute name	Data type	Cardinality	Description
notificationTypes	Enum (inlined)	0..N	Match particular notification types.  Permitted values: <ul style="list-style-type: none"> <li>- NsdOnBoardingNotification</li> <li>- NsdOnboardingFailureNotification</li> <li>- NsdChangeNotification</li> <li>- NsdDeletionNotification</li> <li>- PnfdOnBoardingNotification</li> <li>- PnfdOnBoardingFailureNotification</li> <li>- PnfdDeletionNotification</li> </ul> See note 1.
nsdInfoId	Identifier	0..N	Match the NsdInfo identifier which is allocated by the NFVO. See note 2.
nsdId	Identifier	0..N	Match the NSD identifier, which is allocated by the NSD designer. See note 2.
nsdName	String	0..N	Match the name of the onboarded NSD.
nsdVersion	Version	0..N	Match the NSD version listed as part of this attribute.
nsdDesigner	String	0..N	Match the NSD designer of the on-boarded NSD.
nsdInvariantId	Identifier	0..N	Match the NSD invariant identifier which is allocated by the NSD designer and identifies an NSD in a version independent manner.
vnfPkgIds	Identifier	0..N	Match VNF packages with a package identifier listed in the attribute.
pnfdInfoIds	Identifier	0..N	Match the PnfdInfo identifier for the PNFD referenced by the on-boarded NSD. See note 3.
nestedNsdInfoIds	Identifier	0..N	Match the NsdInfo identifier for the nested NSD referenced by the on-boarded NSD.
nsdOnboardingState	NsdOnboardingStateType	0..N	Match particular on-boarding state of the NSD.
nsdOperationalState	NsdOperationalStateType	0..N	Match particular operational state of the on-boarded NSD.
nsdUsageState	NsdUsageStateType	0..N	Match particular usage state of the on-boarded NSD.
pnfdId	Identifier	0..N	Match the PNFD identifier which is copied from the PNFD content. See note 3.
pnfdName	String	0..N	Match the name of the onboarded PNFD.
pnfdVersion	Version	0..N	Match the PNFD designer of the on-boarded PNFD.
pnfdProvider	String	0..N	Match the provider of the on-boarded PNFD.
pnfdInvariantId	Identifier	0..N	Match the PNFD in a version independent manner.
pnfdOnboardingState	PnfdOnboardingStateType	0..N	Match particular onboarding state of the PNFD.
pnfdUsageState	PnfdUsageStateType	0..N	Match the usage state of the individual PNF descriptor resource.
NOTE 1: The permitted values of the "notificationTypes" attribute are spelled exactly as the names of the notification types to facilitate automated code generation systems.			
NOTE 2: The attributes "nsdId" and "nsdInfoId" are alternatives to reference to a particular NSD in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.			
NOTE 3: The attributes "pnfdId" and "pnfdInfoId" are alternatives to reference to a particular PNFD in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.			

### 5.5.3.3 Type: NsdmLinks

This type represents the links to resources that an NSD management notification can contain. It shall comply with the provisions defined in table 5.5.3.3-1.

**Table 5.5.3.3-1: Definition of the NsdmLinks data type**

Attribute name	Data type	Cardinality	Description
nsdInfo	NotificationLink	1	Link to the resource representing the NSD to which the notified change applies, i.e. the individual NS descriptor resource that represents the NSD.
subscription	NotificationLink	1	Link to the related subscription.

#### 5.5.3.4 Type: PnfdmLinks

This type represents the links to resources that a PNFD management notification can contain. It shall comply with the provisions defined in table 5.5.3.4-1.

**Table 5.5.3.4-1: Definition of the PnfdmLinks data type**

Attribute name	Data type	Cardinality	Description
pnfdInfo	NotificationLink	1	Link to the resource representing the PNFD to which the notified change applies, i.e. the individual PNF descriptor resource that represents the PNFD.
subscription	NotificationLink	1	Link to the related subscription.

#### 5.5.3.5 Type: NsdArchiveArtifactInfo

This type represents an artifact contained in an NSD archive. It shall comply with provisions defined in table 5.5.3.5-1.

**Table 5.5.3.5-1: Definition of the NsdArchiveArtifactInfo data type**

Attribute name	Data type	Cardinality	Description
artifactPath	String	1	Path in the NSD archive, which identifies the artifact and also allows to access a copy of the artifact.
checksum	Checksum	1	Checksum of the artifact file. Permitted hash algorithms are defined in ETSI GS NFV-SOL 007 [18].
metadata	KeyValuePairs	0..1	The metadata of the artifact that are available in the NSD Archive, such as Content type, size, creation date, etc.

#### 5.5.3.6 Type: PnfdArchiveArtifactInfo

This type represents an artifact contained in a PNFD archive. It shall comply with provisions defined in table 5.5.3.6-1.

**Table 5.5.3.6-1: Definition of the PnfdArchiveArtifactInfo data type**

Attribute name	Data type	Cardinality	Description
artifactPath	String	1	Path in the PNFD archive, which identifies the artifact and also allows to access a copy of the artifact.
checksum	Checksum	1	Checksum of the artifact file. Permitted hash algorithms are defined in ETSI GS NFV-SOL 004 [5].
nonManoArtifactSetId	String	0..1	Non-MANO artifact set identifier of the non-MANO artifact set to which the artifact belongs, as defined in clause 4.3.7 of ETSI GS NFV-SOL 004 [5]. Shall be provided if the artifact is a non-MANO artifact, and shall be omitted otherwise.
metadata	KeyValuePairs	0..1	The metadata of the artifact that are available in the PNFD Archive, such as Content type, size, creation date, etc.



## 5.5.4 Referenced simple data types and enumerations

### 5.5.4.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

### 5.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.2.

### 5.5.4.3 Enumeration: NsdOperationalStateType

The enumeration NsdOperationalStateType shall comply with the provisions defined in table 5.5.4.3-1. It indicates the operational state of the resource.

**Table 5.5.4.3-1: Enumeration NsdOperationalStateType**

Enumeration value	Description
ENABLED	The operational state of the resource is enabled.
DISABLED	The operational state of the resource is disabled.

### 5.5.4.4 Enumeration: NsdUsageStateType

The enumeration NsdUsageStateType shall comply with the provisions defined in table 5.5.4.4-1. It indicates the usage state of the resource.

**Table 5.5.4.4-1: Enumeration NsdUsageStateType**

Enumeration value	Description
IN_USE	The resource is in use.
NOT_IN_USE	The resource is not-in-use.

### 5.5.4.5 Enumeration: NsdOnboardingStateType

The enumeration NsdOnboardingStateType shall comply with the provisions defined in table 5.5.4.5-1. It indicates the onboarding state of the NSD.

**Table 5.5.4.5-1: Enumeration NsdOnboardingStateType**

Enumeration value	Description
CREATED	The NSD information object has been created.
UPLOADING	The associated NSD content is being uploaded.
PROCESSING	The associated NSD content is being processed, e.g. validation.
ONBOARDED	The associated NSD content has been on-boarded successfully.
ERROR	There was an error during upload or processing of the NSD content.

### 5.5.4.6 Enumeration: PnfdOnboardingStateType

The enumeration PnfdOnboardingStateType shall comply with the provisions defined in table 5.5.4.6-1. It indicates the onboarding state of the individual PNF descriptor resource.

**Table 5.5.4.6-1: Enumeration PnfdOnboardingStateType**

Enumeration value	Description
CREATED	The PNF descriptor resource has been created.
UPLOADING	The associated PNFD content is being uploaded.
PROCESSING	The associated PNFD content is being processed, e.g. validation.
ONBOARDED	The associated PNFD content has been on-boarded successfully.
ERROR	There was an error during upload or processing of the associated PNFD content.

#### 5.5.4.7 Enumeration: PnfdUsageStateType

The enumeration PnfdUsageStateType shall comply with the provisions defined in table 5.5.4.7-1. It indicates the usage state of the resource.

**Table 5.5.4.7-1: Enumeration PnfdUsageStateType**

Enumeration value	Description
IN_USE	The resource is in use.
NOT_IN_USE	The resource is not-in-use.

---

## 6 NS Lifecycle Management interface

### 6.1 Description

This interface allows the OSS/BSS to invoke NS lifecycle management operations of NS instances towards the NFVO, and to subscribe to notifications regarding NS lifecycle changes provided by the NFVO.

The operations provided through this interface are as follows:

- Create NS Identifier
- Instantiate NS
- Scale NS
- Update NS
- Query NS
- Terminate NS
- Delete NS Identifier
- Heal NS
- Get Operation Status
- Subscribe
- Query Subscription Information
- Notify
- Terminate Subscription
- Query VNF snapshot information
- Delete VNF snapshot

This interface also enables to invoke error handling procedures (Retry, Rollback, Continue, Cancel, Fail) on the actual NS lifecycle management operation occurrences, and API version information retrieval.

## 6.1a API version

For the NS lifecycle management interface version as specified in the present document, the MAJOR version field shall be 2, the MINOR version field shall be 10, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [16] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v2".

## 6.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [16]. The string "nslcm" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 6.2-1 shows the overall resource URI structure defined for the NS lifecycle management interface.

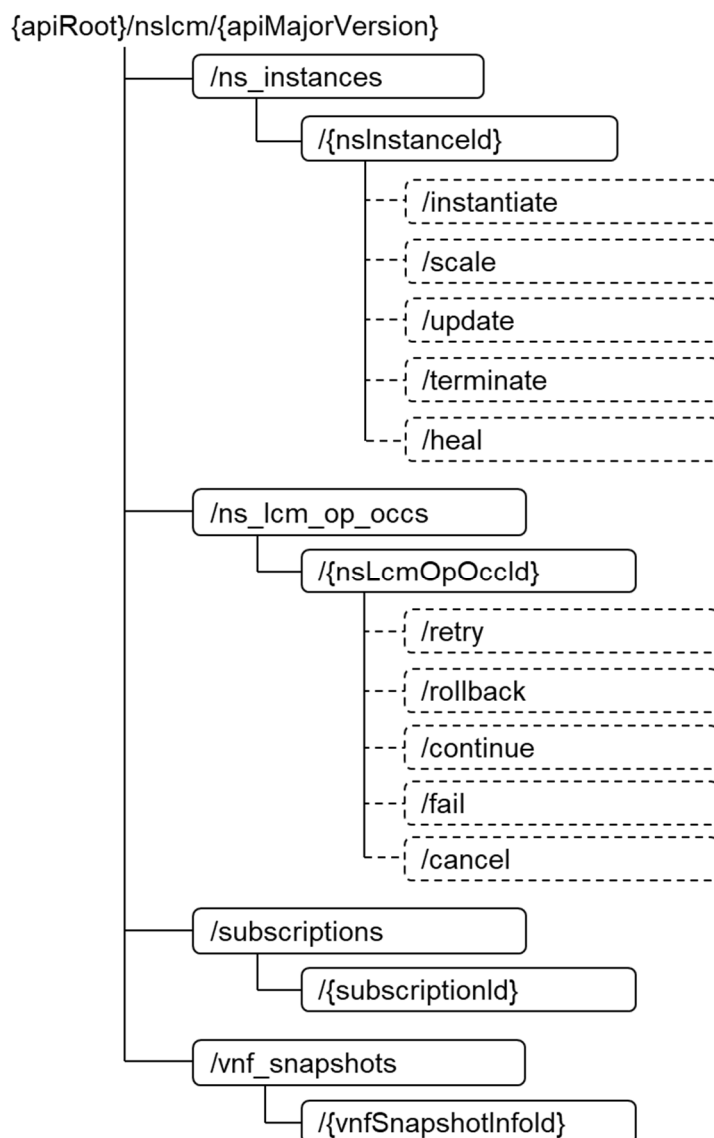


Figure 6.2-1: Resource URI structure of NS Lifecycle Management Interface

Table 6.2-1 lists the individual resources defined, and the applicable HTTP methods.

The NFVO shall support responding to requests for all HTTP methods on the resources in table 6.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [16].

**Table 6.2-1: Resources and methods overview of the NS Lifecycle Management interface**

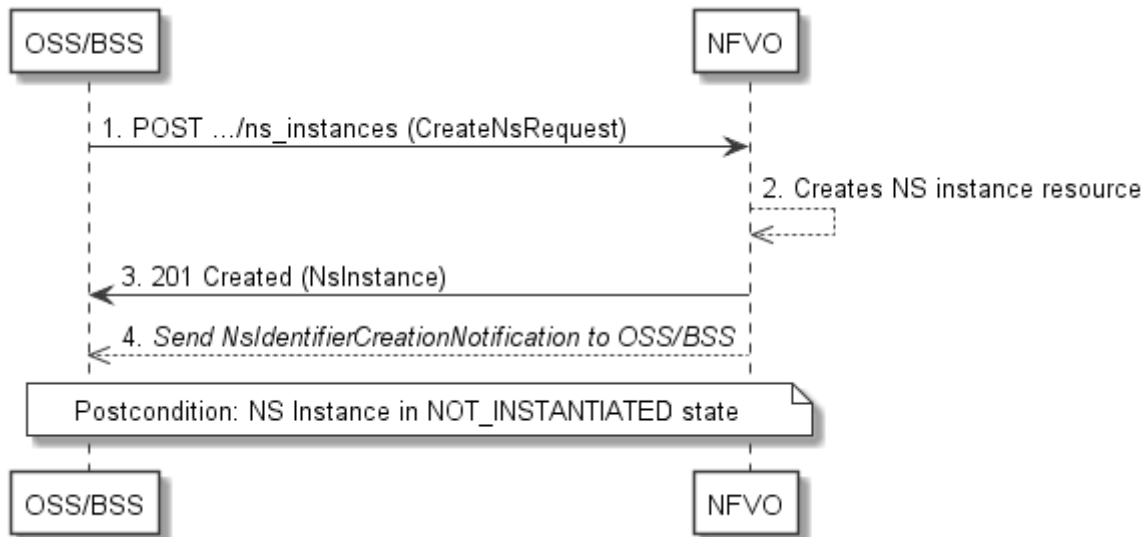
Resource name	Resource URI	HTTP Method	Cat	Meaning
NS instances	/ns_instances	GET	M	Query multiple NS instances.
		POST	M	Create a new "Individual NS instance" resource.
Individual NS instance	/ns_instances/{nsInstanceld}	GET	M	Read an "Individual NS instance" resource.
		DELETE	M	Delete an "Individual NS instance" resource.
Instantiate NS task	/ns_instances/{nsInstanceld}/instantiate	POST	M	Instantiate an NS.
Scale NS task	/ns_instances/{nsInstanceld}/scale	POST	M	Scale an NS instance.
Update NS task	/ns_instances/{nsInstanceld}/update	POST	M	Updates an NS instance.
Terminate NS task	/ns_instances/{nsInstanceld}/terminate	POST	M	Terminate an NS instance.
Heal NS task	/ns_instances/{nsInstanceld}/heal	POST	M	Heal an NS instance.
NS LCM operation occurrences	/ns_lcm_op_occs	GET	M	Query information about multiple NS lifecycle management operation occurrences.
Individual NS LCM operation occurrence	/ns_lcm_op_occs/{nsLcmOpOcclld}	GET	M	Read information about an individual NS lifecycle management operation occurrence.
		PATCH	M	Modify information about an individual NS lifecycle management operation occurrence.
Retry operation task	/ns_lcm_op_occs/{nsLcmOpOcclld}/retry	POST	M	Retry an NS lifecycle management operation occurrence.
Rollback operation task	/ns_lcm_op_occs/{nsLcmOpOcclld}/rollback	POST	M	Rollback an NS lifecycle management operation occurrence.
Continue operation task	/ns_lcm_op_occs/{nsLcmOpOcclld}/continue	POST	M	Continue an NS lifecycle management operation occurrence.
Fail operation task	/ns_lcm_op_occs/{nsLcmOpOcclld}/fail	POST	M	Mark an NS lifecycle management operation occurrence as failed.
Cancel operation task	/ns_lcm_op_occs/{nsLcmOpOcclld}/cancel	POST	M	Cancel an NS lifecycle management operation occurrence.
Subscriptions	/subscriptions	POST	M	Subscribe to NS lifecycle change notifications.
		GET	M	Query multiple subscriptions.
Individual subscription	/subscriptions/{subscriptionld}	GET	M	Read an "Individual subscription" resource.
		DELETE	M	Terminate a subscription.
Notification endpoint	(provided by API consumer)	POST	See note	Notify about NS lifecycle change. See note.
		GET	See note	Test the notification endpoint. See note.
VNF snapshots	/vnf_snapshots	GET	M	Query multiple VNF snapshots.
Individual VNF snapshot	/vnf_snapshots/{vnfSnapshotlnfold}	GET	M	Read an "Individual VNF snapshot" resource.
		DELETE	M	Delete an "Individual VNF snapshot" resource.
NOTE:	The NFVO shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the OSS/BSS. If the OSS/BSS supports invoking the POST method on the "Subscriptions" resource towards the NFVO, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.			

Table 6.4.1.2-1 specifies the preconditions and postconditions applicable to the different NS lifecycle management operations triggered by task resources.

## 6.3 Sequence diagrams (informative)

### 6.3.1 Flow of the creation of an NS instance resource

This clause describes the procedure for the creation of an "Individual NS instance" resource.



**Figure 6.3.1-1: Flow of the creation of an NS instance resource**

**NOTE:** Due to possible race conditions, the 201 response and the NsIdentifierCreationNotification can arrive in any order at the OSS/BSS.

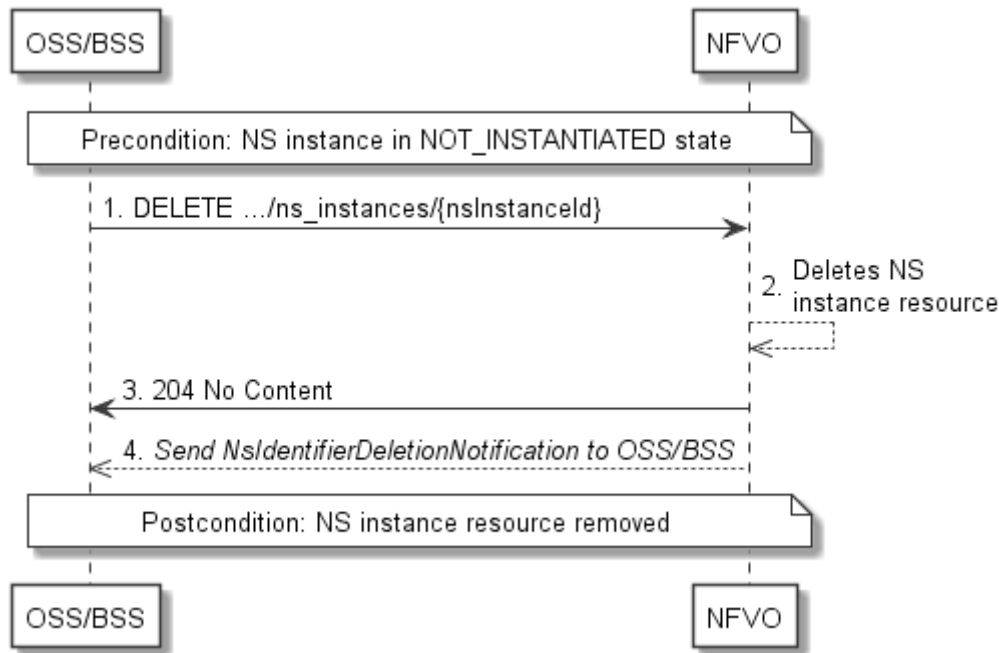
The procedure consists of the following steps as illustrated in figure 6.3.1-1:

- 1) The OSS/BSS sends a POST request to the "NS Instances" resource including in the payload body a data structure of type "CreateNsRequest".
- 2) The NFVO creates a new "Individual NS instance" resource in NOT\_INSTANTIATED state, and the associated NS instance identifier.
- 3) The NFVO returns a 201 Created response containing a representation of the "Individual NS instance" resource just created by the NFVO, and provides the URI of the newly-created resource in the "Location" HTTP header.
- 4) The NFVO sends an NsIdentifierCreationNotification (see clause 6.3.6) to the OSS/BSS to indicate the creation of the "Individual NS instance" resource and the associated NS instance identifier.

**Postcondition:** Upon successful completion, a new "Individual NS instance" resource has been created in "NOT\_INSTANTIATED" state.

### 6.3.2 Flow of the deletion of an NS instance resource

This clause describes the procedure for the deletion of an "Individual NS instance" resource.



**Figure 6.3.2-1: Flow of the deletion of an NS instance resource**

**NOTE:** Due to possible race conditions, the 204 response and the NsIdentifierDeletionNotification can arrive in any order at the OSS/BSS.

**Precondition:** The resource representing the NS instance to be deleted needs to be in NOT\_INSTANTIATED state.

The procedure consists of the following steps as illustrated in figure 6.3.2-1:

- 1) The OSS/BSS sends a DELETE request to the "Individual NS Instance" resource.
- 2) The NFVO deletes the "Individual NS instance" resource and the associated NS instance identifier.
- 3) The NFVO returns a "204 No Content" response with an empty payload body.
- 4) The NFVO sends to the OSS/BSS an NS identifier deletion notification to indicate the deletion of the "Individual NS instance" resource and the associated NS instance identifier.

**Postcondition:** The resource representing the NS instance has been removed from the list of NS instance resources.

**Error Handling:** If the "Individual NS instance" is not in NOT\_INSTANTIATED state, the NFVO rejects the deletion request.

### 6.3.3 Flow of NS lifecycle management operations triggered by task resources

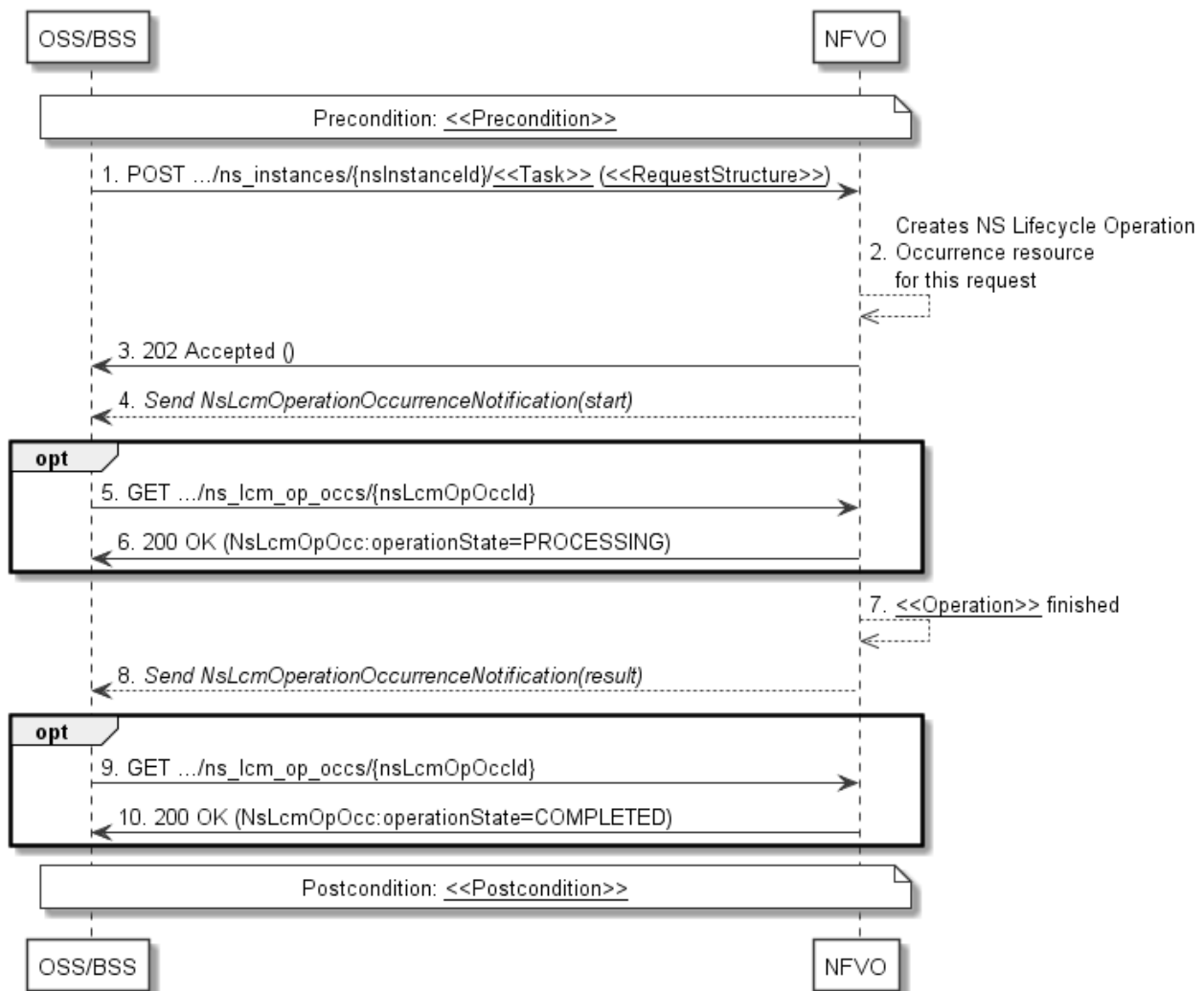
This clause describes the general sequence for NS Lifecycle Management operations that operate on an NS instance resource and are triggered by task resources. The flows for these operations are very similar. The differences between the individual operations are covered in table 6.4.1.2-1.

This flow is applicable to the following operations:

- Instantiate NS
- Scale NS
- Update NS
- Heal NS

- Terminate NS

Figure 6.3.3-1 illustrates the general lifecycle management flow. Placeholders in this flow allow for differentiating between the operations and are marked with double angular brackets "<<...>>".



**Figure 6.3.3-1: Flow of NS lifecycle operations triggered by task resources**

**NOTE:** Due to possible race conditions, the 202 response and the "start" NsLcmOperationOccurrenceNotification can arrive in any order at the OSS/BSS.

**Precondition:** The precondition depends on the actual operation and is described by the template parameter <<Precondition>>. Table 6.4.1.2-1 specifies the applicable precondition.

An NS lifecycle operation, as illustrated in figure 6.3.3-1, consists of the following steps:

- 1) The OSS/BSS sends a POST request to the <<Task>> resource that represents the lifecycle operation to be executed on the NS instance, and includes in the payload body a data structure of type <<RequestStructure>>. The name <<Task>> of the task resource and the <<RequestStructure>> depend on the operation and are described in table 6.4.1.2-1.
- 2) The NFVO creates a new "Individual NS Lifecycle Operation Occurrence" resource for the request.
- 3) The NFVO returns a "202 Accepted" response with an empty payload body and a "Location" HTTP header that points to the new "Individual NS Lifecycle Operation Occurrence" resource, i.e. it includes the URI of that resource which is ".../ns\_lcm\_op\_occs/{nsLcmOpOccId}".
- 4) The NFVO sends to the OSS/BSS an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the start of the lifecycle management operation occurrence. See note.

- 5) If desired, the NFVO can poll the "Individual NS Lifecycle Operation Occurrence" resource to obtain information about the ongoing operation by sending a GET request to the resource that represents the NS Lifecycle Operation Occurrence.
- 6) In the response to that request, the NFVO returns to the OSS/BSS information of the operation, such as the operation status, by providing in the payload body a data structure of type "NsLcmOpOcc".
- 7) The NFVO has finished the operation <<Operation>>.
- 8) The NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the completion of the lifecycle management operation occurrence with the success state "COMPLETED".
- 9) If desired, the OSS/BSS can send a new GET request to the "Individual NS Lifecycle Operation Occurrence" resource.
- 10) In the response to that request, the NFVO returns to the OSS/BSS information about the result of the operation, by providing in the payload body a data structure of type "NsLcmOpOcc".

**Postcondition:** The postcondition depends on the actual operation and is described by the template parameter <<Postcondition>>. Table 6.4.1.2-1 specifies the applicable postcondition.

**Error handling:** If the NS lifecycle management operation fails, error information is provided in the notification message that reports the erroneous completion of the procedure, and is also available in the resource that represents the actual NS lifecycle management operation occurrence related to this NS lifecycle management operation.

Table 6.4.1.2-1 defines how the flow described above is parameterized for the different NS lifecycle management operations.

### 6.3.3a Flows of NS lifecycle management operations triggered by task resources with pre-emption

#### 6.3.3a.1 Overview

This clause provides sequences for NS Lifecycle Management operations that operate on NS instance resources and are triggered by task resources, with the focus on pre-emption in low-resource situations.

The sequences illustrate the use of the API, e.g. when realizing multiple network slices with different priorities that allow the high-priority slices to be prioritized in low-resource situations.

This can be accomplished by using the "priority" attribute in the NsInstance structure, the NS LCM Coordination interface and the capacity shortage notification.

NFV-MANO is assumed to use internal policies and SLAs (out of scope of standardization) to indicate how to handle the NS instance priority level values for the different tenants. In case of a specific action is required, a different priority value of the pre-empted NS instance can be set by the operator's OSS/BSS.

The flow covers the use cases of NS LCM operations with pre-emption. The NS instance priority defines the relative importance of the NS instance compared with other NS instances at resource reservation or resource allocation time. In case of resource limitations, the attribute allows the NFVO to decide confiscating resources from or denying allocation of resources to lower priority NS instances, based on the NS instance priority. This pre-emption capability allows to reallocate resources that were already assigned by NFVO to NS instances with a lower priority level to NS instances with a higher priority level. In case resources need to be confiscated, the NFVO starts a scale-in or terminate operations for one or more lower priority NS instances. Loss of resources can have a performance impact on the affected NS instances. The NS lifecycle management with pre-emption can lead to an increase of the runtime of NS LCM operations.

The NFVO can coordinate with the OSS/BSS regarding the NS instances that are to be scaled in or terminated using the sending NS LCM Coordination interface. Operations on this interface are long-running, potentially with several iterations including delay times in the OSS/BSS response. See clause 12.



The NFVO notifies the OSS/BSS by sending the capacity shortage notification with "status" = "LCM\_RESOURCES\_NOT\_AVAILABLE", to indicate that a shortage situation has been identified which affects certain NS instances by deallocating resources from them or by suspending certain requested LCM operations on them due to lack of resources. Further, The NFVO notifies the OSS/BSS by sending the capacity shortage notification with "status" = "LCM\_SHORTAGE\_END" to indicate the shortage situation has ended and the OSS/BSS can retry the NS LCM operations that were suspended into "FAILED\_TEMP" state. After pre-emption, no automatic action to retry earlier NS LCM operations is executed by the NFVO.

For any NS LCM operation occurrence that was pre-empted, the NFVO sets the flag "affectedCondition" attribute to "PRE-EMPTED" in the "NsLcmCapacityShortageNotification" notifications. Further, it sets the "operationState" attribute to "FAILED\_TEMP" in the NsLcmOpOcc structure that represents the LCM operation occurrence.

Resolving a shortage, resources can become available from the following situations:

- NS instances have been terminated or scaled in.
- Resources become available because the operator has acquired more resources.
- The NFVO receives information from the VIM that additional resources are available.

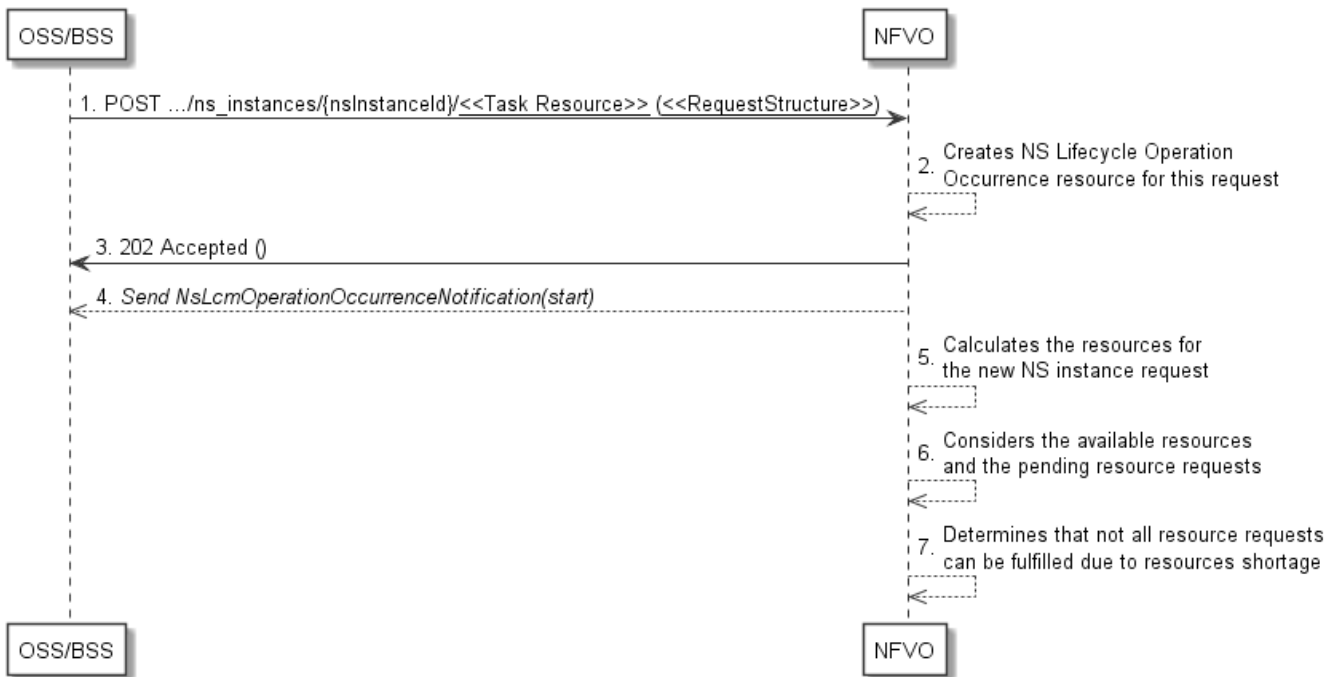
In the following, there are four exemplary shortage use cases to be considered when a resources shortage situation has been identified:

- Shortage use case #1: NS LCM operation (a) on an NS instance (A) is issued while no other NS instances with lower priority exist. Operation (a) fails due to resource shortage, because no resources can be confiscated from other NS instances.
- Shortage use case #2: NS LCM operation (b) on higher priority NS instance is issued while at least one other NS LCM operation (a) on at least one other lower priority NS instance is running and (a) fails due to resource shortage.
- Shortage use case #3: NS LCM operation (a) on an NS instance (A) is issued and triggers auto-scale-in/auto-terminate ("painful") NS LCM operations (b, c, ...) on NS instances (B, C, ...) that have a lower priority than NS instance (A).
- Shortage use case #4: Shortage ends.

The following clauses illustrate the shortage use cases related flows.

### 6.3.3a.2 Steps shared by multiple flows

The following initial steps are shared by multiple shortage scenarios. They are defined below and are referenced from the flows of the individual scenarios.

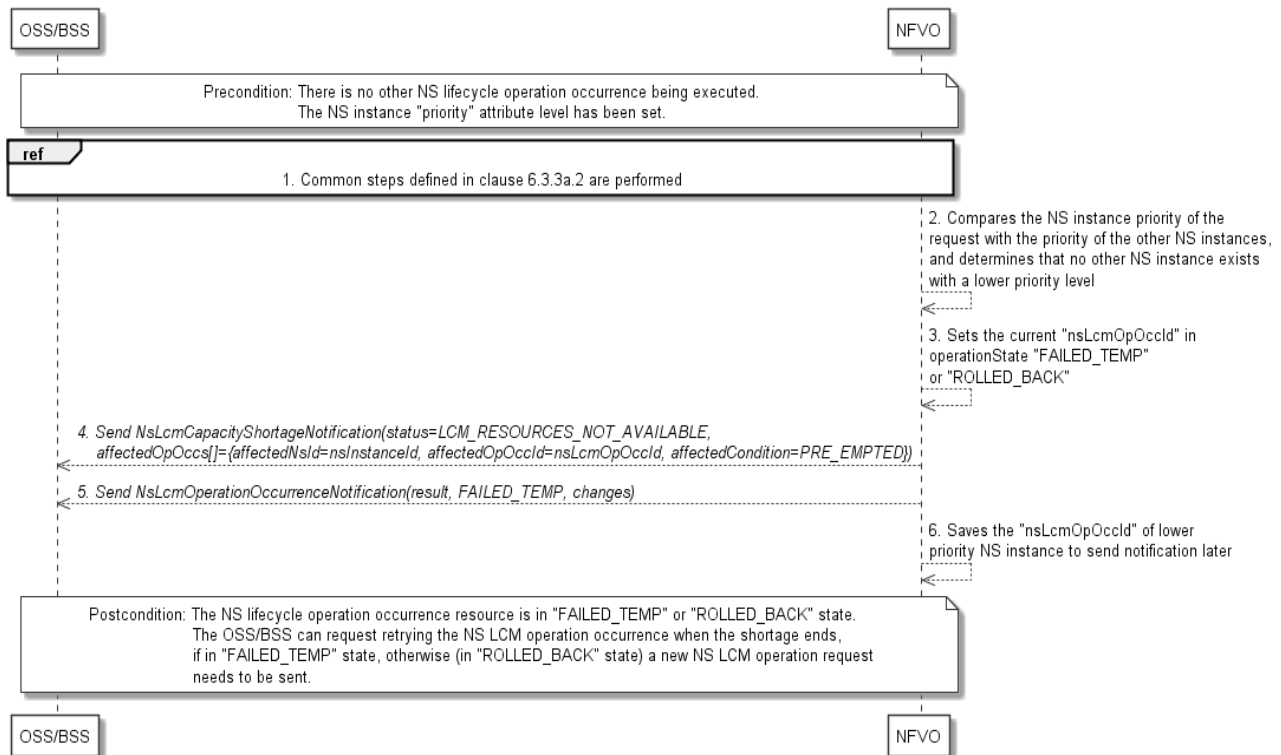


**Figure 6.3.3a.2-1: Steps shared by multiple flows**

- 1) The OSS/BSS sends a POST request to the Instantiate/Scale/Heal task resource that represents the NS lifecycle operation to be executed on the NS instance.
- 2) The NFVO creates a new "Individual NS Lifecycle Operation Occurrence" resource for the request.
- 3) The NFVO returns a "202 Accepted" response with an empty payload body and a "Location" HTTP header that points to the new "Individual NS Lifecycle Operation Occurrence" resource, i.e. it includes the URI of that resource which is ".../ns\_lcm\_op\_occs/{nsLcmOpOccId}".
- 4) The NFVO sends to the OSS/BSS an NS lifecycle management operation occurrence notification ("NsLcmOperationOccurrenceNotification") to indicate the "start" of the lifecycle management operation occurrence.
- 5) The NFVO calculates the necessary resources for the NS LCM request. By means outside the scope of the present document, e.g. processing of the information in NSD and VNFD, VNF LCM operation Granting (see ETSI GS NFV-SOL 003 [4]), the NFVO identifies that the new "Individual NS Lifecycle Operation Occurrence" ("nsLcmOpOccId") needs scarce resources to Instantiate/Scale/Heal the NS instance and those resources cannot be allocated during this NS LCM operation occurrence due to resource shortage.
- 6) The NFVO considers the available resources and the pending resources required for the current NS LCM operation.
- 7) The NFVO determines that not all resource requests can be fulfilled due to the resource shortage situation. Those resources were allocated and are being used by other NS instances. As the NS LCM operation is competing for the same resources and a conflict on resource allocation was identified, the situation is characterized as a resource shortage situation. In case the NFVO identifies that enough resources are available to fulfil the request, the task flow in clause 6.3.3 will be executed as a normal situation has been identified.

### 6.3.3a.3 Shortage Use Case #1

Shortage Use Case #1: NS LCM operation (a) on an NS instance (A) is issued while no other NS instances with lower priority exist. Operation (a) fails due to resource shortage, because no resources can be confiscated from other NS instances.



**Figure 6.3.3a.3-1: Flow of Shortage use case #1**

**Precondition:** There is no other NS lifecycle operation occurrence being executed. The NS instance "priority" attribute level has been set.

An NS lifecycle operation according to shortage use case #1, as illustrated in figure 6.3.3a.3-1, consists of the following steps:

- 1) The common steps defined in clause 6.3.3a.2 are performed.
- 2) The NFVO compares the current NS instance priority level (the "priority" attribute of the "NsInstance" structure) received in the request with the priority levels of the other NS instances and determines that no other NS instance exists with a lower priority level than the current NS instance. This means that the NFVO cannot confiscate resources from any other lower-priority NS instance and the current NS LCM operation occurrence is bound to fail.

NOTE 1: If other NS instances exist with the same priority level value, the NFVO handles these NS instances on a "first come, first served" basis. As the current NS instance was not the first to come and because enough resources cannot be made available, the operation occurrence fails for lack of resources also in this case.

NOTE 2: The NFVO cannot pre-empt network services with the same priority. If a specific action is needed in this case, the OSS/BSS can change the priority level for the current NS instance or of other NS instance(s) and the shortage use case #3 will be executed, see clause 6.3.3a.3.

- 3) The NFVO sets the "operationState" attribute of the current "nsLcmOpOccId" to "FAILED\_TEMP" or "ROLLED\_BACK".

- 4) The NFVO sends to the OSS/BSS an NS LCM Capacity Shortage notification ("NsLcmCapacityShortageNotification"), with "status" = "LCM\_RESOURCES\_NOT\_AVAILABLE", to indicate that a shortage situation has been identified for the current "nsLcmOpOccId" and the NS LCM operation identified by the "nsLcmOpOccId" cannot be executed because the necessary resources are not available.

NOTE 3: The attributes "preemptingNsLcmOpOccId" and "highPrioNsInstanceId" are not present in the notification, and the "affectedOpOccs" structure ("affectedNsId" = current "nsInstanceId", "affectedOpOccId" = current "nsLcmOpOccId" and "affectedCondition" = "PRE\_EMPTED") is present, see clause 6.5.2.19.

- 5) The NFVO sends to the OSS/BSS an NS lifecycle management operation occurrence notification ("NsLcmOperationOccurrenceNotification", see clause 6.3.6) to indicate that the NS lifecycle management operation occurrence identified by "nsLcmOpOccId" has been suspended into an error state with "operationState" = "FAILED\_TEMP". The order in which the response and the notification arrive at the OSS/BSS is not defined.

NOTE 4: The steps 4) and 5) can be executed in any order by the NFVO.

- 6) The NFVO saves the "nsLcmOpOccId" of lower priority NS instance to send notification later when the shortage ends.

**Postcondition:** The NS lifecycle operation occurrence resource is in "FAILED\_TEMP" or "ROLLED\_BACK" state. The OSS/BSS can request retrying the NS LCM operation occurrence when the shortage ends, and the resources are available, as long as it is in "FAILED\_TEMP" state. Operation occurrences in "ROLLED\_BACK" state cannot be retried; a new NS LCM operation request needs to be sent.

NOTE 5: If supported, the NFVO could also automatically roll back the operation for which not enough resources exist, instead of suspending it into "FAILED\_TEMP".

**Error handling:** If the NS lifecycle management operation fails, error information is provided in the notification message that reports the failure of the procedure and is also available in the resource that represents the actual NS lifecycle management operation occurrence related to this NS lifecycle management operation.

#### 6.3.3a.4 Shortage Use Case #2

Shortage Use Case #2: NS LCM operation (b) on higher priority NS instance is issued while at least one other NS LCM operation (a) on at least one other lower priority NS instance is running and (a) fails due to resource shortage.

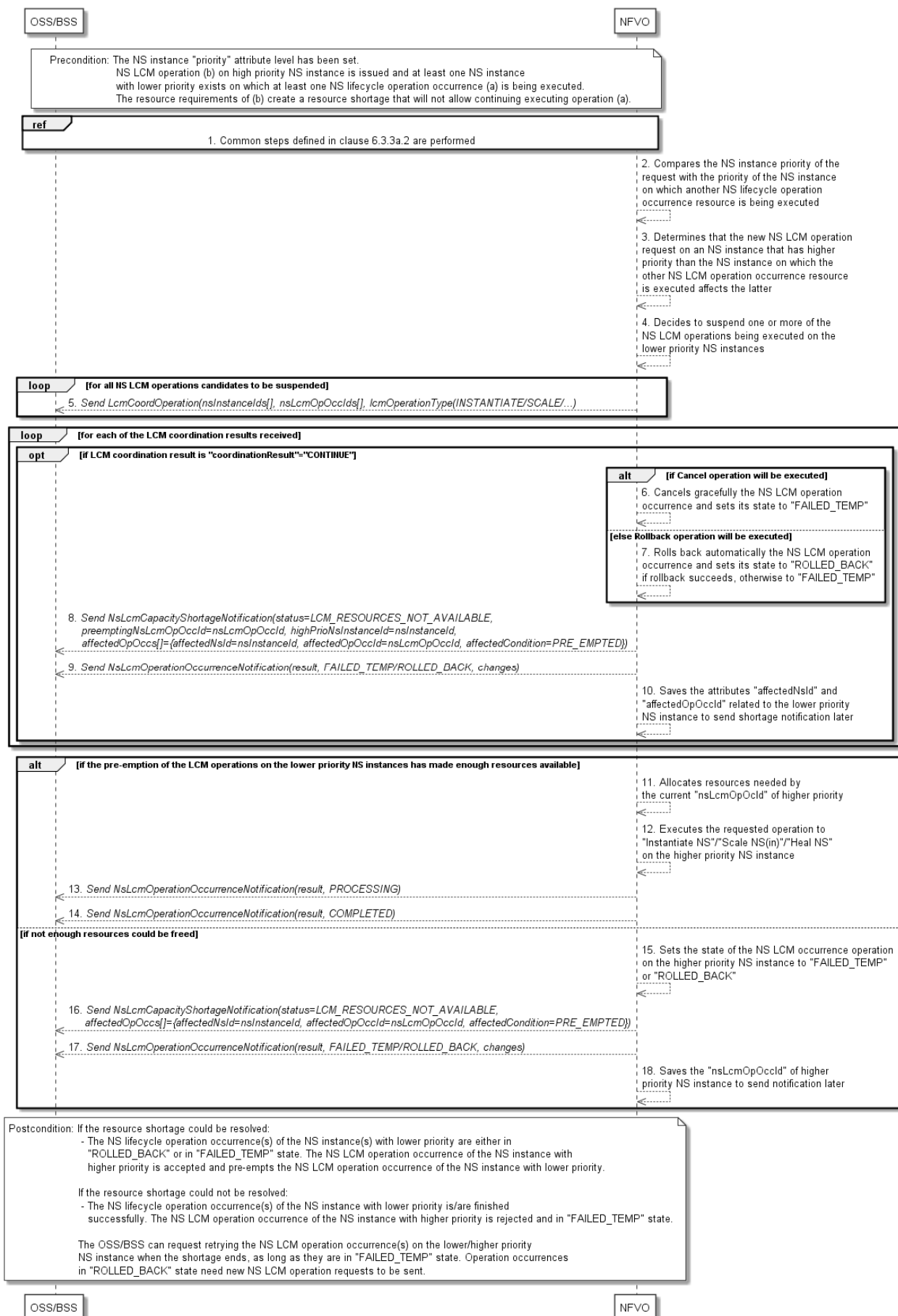


Figure 6.3.3a.4-1: Flow of Shortage use case #2

**Precondition:** The NS instance "priority" attribute level has been set. NS LCM operation (b) on high priority NS instance is issued and at least one NS instance with lower priority exists on which at least one NS lifecycle operation occurrence (a) is being executed. The resource requirements of (b) create a resource shortage that will not allow continuing executing operation (a).

An NS lifecycle operation according to shortage use case #2, as illustrated in figure 6.3.3a.4-1, consists of the following steps:

- 1) The common steps defined in clause 6.3.3a.2 are performed.
- 2) The NFVO compares the current NS instance priority level (the "priority" attribute of the "NsInstance" structure received in the request) with the priority level of the NS instance(s) on which another NS lifecycle operation occurrence resource(s) is being executed.
- 3) The NFVO determines that the new NS LCM operation request on an NS instance that has higher priority affects the other NS LCM operation occurrence resource as it is executed on an NS instance with lower priority.
- 4) The NFVO decides to suspend one or more of the NS LCM operations being executed on the lower priority NS instances.
- 5) Through a loop, for all NS LCM operations candidates to be suspended the NFVO sends to the OSS/BSS NS LCM Coordination operation request to confirm which of the LCM operation occurrences on the lower priority NS instances can be suspended.

In case the OSS/BSS sends the LCM coordination result providing in the response payload body with "coordinationResult" = "CONTINUE" (see clause 12.5.2.3), the NFVO executes the steps 6) to 14).

For each of the ongoing NS LCM operation occurrence(s) on the lower priority NS instance(s) to be suspended, the NFVO executes one of the following two options (step 6) or step 7):

- 6) The NFVO cancels gracefully the NS LCM operation occurrence and sets its state to "FAILED\_TEMP".
- 7) The NFVO rolls back automatically the NS LCM operation occurrence and sets its state to "ROLLED\_BACK" if rollback succeeds, otherwise to "FAILED\_TEMP".
- 8) The NFVO sends to the OSS/BSS NS LCM Capacity Shortage notification(s) ("NsLcmCapacityShortageNotification"), with "status" = "LCM\_RESOURCES\_NOT\_AVAILABLE", of the NS instance(s) with lower priority affected by suspending the LCM operation(s), to indicate that a shortage situation has been identified for the "nsLcmOpOccId".

NOTE 1: The attributes "preemptingNsLcmOpOccId" (= higher priority "nsLcmOpOccId"), "highPrioNsInstanceId" (= higher priority "nsInstanceId") and the "affectedOpOccs" structure ("affectedNsId" = lower priority "nsInstanceId", "affectedOpOccId" = lower priority "nsLcmOpOccId" and "affectedCondition" = "PRE\_EMPTED") are present in the notification, see clause 6.5.2.19.

- 9) The NFVO sends to the OSS/BSS NS lifecycle management operation occurrence notification(s) (see clause 6.3.6) to indicate the suspension of the lifecycle management operation occurrence(s) with the operationState "FAILED\_TEMP" or "ROLLED\_BACK" of lower priority.
- 10) The NFVO saves the attributes "affectedNsId" and "affectedOpOccId" related to the lower priority NS instance to send shortage notification(s) later when the shortage ends.

NOTE 2: The steps 8), 9) and 10) can be executed in any order by the NFVO.

If the pre-emption of the LCM operations on the lower priority NS instances has made enough resources available, the NFVO executes the steps 11) to 14):

- 11) The NFVO allocates the resources needed by the current "nsLcmOpOccId" of higher priority.
- 12) The NFVO executes the requested operation "Instantiate NS"/"Scale NS(in)"/"Heal NS" on the higher priority NS instance.
- 13) The NFVO sends to the OSS/BSS an NS lifecycle management operation occurrence notification ("NsLcmOperationOccurrenceNotification") with operationState "PROCESSING" to indicate the "start" of the "Instantiate NS"/"Scale NS(in)"/"Heal NS" operation on "nsLcmOpOccId" of higher priority.

- 14) The NFVO sends to the OSS/BSS an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the completion of the lifecycle management operation occurrence with the success operationState "COMPLETED" as the "result" of the LCM operation on "nsLcmOpOccId" of higher priority.

Otherwise, if not enough resources could be made available by the pre-emption(s), the NFVO executes the steps 15) to 18):

- 15) The NFVO sets the state of the NS LCM operation occurrence on the higher priority NS instance to "FAILED\_TEMP" or "ROLLED\_BACK".
- 16) The NFVO sends to the OSS/BSS NS LCM Capacity Shortage notification(s) ("NsLcmCapacityShortageNotification"), with "status" = "LCM\_RESOURCES\_NOT\_AVAILABLE", of the NS instance with higher priority affected by suspending the LCM operation, to indicate that a shortage situation has been identified for the "nsLcmOpOccId".

NOTE 3: The attributes "preemptingNsLcmOpOccId" and "highPrioNsInstanceId" are not present in the notification, and the "affectedOpOccs" structure ("affectedNsId" = higher priority "nsInstanceId", "affectedOpOccId" = higher priority "nsLcmOpOccId" and "affectedCondition" = "PRE\_EMPTED") is present, see clause 6.5.2.19.

- 17) NFVO sends to the OSS/BSS NS a lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the suspension of the lifecycle management operation occurrence with the operationState "FAILED\_TEMP" or "ROLLED\_BACK" as the "result" of the "Instantiate NS"/"Scale NS(in)"/"Heal NS" operation on "nsLcmOpOccId" of higher priority. The order in which the response and the notification arrive at the OSS/BSS is not defined.
- 18) The NFVO saves the "nsLcmOpOccId" of higher priority NS instance to send shortage notification later when the shortage ends.

**Postcondition:** If the resource shortage could be resolved:

- The NS lifecycle operation occurrence(s) of the NS instance(s) with lower priority are either in "ROLLED\_BACK" or in "FAILED\_TEMP" state. The NS LCM operation occurrence of the NS instance with higher priority is accepted and pre-empts the NS LCM operation occurrence of the NS instance with lower priority.

If the resource shortage could not be resolved:

- One or more of the NS lifecycle operation occurrence(s) of the NS instance with lower priority is/are finished successfully. The NS LCM operation occurrence of the NS instance with higher priority is rejected and in "FAILED\_TEMP" state.
- The OSS/BSS can request retrying the NS LCM operation occurrences on the lower/higher priority NS instances when the shortage ends, and the resources are available, as long as they are in "FAILED\_TEMP" state. Operation occurrences in "ROLLED\_BACK" state cannot be retried; a new NS LCM operation request needs to be sent.

**Error handling:** If the NS lifecycle management operation fails, error information is provided in the notification message that reports the erroneous completion of the procedure and is also available in the resource that represents the actual NS lifecycle management operation occurrences.

NOTE 4: If suspending the LCM operation on the lower priority NS instance has not saved enough resources to execute the NS LCM operation on the high-priority NS instance, the flow related to shortage use case #3 can be executed in addition. It can also be executed instead of executing this flow #2.

### 6.3.3a.5 Shortage Use Case #3

Shortage use case #3: NS LCM operation (a) on an NS instance (A) is issued and triggers auto-scale-in/auto-terminate ("painful") NS LCM operations (b, c, ...) on NS instances (B, C, ...) that have a lower priority than NS instance (A).

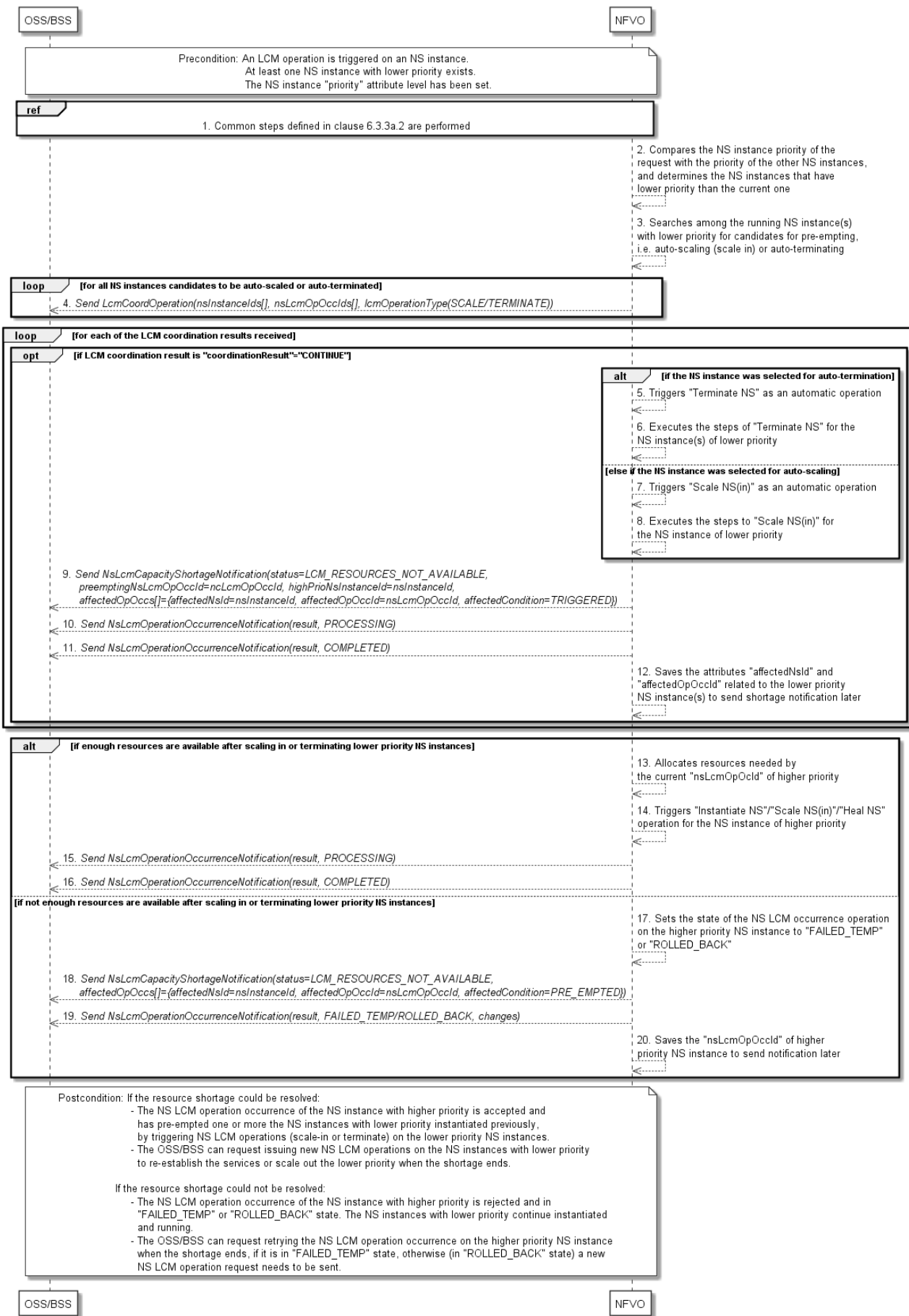


Figure 6.3.3a.5-1: Flow of Shortage use case #3



**Precondition:** An LCM operation is triggered on an NS instance. At least one NS instance with lower priority exists. The NS instance "priority" attribute has been set.

An NS lifecycle operation according to shortage use case #3, as illustrated in figure 6.3.3a-5-1, consists of the following steps:

- 1) The common steps defined in clause 6.3.3a.2 are performed.
- 2) The NFVO compares the current NS instance priority level (the "priority" attribute of the "NsInstance" structure) received in the request with the priority levels of the other NS instances and determines the NS instances that have lower priority than the current one.
- 3) The NFVO searches among the running NS instance(s) with lower priority for NS instance candidates that could be pre-empted, i.e. auto-scaled (scale in) or auto-terminated, so that the necessary resources could be made available to be able to fulfil the NS LCM request on the higher-priority NS instance.
- 4) After identifying the NS instances to be pre-empted, e.g. based on operator policies and SLAs, the NFVO sends to the OSS/BSS an NS LCM Coordination operation request for the NS instances candidates that are to be auto-scaled or auto-terminated.

NOTE 1: Only if the OSS/BSS agrees ("coordinationResult" = "CONTINUE"), any resources can be taken away from the NS instances of lower priority and allocated to NS instance of higher priority.

Through a loop, the NFVO verifies the LCM coordination results. In case the OSS/BSS sends the LCM coordination result providing in the response payload body the attribute "coordinationResult" = "CONTINUE" (see clause 12.5.2.3), the NFVO executes the steps 5) to 16):

If the NS instance was selected for auto-termination as result of the negotiation with OSS/BSS through the LCM operation coordination interface, the NFVO executes the steps 5) to 6):

- 5) The NFVO triggers "Terminate NS" as an automatic operation.

NOTE 2: The NFVO sets the "isAutomaticInvocation" attribute of the "NsLcmOpOcc" structure to "TRUE" for the "nsLcmOpOccId(s)" of the "Terminate NS" operation(s) on the lower priority NS(s) to indicate that these operation(s) have been automatically triggered by the NFVO.

- 6) The NFVO executes the steps of "Terminate NS" for the NS instance(s) of lower priority. See clause 6.3.3.

If the NS instance was selected for auto-scaling as result of the negotiation with OSS/BSS through coordination operation interface, the NFVO executes the steps 7) to 8):

- 7) The NFVO triggers the "Scale NS(in)" as an automatic operation.

NOTE 3: The NFVO sets the "isAutomaticInvocation" attribute of the "NsLcmOpOcc" structure to "TRUE" for the "nsLcmOpOccId(s)" of the "Scale NS" operation(s) on the lower priority NS(s) to indicate that these scale-in operation(s) have been automatically triggered by the NFVO.

- 8) The NFVO executes the steps to "Scale NS(in)" for the NS instance of lower priority. See clause 6.3.3.

- 9) The NFVO sends to the OSS/BSS an NS LCM Capacity Shortage notification ("NsLcmCapacityShortageNotification"), with "status" = "LCM\_RESOURCES\_NOT\_AVAILABLE" and "affectedCondition" = "TRIGGERED" indicating the NS instance affected by auto-scale or auto-terminate NS LCM operation(s) and the related "nsLcmOpOccId" attribute values of the high-priority and low-priority NS instances.

NOTE 4: The attributes "preemptingNsLcmOpOccId" (= higher priority "nsLcmOpOccId"), "highPrioNsInstanceId" (= higher priority "nsInstanceId") and the "affectedOpOccs[]" structure ("affectedNsId" = lower priority "nsInstanceId", "affectedOpOccId" = lower priority "nsLcmOpOccId" and "affectedCondition" = "TRIGGERED") are present in the notification, see clause 6.5.2.19.

- 10) The NFVO sends to the OSS/BSS NS lifecycle management operation occurrence notification(s) (see clause 6.3.6) with operationState "PROCESSING" to indicate the "start" of the "Scale NS(in)" or "Terminate NS" operations on the NS instance(s) of lower priority.

NOTE 5: These notifications are related to the NS instances that were forcefully scaled in or terminated. The operations release resources allocated to the NS instances of lower priority which become available to be allocated by the operation on the higher priority NS instance.

- 11) Once the auto-terminate or auto-scale(in) operations are finished, the NFVO sends to the OSS/BSS NS lifecycle management operation occurrence notifications (see clause 6.3.6) to indicate the completion of the lifecycle management operation occurrences with the success operationState "COMPLETED" as the "result" of the "Scale NS(in)" or "Terminate NS" operations on NS instances of lower priority that were forcefully scaled in or terminated.
- 12) The NFVO saves the attributes "affectedNsId" and "affectedOpOccId" related to the lower priority NS instance to send shortage notification(s) later when the shortage ends.

NOTE 6: The steps 9) and 10) can be executed in any order by the NFVO.

If enough resources are available after scaling in or terminating lower priority NS instances, the NFVO executes the steps 13) to 16):

- 13) The NFVO allocates the resources needed by the current "nsLcmOpOccId" of higher priority.
- 14) The NFVO executes the steps to Instantiate/Scale/Heal the NS instance of higher priority. See clause 6.3.3.
- 15) The NFVO sends to the OSS/BSS an NS lifecycle management operation occurrence notification ("NsLcmOperationOccurrenceNotification") with operationState "PROCESSING" to indicate the "start" of the "Instantiate NS"/"Scale NS(in)"/"Heal NS" operation on "nsLcmOpOccId" of higher priority.
- 16) When finished, the NFVO sends to the OSS/BSS an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the completion of the lifecycle management operation occurrence with the success operationState "COMPLETED" as the "result" of the LCM operation on "nsLcmOpOccId" of higher priority.

Alternatively, if not enough resources are available after scaling in or terminating lower priority NS instances, the steps 17) to 20) are executed:

- 17) The NS LCM operation occurrence on the high-priority NS instance fails for lack of resources, and the NFVO sets the current operationState of the NS LCM operation occurrence on the higher priority NS instance to "FAILED\_TEMP" or "ROLLED\_BACK".
- 18) The NFVO sends to the OSS/BSS an NS LCM Capacity Shortage notification ("NsLcmCapacityShortageNotification"), with "status" = "LCM\_RESOURCES\_NOT\_AVAILABLE" referring to the "nsLcmOpOccId" on the higher priority NS instance that could not be executed due to lack of resources.

NOTE 7: The attributes "preemptingNsLcmOpOccId" and "highPrioNsInstanceId" are not present in the notification, and the "affectedOpOccs" structure ("affectedNsId" = higher priority "nsInstanceId", "affectedOpOccId" = higher priority "nsLcmOpOccId" and "affectedCondition" = "PRE\_EMPTED") is present, see clause 6.5.2.19.

- 19) The NFVO sends to the OSS/BSS an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the unsuccessful completion of the lifecycle management operation occurrence with the operationState "FAILED\_TEMP" or "ROLLED\_BACK" as the "result" of the "Instantiate NS"/"Scale NS(in)"/"Heal NS" operation on the NS instance of higher priority. The order in which the response and the notification arrive at the OSS/BSS is not defined.
- 20) The NFVO saves the "nsLcmOpOccId" of higher priority NS instance to send shortage notification later when the shortage ends.

**Postcondition:** If the resource shortage could be resolved:

- The NS LCM operation occurrence of the NS instance with higher priority is accepted and has pre-empted one or more the NS instances with lower priority instantiated previously, by triggering NS LCM operations (scale-in or terminate) on the lower priority NS instances to confiscate resources from them to fulfil the resource needs of the NS LCM operation on the higher-priority NS instance.

- The OSS/BSS can request issuing new NS LCM operations on the NS instances with lower priority to re-establish the services or scale out the lower priority when the shortage ends, and the resources are available.

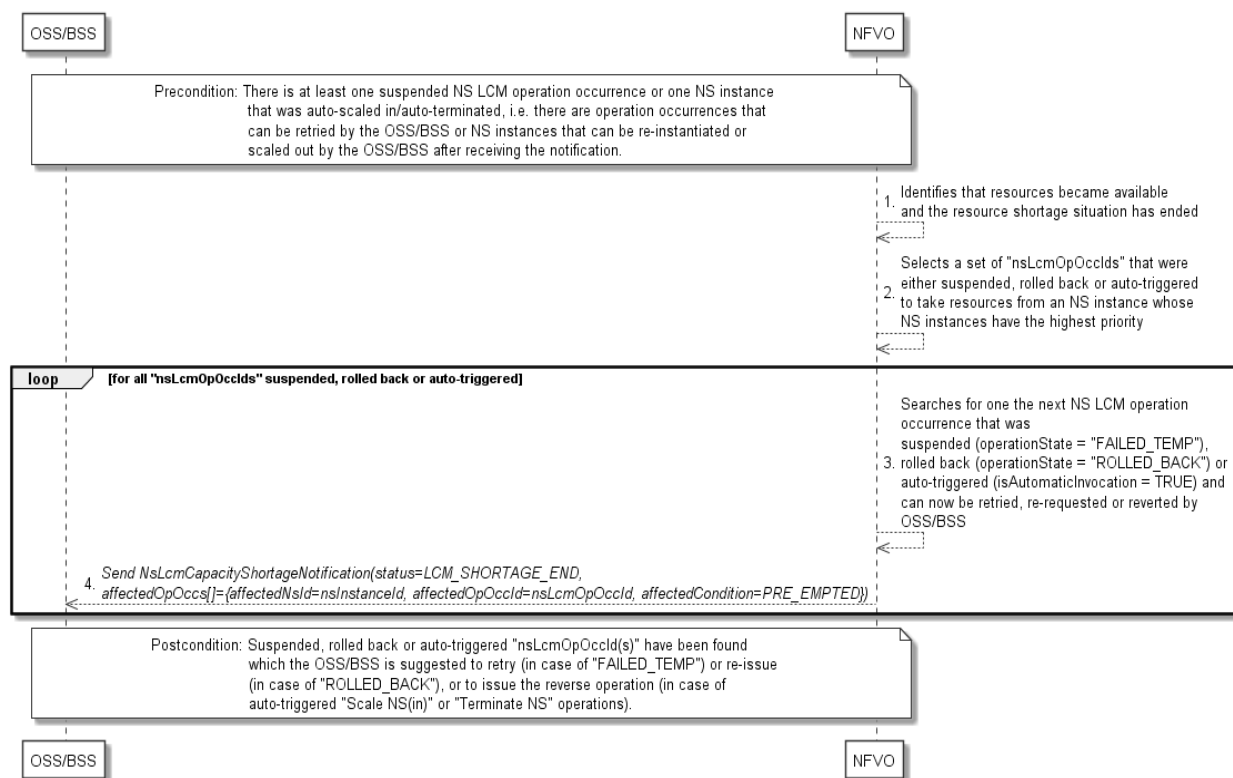
If the resource shortage could not be resolved:

- The NS LCM operation occurrence of the NS instance with higher priority is rejected and either in "FAILED\_TEMP" or "ROLLED\_BACK" state. The NS instances with lower priority continue instantiated and running.
- The OSS/BSS can request retrying the NS LCM operation occurrence on the higher priority NS instance when the shortage ends, and the resources are available, as long as it is in "FAILED\_TEMP" state. Operation occurrences in "ROLLED\_BACK" state cannot be retried; a new NS LCM operation request needs to be sent.

**Error handling:** If the NS lifecycle management operation fails, error information is provided in the notification message that reports the erroneous completion of the procedure and is also available in the resource that represents the actual NS lifecycle management operation occurrences.

### 6.3.3a.6 Shortage Use Case #4

Shortage Use Case #4: Shortage ends.



**Figure 6.3.3a.6-1: Flow of Shortage use case #4**

**Precondition:** There is at least one suspended NS LCM operation occurrence (FAILED\_TEMP, ROLLED\_BACK) or one NS instance that was auto-scaled in/auto-terminated, i.e. there are operation occurrences that can be retried or re-issued by the OSS/BSS or NS instances that can be requested to be re-instantiated or scaled out by the OSS/BSS after receiving the notification.

An NS lifecycle operation according to shortage use case #4, as illustrated in figure 6.3.3a-6-1, consists of the following steps:

- 1) The NFVO identifies that resources became available, and the resource shortage situation has ended.
- 2) The NFVO selects a set of "nsLcmOpOccIds" that were either suspended, rolled back or automatically triggered to take resources from an NS instance using suitable criteria, e.g. having the highest priority level, for which it believes that enough resources are available.

Through a loop iterating the set selected in step 2), for all saved "nsLcmOpOccIds" that represent operation occurrences which were suspended, rolled back or auto-triggerred to take resources from an NS instance the NFVO executes the steps 3) to 4):

- 3) The NFVO searches for one the next NS LCM operation occurrence that was suspended ("operationState" = "FAILED\_TEMP"), rolled back ("operationState" = "ROLLED\_BACK") or auto-triggerred ("isAutomaticInvocation" = TRUE) and can now be retried, re-requested or reverted by OSS/BSS, accepted and executed successfully.
- 4) For the LCM operation occurrence selected previously, the NFVO sends to the OSS/BSS NS LCM Capacity Shortage notifications ("NsLcmCapacityShortageNotification"), with "status" = "LCM\_SHORTAGE\_END".

NOTE 1: The attributes "preemptingNsLcmOpOccId" and "highPrioNsInstanceId" are not present in the notification, and the "affectedOpOccs[]" structure ("affectedNsId" = lower priority "nsInstanceId", "affectedOpOccId" = lower priority "nsLcmOpOccId" and "affectedCondition" = "PRE\_EMPTED") is present, see clause 6.5.2.19.

NOTE 2: The order in which the operation occurrences will be selected, as well as which ones will be retried/re-requested or reverted first by the OSS/BSS is not defined.

**Postcondition:** Suspended, rolled back or auto-triggerred "nsLcmOpOccId(s)" have been found which the OSS/BSS is suggested to retry (in case of "FAILED\_TEMP") or re-issue (in case of "ROLLED\_BACK"), or to issue the reverse operation (in case of automatically triggered "Scale NS(in)" or "Terminate NS" operations).

### 6.3.3b Flow of NS lifecycle management operations triggered by task resources with feasibility check

This clause provides sequences for NS Lifecycle Management operations that operate on NS instance resources and are triggered by task resources, when the feasibility checking capability is requested for the "Instantiate NS" and "Update NS" operations.

The feasibility checking capability allows NFVO to run a pre-check on the resources reservation during the execution of those NS LCM operations, the NFVO executes this flow with the feasibility checking procedure when the "feasibilityCheckReserve" attribute is present in the request structures and the attribute is offset to value other than "NO\_FEASIBILITY\_CHECK". If "feasibilityCheckReserve" attribute is not present or "feasibilityCheckReserve" attribute is set to "NO\_FEASIBILITY\_CHECK", the NFVO does not execute the feasibility checking procedure and the flow of NS lifecycle management operations defined in clause 6.3.3 is executed.

On receipt of a NS Lifecycle Management operation ("Instantiate NS" or "Update NS") with the "feasibilityCheckReserve" attribute, the NFVO notifies the OSS/BSS by sending the NS lifecycle management operation occurrence notification with the value of the "feasibilityCheckReserve" attribute received in the request structure to indicate the start of the feasibility checking procedure by the NFVO according to the procedure options:

- When the attribute is set to NO\_FEASIBILITY\_CHECK, the feasibility checking procedure is not performed. Resources might be reserved as triggered by means outside the present operation request (e.g. configuration or policies), e.g. if it is a scheduled NS instantiation operation. See clause A.2.3 in ETSI GS NFV-IFA 010 [2].
- If the attribute is set to FEASIBILITY\_CHECK\_ONLY, only feasibility checking procedure is performed for the operation, no resources are reserved.
- When the attribute is set to FEASIBILITY\_CHECK\_WITH\_OPERATION, the feasibility checking procedure is performed and based on the procedure result, the actual NS instantiation operation is performed at the time indicated by the "startTime" attribute of the "Instantiate NS" or by the "updateTime" attribute of the "Update NS" request data types, if provided. Resources might be reserved as triggered by means outside the present operation request (e.g. configuration or policies), e.g. if it is a scheduled NS instantiation operation. See clause A.2.3 in ETSI GS NFV-IFA 010 [2].
- If the attribute is set to FEASIBILITY\_CHECK\_WITH\_RESERVATION\_AND\_OPERATION, the feasibility checking procedure is performed and based on the procedure result, resources are reserved as part of the feasibility check, and the actual NS instantiation operation is performed using the resources reserved during the feasibility check, and at the time indicated by the "startTime" attribute of the "Instantiate NS" or by the "updateTime" attribute of the "Update NS" request data types, if provided.

After completing the procedure, the NFVO notifies the OSS/BSS by sending the NS lifecycle management operation occurrence notification. In case the feasibility checking procedure results in success, the notification includes "feasibilityCheckResult" with the specific result code to indicate the operation has been executed successfully. In case the feasibility checking procedure results in failure, the "error" attribute of the notification indicates the specific problem details.

Apart from the reservation capability explicitly requested along with feasibility checking procedure, NFV-MANO may use configuration, internal policies and SLAs (out of scope of standardization) to trigger the resources reservation that is outside the present operation request.

The OSS/BSS may send multiple operation requests and the NFVO runs the feasibility checking procedure for them simultaneously.

Figure 6.3.3b-1 illustrates the lifecycle management flow with feasibility check. Placeholders in this flow allow for differentiating between the Instantiate NS and Update NS operations and are marked with double angular brackets "<<...>>".

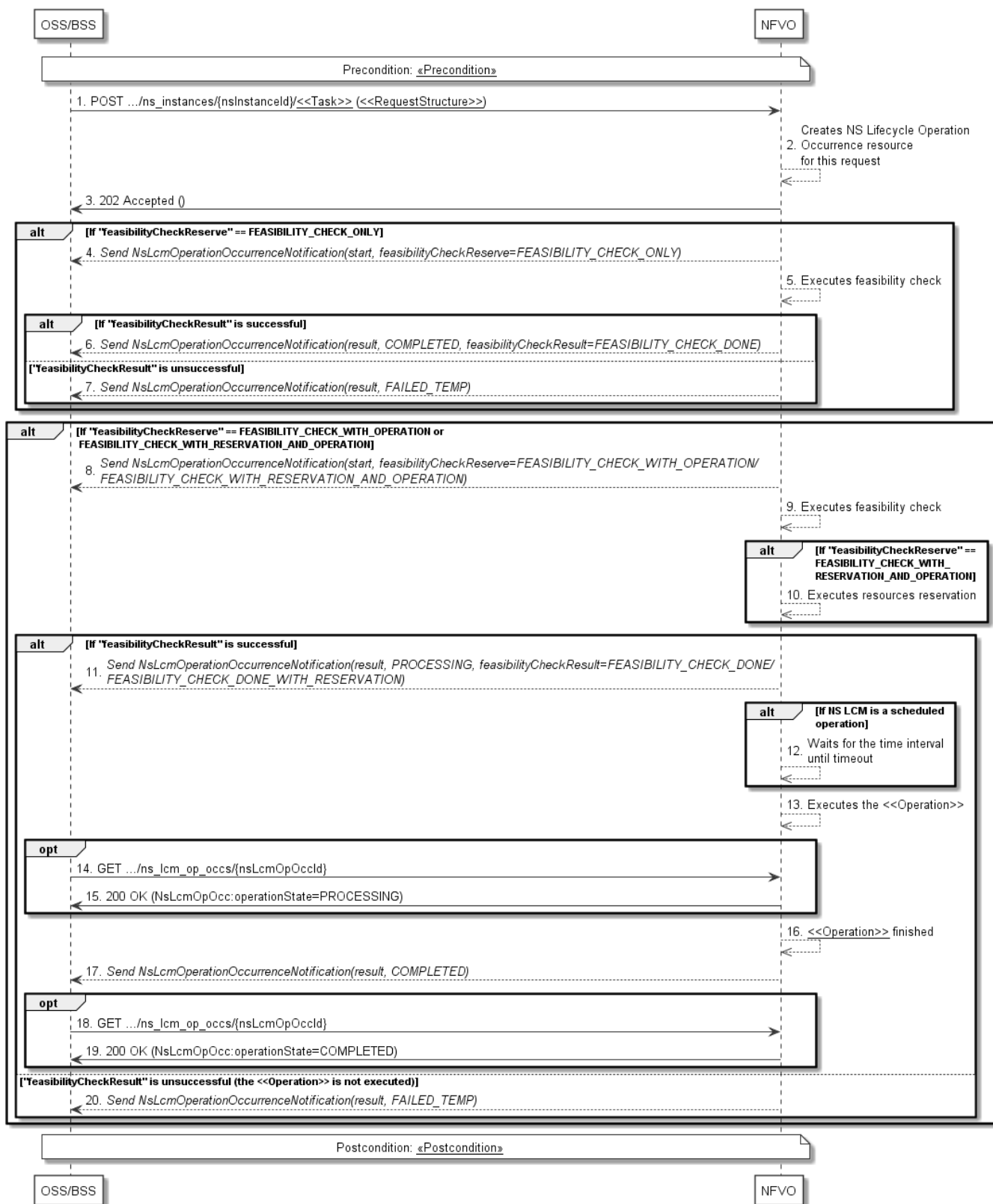


Figure 6.3.3b-1: Flow of NS lifecycle operations triggered by task resources with feasibility check

NOTE: Due to possible race conditions, the 202 response and the "start" NsLcmOperationOccurrenceNotification can arrive in any order at the OSS/BSS.

**Precondition:** The precondition depends on the actual operation (Instantiate NS or Update NS) and is described by the template parameter <<Precondition>>. Table 6.4.1.2-1 specifies the applicable precondition. The "feasibilityCheckReserve" attribute is present in the request and set to value other than "NO\_FEASIBILITY\_CHECK".

An NS lifecycle operation with feasibility check capability, as illustrated in figure 6.3.3b-1, consists of the following steps:

- 1) The OSS/BSS sends a POST request to the <<Task>> resource that represents the lifecycle operation to be executed on the NS instance and includes in the payload body a data structure of type <<RequestStructure>>. The name <<Task>> of the task resource and the <<RequestStructure>> depend on the operation and are described in table 6.4.1.2-1.  
  
The OSS/BSS sends in the request structure the "feasibilityCheckReserve" attribute according to the options available.
- 2) The NFVO creates a new "Individual NS Lifecycle Operation Occurrence" resource for the request.
- 3) The NFVO returns a "202 Accepted" response with an empty payload body and a "Location" HTTP header that points to the new "Individual NS Lifecycle Operation Occurrence" resource, i.e. it includes the URI of that resource which is ".../ns\_lcm\_op\_occs/{nsLcmOpOccId}".

The NFVO checks the value of the "feasibilityCheckReserve" attribute in the request structure, the NFVO executes the steps 4-20.

If the attribute value is equal to "FEASIBILITY\_CHECK\_ONLY", the NFVO executes the steps 4-7:

- 4) The NFVO sends to the OSS/BSS an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the start of the feasibility checking procedure execution with the attribute "feasibilityCheckReserve" = "FEASIBILITY\_CHECK\_ONLY" received in the request. See note.
- 5) The NFVO executes the feasibility checking procedure for the request.  
  
In this case only feasibility check is executed without performing the operation, the result of the operation indicates the outcome of the feasibility for the NS operation and includes no operation to be performed.
- 6) When finished, if the result is successful, the NFVO sends to the OSS/BSS an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the completion of the feasibility checking procedure execution with the success "feasibilityCheckResult" = "FEASIBILITY\_CHECK\_DONE" as the result of the feasibility checking procedure. In this case the NS LCM operation is not executed. See note.
- 7) In case of unsuccessful, the NFVO sends to the OSS/BSS an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the procedure has failed (FAILED\_TEMP) and additional information on the failure is reported in the "error" attribute of the NsLcmOperationOccurrenceNotification structure along with the result. The OSS/BSS can try the feasibility checking procedure later. See note.

If the attribute value is equal to "FEASIBILITY\_CHECK\_WITH\_OPERATION" or "FEASIBILITY\_CHECK\_WITH\_RESERVATION\_AND\_OPERATION", the NFVO executes the steps 8-10:

- 8) The NFVO sends to the OSS/BSS an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the start of the feasibility checking procedure execution with the attribute "feasibilityCheckReserve" = "FEASIBILITY\_CHECK\_WITH\_OPERATION" or "FEASIBILITY\_CHECK\_WITH\_RESERVATION\_AND\_OPERATION" received in the request, and to indicate the start of the NS lifecycle management operation occurrence. See note.
- 9) The NFVO executes the feasibility checking procedure for that request.  
  
The feasibility check and the operation are executed in this case, the result of the operation indicates the outcome of the feasibility for the NS operation and includes the NS operation to be processed with the result.
- 10) If the value of the attribute value is equal to "FEASIBILITY\_CHECK\_WITH\_RESERVATION\_AND\_OPERATION", the NFVO executes the resources reservation procedure for the request.

If the feasibility checking procedure is successful the NFVO executes the step 11-19:

- 11) When finished, if the result is successful, the NFVO sends to the OSS/BSS an NS lifecycle management operation occurrence notification (see clause 6.3.6) with "operationState" = "PROCESSING" and feasibility checking procedure execution with the success "feasibilityCheckResult" = "FEASIBILITY\_CHECK\_DONE" or "FEASIBILITY\_CHECK\_DONE\_WITH\_RESERVATION" as the result of the feasibility checking procedure. See note.

- 12) If the NS lifecycle management operation is a scheduled operation, i.e. the "startTime" attribute or the "updateTime" attribute is present in the request, the NFVO waits for the time interval until timeout.
- 13) The NFVO executes the <<Operation>>.
- 14) If desired, the NFVO can poll the "Individual NS Lifecycle Operation Occurrence" resource to obtain information about the ongoing operation by sending a GET request to the resource that represents the NS Lifecycle Operation Occurrence.
- 15) In the response to that request, the NFVO returns to the OSS/BSS information of the operation, such as the operation status, by providing in the payload body a data structure of type "NsLcmOpOcc".
- 16) The NFVO has finished the operation <<Operation>>.
- 17) The NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the completion of the lifecycle management operation occurrence with the success state "COMPLETED".
- 18) If desired, the OSS/BSS can send a new GET request to the "Individual NS Lifecycle Operation Occurrence" resource.
- 19) In the response to that request, the NFVO returns to the OSS/BSS information about the result of the operation, by providing in the payload body a data structure of type "NsLcmOpOcc".
- 20) In case of unsuccessful, the NFVO sends to the OSS/BSS an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the procedure has failed (FAILED\_TEMP) and additional information on the failure is reported in the "error" attribute of the NsLcmOperationOccurrenceNotification structure along with the result. The OSS/BSS can try the feasibility checking procedure later. See note.

**Postcondition:** The postcondition depends on the actual operation and is described by the template parameter <<Postcondition>>. Table 6.4.1.2-1 specifies the applicable postcondition.

The feasibility check is executed, or feasibility check and NS LCM operation are executed, or feasibility check, resources reservation and NS LCM operation are executed.

**Error handling:** If the NS lifecycle management operation fails, error information is provided in the notification message that reports the erroneous completion of the procedure and is also available in the resource that represents the actual NS lifecycle management operation occurrence related to this NS lifecycle management operation.

Table 6.4.1.2-1 defines how the flow described above is parameterized for the different NS lifecycle management operations.

### 6.3.4 Flow of the get operations status operation

This clause describes a sequence for obtaining the status of an NS lifecycle management operation occurrence.

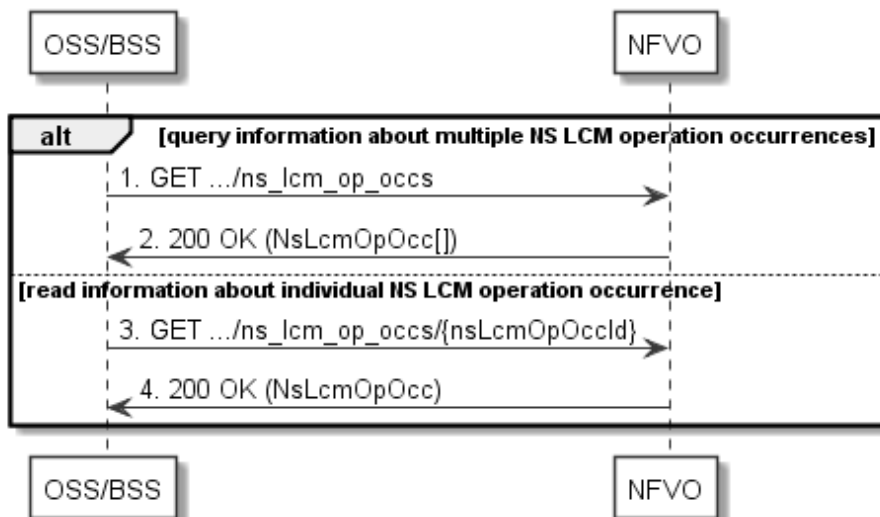


Figure 6.3.4-1: Flow of get NS lifecycle operation status



Obtaining the NS lifecycle operation status, as illustrated in figure 6.3.4-1, consists of the following steps:

- 1) If the OSS/BSS intends to query all NS lifecycle management operation occurrences, it sends a GET request to the "NS LCM operation occurrences" resource.
- 2) The NFVO returns a "200 OK" response to the OSS/BSS, and includes zero or more data structures of type "NsLcmOpOcc" in the payload body.
- 3) If the OSS/BSS intends to read information about a particular NS LCM operation occurrence, it sends a GET request to the "Individual NS LCM operation occurrence" resource, addressed by the appropriate NS LCM operation occurrence identifier in its resource URI.
- 4) The NFVO returns a "200 OK" response to the OSS/BSS, and includes one data structure of type "NsLcmOpOcc" in the payload body.

**Error Handling:** In case of failure, appropriate error information is provided in the response.

### 6.3.5 Flow of managing subscriptions

This clause describes the procedure for creating, querying/reading, and terminating subscriptions to notifications related to NS lifecycle management.

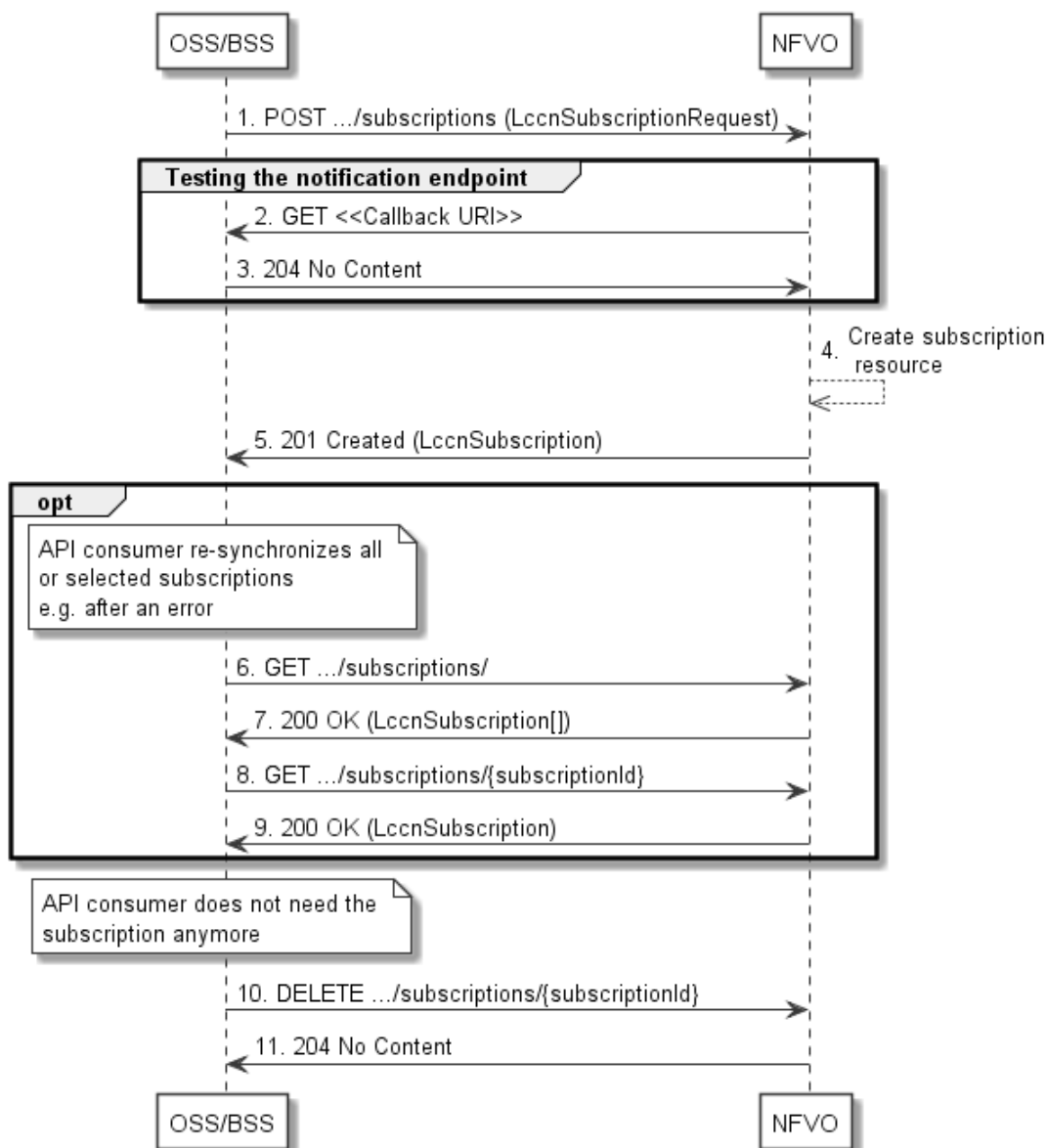


Figure 6.3.5-1: Flow of managing subscriptions

The procedure consists of the following steps as illustrated in figure 6.3.5-1:

- 1) The OSS/BSS sends a POST request to the "Subscriptions" resource including in the payload body a data structure of type "LccnSubscriptionRequest". That data structure contains filtering criteria and a callback URI to which the NFVO will subsequently send notifications about events that match the filter.
- 2) To test the notification endpoint that has been registered by the OSS/BSS as part of the subscription, the NFVO sends a GET request to the notification endpoint URI.
- 3) The OSS/BSS returns a "204 No Content" response to indicate success.
- 4) The NFVO creates a new subscription to notifications related to NS lifecycle changes, and an "Individual subscription" resource that represents this subscription.
- 5) The NFVO returns a "201 Created" response containing a data structure of type "LccnSubscription" representing the "Individual subscription" resource just created by the NFVO, and provides the URI of the newly-created resource in the "Location" HTTP header.
- 6) If desired, e.g. to recover from an error situation, the OSS/BSS can query information about its subscriptions by sending a GET request to the resource representing the subscriptions.
- 7) In that case, the NFVO returns a "200 OK" response that contains zero or more representations of all existing subscriptions that were created by the OSS/BSS.
- 8) If desired, e.g. to recover from an error situation, the OSS/BSS can read information about a particular subscription by sending a GET request to the resource representing that individual subscription.
- 9) In that case, the NFVO returns a "200 OK" response that contains a representation of that individual subscription.
- 10) If the OSS/BSS does not need the subscription anymore, it terminates the subscription by sending a DELETE request to the resource that represents the individual subscription to remove.
- 11) The OSS/BSS acknowledges the successful termination of the subscription by returning a "204 No Content" response.

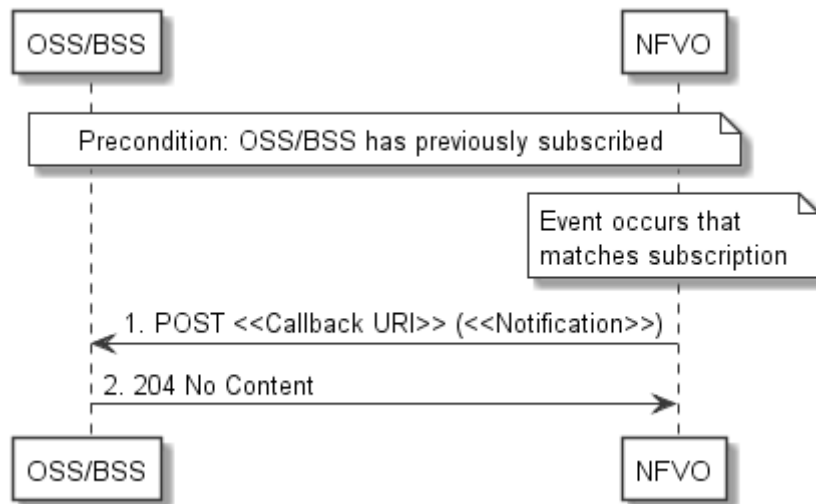
**Error handling:** The NFVO rejects a subscription if the subscription information is not valid: endpoint cannot be reached, subscription information is malformed, etc.

## 6.3.6 Flow of sending notifications

This clause describes the procedure for sending notifications related to NS lifecycle management.

NOTE 1: Notifications merely report to subscribed NFV-MANO entities the state changes of an NS instance and/or NS LCM operation occurrence. They are triggered during the execution of the operation's flow or at its end but have no impact on the course of the procedure that has triggered them or on the state of the NS instance. If this flow is invoked as part of another flow, the invoking procedure does not wait for the acknowledgement of the delivery of the notification.

NOTE 2: Race conditions between LCM operation requests/responses on one hand and notification delivery requests/responses on the other hand can occur as these are delivered through different HTTP connections.



**Figure 6.3.6-1: Flow of sending notifications**

**Precondition:** The OSS/BSS has subscribed previously to notifications related to NS lifecycle management.

The procedure consists of the following steps as illustrated in figure 6.3.6-1:

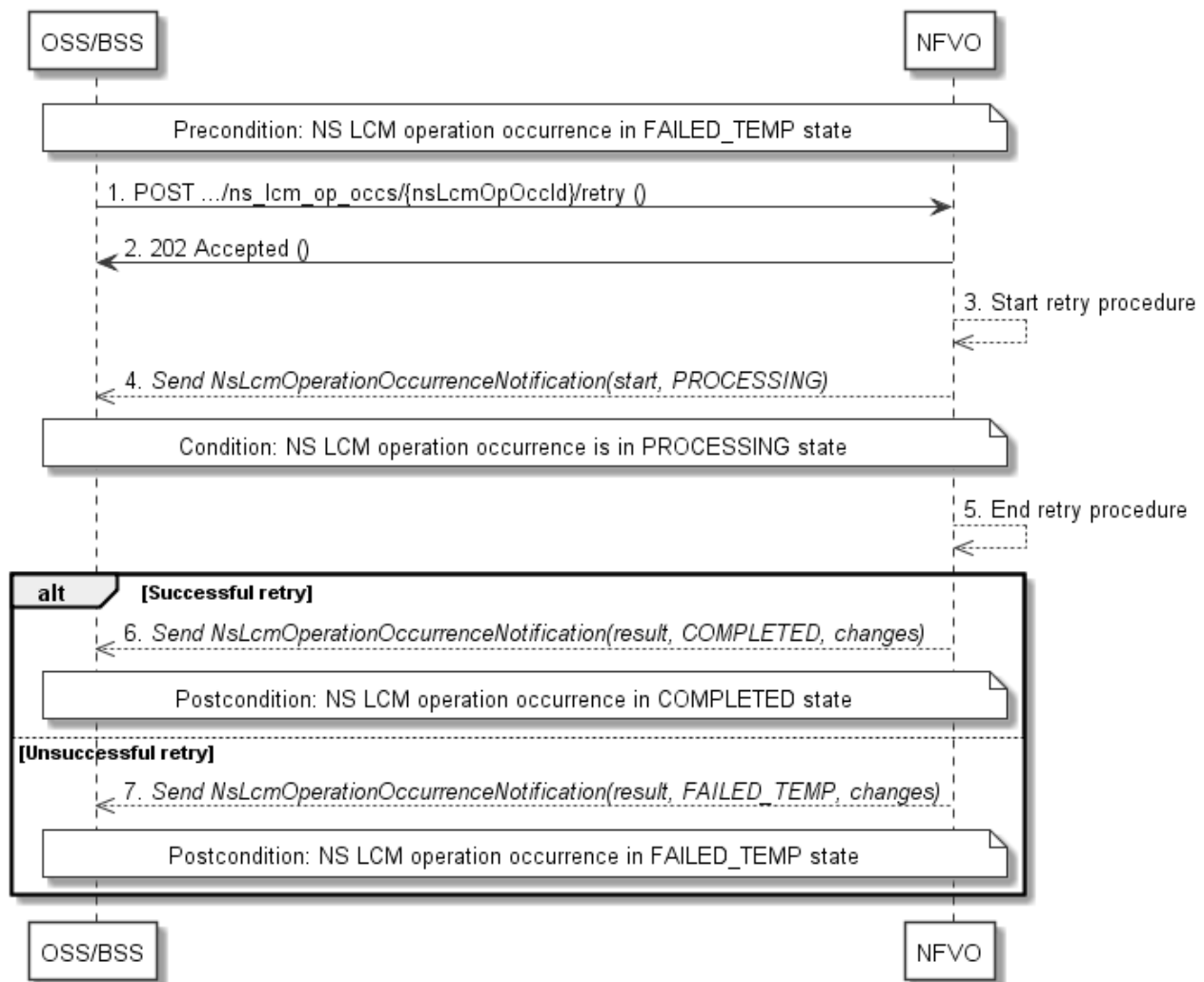
- 1) If an event occurs that matches the filtering criteria which are part of the subscription, the NFVO generates a notification that includes information about the event, and sends it in the body of a POST request to the URI which the OSS/BSS has registered as part of the subscription request. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API (see clauses 6.5.2.5, 6.5.2.6, 6.5.2.7, 6.5.2.8 and 6.5.2.19).
- 2) The OSS/BSS acknowledges the successful delivery of the notification by returning a "204 No Content" response.

**Error handling:** If the NFVO does not receive the "204 No Content" response from the OSS/BSS, it can retry sending the notification.

### 6.3.7 Flow of retrying an NS lifecycle management operation

This clause describes a sequence for retrying an NS lifecycle management operation occurrence that is represented by an "Individual NS LCM operation occurrence" resource. Retry is used if an operation is in FAILED\_TEMP state, and there is reason to believe that the operation will eventually succeed when retried, for instance because obstacle that led to an error during the execution of the LCM operation have been removed by an automated procedure, or by manual intervention. The "retry" operation is also called "idempotent retry" because it is possible to invoke retry multiple times, without side effects.

A comprehensive description of the handling of NS lifecycle management errors is provided in clause 6.6.



**Figure 6.3.7-1: Flow of retrying an NS lifecycle management operation**

NOTE: Due to possible race conditions, the 202 response and the "PROCESSING" NsLcmOperationOccurrenceNotification can arrive in any order at the OSS/BSS.

**Precondition:** The NS lifecycle operation occurrence is in FAILED\_TEMP state.

Retrying an NS lifecycle operation, as illustrated in figure 6.3.7-1, consists of the following steps:

- 1) The OSS/BSS sends a POST request with an empty body to the "Retry operation task" resource of the NS LCM operation occurrence that is to be retried.
- 2) The NFVO returns a "202 Accepted" response.
- 3) The NFVO starts the retry procedure.
- 4) The NFVO sends an NS lifecycle management operation occurrence notification of type "start" to indicate that the NS LCM operation occurrence enters the "PROCESSING" state.
- 5) The NFVO finishes the retry procedure.
- 6) On successful retry, the NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate successful completion of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs).
- 7) On unsuccessful retry, the NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate an intermediate error (retry failed) of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs).

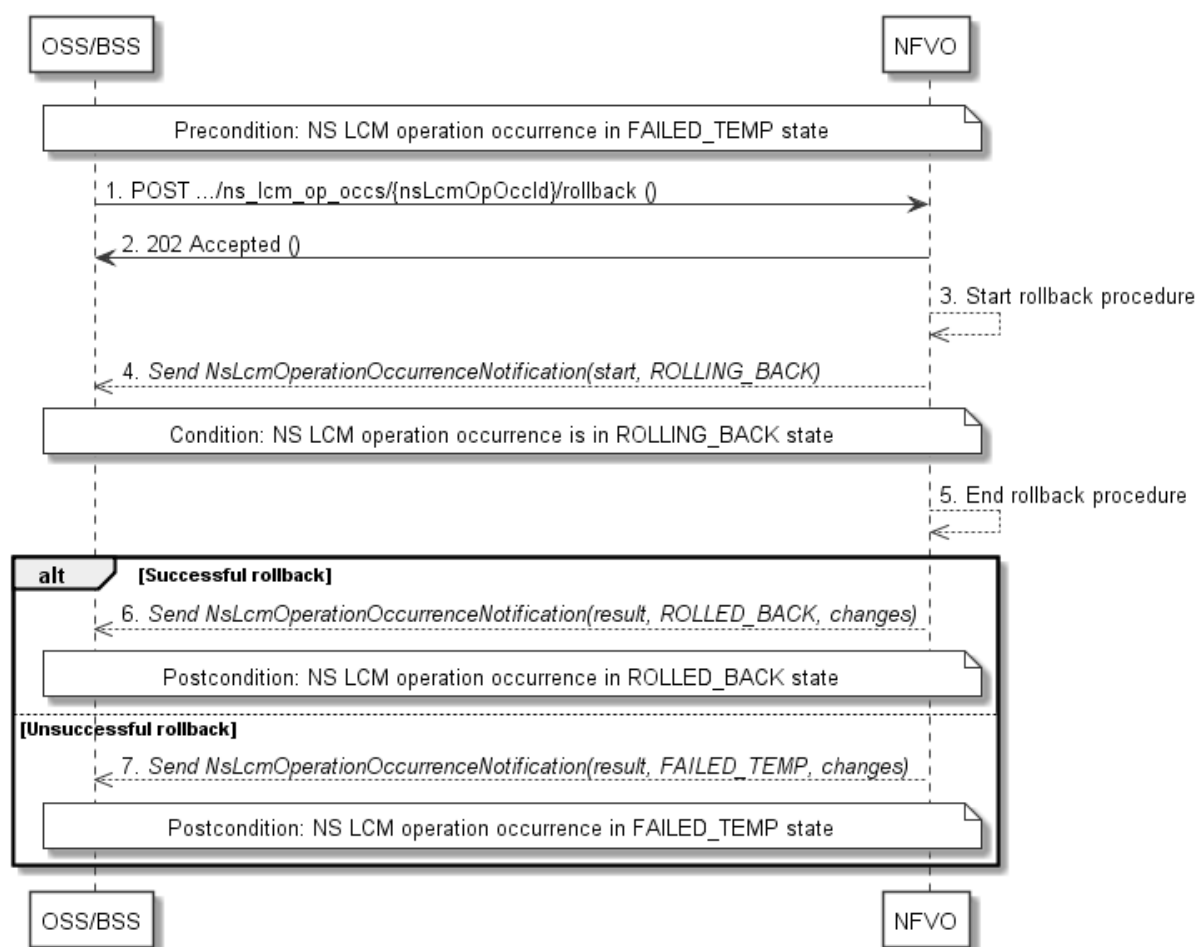
**Postcondition:** The NS lifecycle operation occurrence resource is in one of the following states: FAILED\_TEMP, COMPLETED. COMPLETED is a terminal state (see clause 6.6.2.2).

**Error handling:** The operation is rejected in case the "Individual NS LCM operation occurrence" resource is in any other state than FAILED\_TEMP, or in case Retry is not supported by for the particular NS LCM operation for the particular NS.

### 6.3.8 Flow of rolling back an NS lifecycle management operation

This clause describes a sequence for rolling back an NS lifecycle management operation occurrence that is represented by an "Individual NS LCM operation occurrence" resource. Rollback can be used for example if an operation is in FAILED\_TEMP state, and there is no reason to believe that retrying the operation will eventually succeed.

A comprehensive description of the handling of NS lifecycle management errors is provided in clause 6.6.



**Figure 6.3.8-1: Flow of rolling back an NS lifecycle management operation**

**NOTE:** Due to possible race conditions, the 202 response and the "ROLLING\_BACK" NsLcmOperationOccurrenceNotification can arrive in any order at the OSS/BSS.

**Precondition:** The NS lifecycle operation occurrence is in FAILED\_TEMP state.

Initiating the rollback of an NS lifecycle management operation, as illustrated in figure 6.3.8-1, consists of the following steps:

- 1) The OSS/BSS sends a POST request with an empty body to the "Rollback operation task" resource of the NS LCM operation occurrence that is to be rolled back.
- 2) The NFVO returns a "202 Accepted" response.
- 3) The NFVO starts the rollback procedure.

- 4) The NFVO sends an NS lifecycle management operation occurrence notification of type "start" to indicate that the NS LCM operation occurrence enters the "ROLLING\_BACK" state.
- 5) The NFVO finishes the rollback procedure.
- 6) On successful rollback, the NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate successful completion of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs).
- 7) On unsuccessful retry, the NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate an intermediate error (rollback failed) of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs).

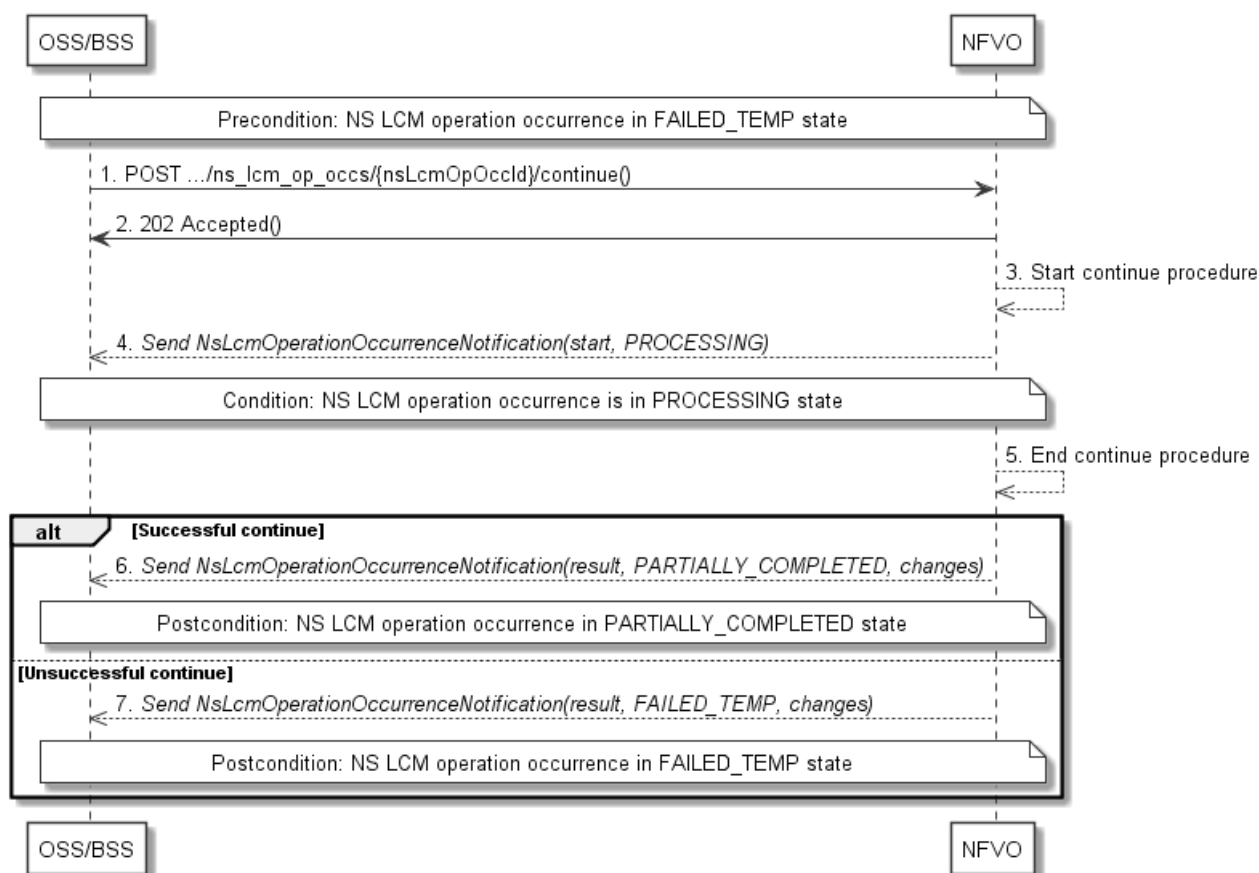
**Postcondition:** The NS lifecycle operation occurrence resource is in one of the following states: FAILED\_TEMP, ROLLED\_BACK. ROLLED\_BACK is a terminal state (see clause 6.6.2.2).

**Error handling:** The operation is rejected in case the NS lifecycle operation occurrence resource is in any other state than FAILED\_TEMP, or in case Rollback is not supported for the particular NS LCM operation for the particular NS.

### 6.3.9 Flow of continuing an NS lifecycle management operation

This clause describes a sequence for continuing an NS lifecycle management operation occurrence that is represented by an "Individual NS LCM operation occurrence" resource. Continue is used if an operation is in FAILED\_TEMP state, and there is reason to believe that the current operation can continue despite the error. The error can be fixed later, typically after current NS lifecycle management operation finishes.

A comprehensive description of the handling of NS lifecycle management errors is provided in clause 6.6.



**Figure 6.3.9-1: Flow of continuing an NS lifecycle management operation**

**NOTE:** Due to possible race conditions, the 202 response and the "PROCESSING" NsLcmOperationOccurrenceNotification can arrive in any order at the OSS/BSS.

**Precondition:** The NS lifecycle operation occurrence is in FAILED\_TEMP state.

Continuing an NS lifecycle operation, as illustrated in figure 6.3.9-1, consists of the following steps:

- 1) The OSS/BSS sends a POST request with an empty body to the "Continue operation task" resource of the NS LCM operation occurrence that is to be retried.
- 2) The NFVO returns a "202 Accepted" response.
- 3) The NFVO starts the continue procedure.
- 4) The NFVO sends an NS lifecycle management operation occurrence notification of type "start" to indicate that the NS LCM operation occurrence enters the "PROCESSING" state.
- 5) The NFVO finishes the continue procedure.
- 6) On successful continue, the NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate successful completion of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs).
- 7) On unsuccessful continue, the NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate an intermediate error (continue failed) of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs).

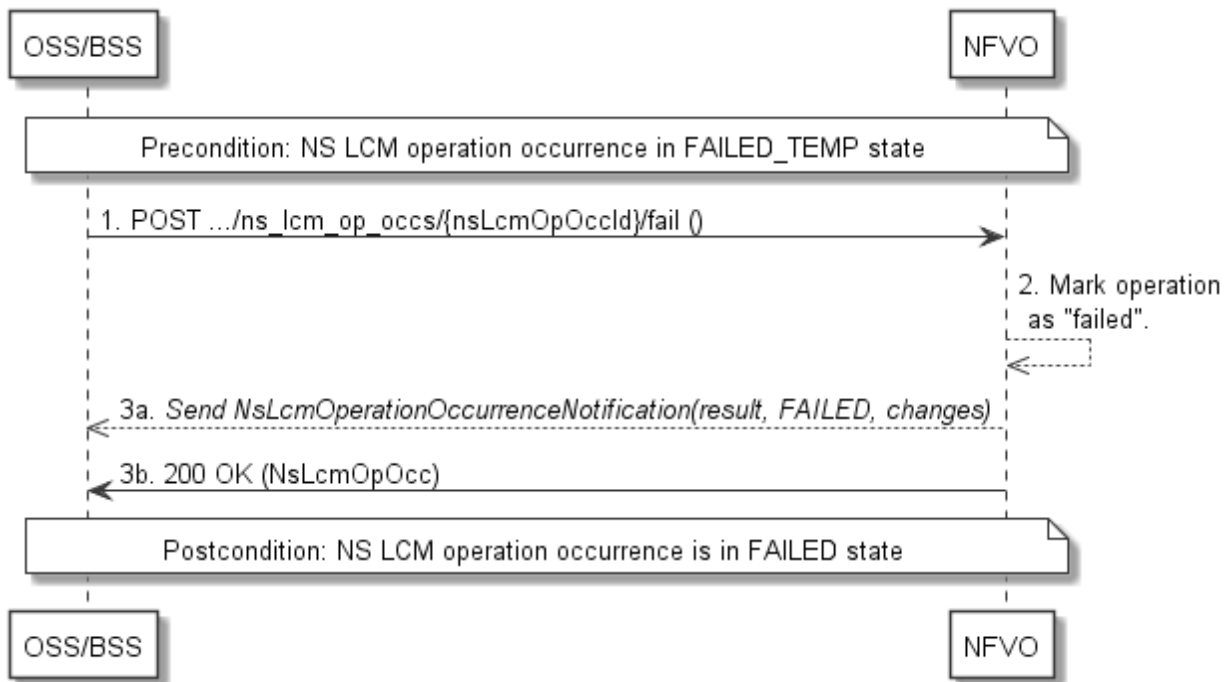
**Postcondition:** The NS lifecycle operation occurrence resource is in one of the following states: FAILED\_TEMP, PARTIALLY\_COMPLETED. PARTIALLY\_COMPLETED is a terminal state (see clause 6.6.2.2).

**Error handling:** The operation is rejected in case the "NS LCM operation occurrence" resource is in any other state than FAILED\_TEMP, or in case Continue is not supported for the particular NS LCM operation for the particular NS.

### 6.3.10 Flow of failing an NS lifecycle management operation

This clause describes a sequence for declaring as "failed" an NS lifecycle management operation occurrence that is represented by an "Individual NS LCM operation occurrence" resource. If there is neither an assumption that the operation can eventually succeed after further retries, nor that the operation can be successfully rolled back, the operation can be declared as "failed". This will unblock the invocation of other LCM operations, such as HealNs, or non-graceful NS termination, on the affected NS instance.

A comprehensive description of the handling of NS lifecycle management errors is provided in clause 6.6.



**Figure 6.3.10-1: Flow of declaring an NS lifecycle management operation as failed**

**NOTE:** Due to possible race conditions, the 200 response and the "FAILED" NsLcmOperationOccurrenceNotification can arrive in any order at the OSS/BSS.

**Precondition:** The NS lifecycle operation occurrence is in FAILED\_TEMP state.

Declaring an NS lifecycle management operation as failed, as illustrated in figure 6.3.10-1, consists of the following steps:

- 1) The OSS/BSS sends a POST request with an empty body to the "Fail operation task" resource of the NS LCM operation occurrence that is to be marked as failed.
- 2) The NFVO marks the operation as failed.
- 3) The NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the final failure of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs). Furthermore, it returns a "200 OK" response, and includes in the body an NsLcmOpOcc structure. The order in which the response and the notification arrive at the OSS/BSS is not defined.

**Postcondition:** The NS lifecycle operation occurrence resource is FAILED state. This is a terminal state (see clause 6.6.2.2).

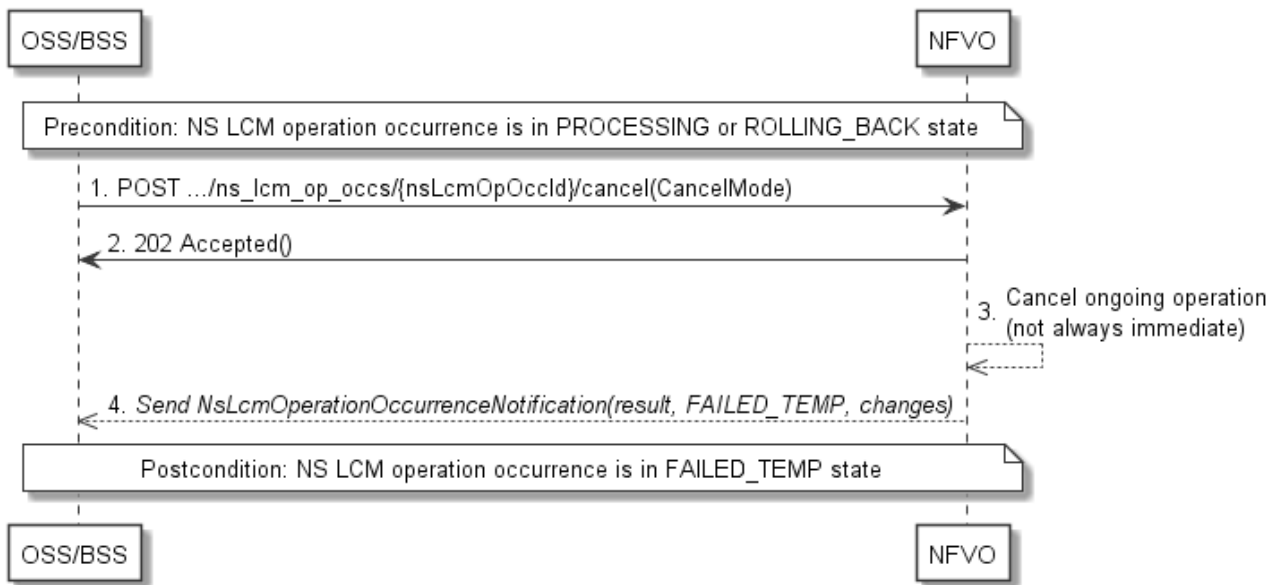
**Error handling:** The operation is rejected in case the NS lifecycle operation occurrence resource is in any other state than FAILED\_TEMP.

### 6.3.11 Flow of cancelling an NS lifecycle management operation

This clause describes a sequence for cancelling an ongoing NS LCM operation occurrence, or a rollback of an NS LCM operation occurrence. The possibility and timing of cancellation is dependent on the implementation of the underlying lifecycle management operation.

A comprehensive description of the handling of NS lifecycle management errors is provided in clause 6.6.





**Figure 6.3.11-1: Flow of cancelling an NS lifecycle management operation in "PROCESSING" or "ROLLING\_BACK" state**

**NOTE:** Due to possible race conditions, the 202 response and the "FAILED\_TEMP" NsLcmOperationOccurrenceNotification can arrive in any order at the OSS/BSS.

**Precondition:** The NS lifecycle operation occurrence is in PROCESSING or ROLLING\_BACK state.

Cancelling an NS lifecycle operation when it is in "PROCESSING" or "ROLLING\_BACK" state, as illustrated in figure 6.3.11-1, consists of the following steps:

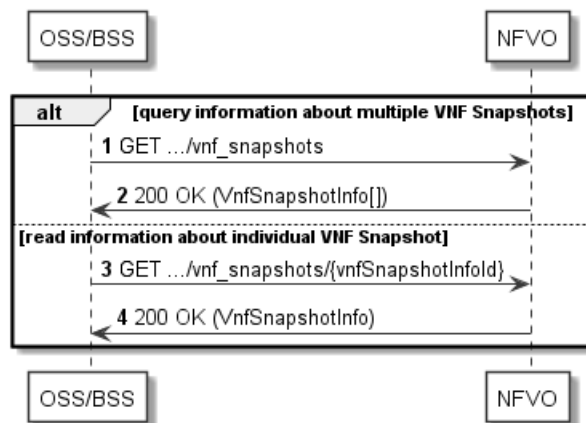
- 1) The OSS/BSS sends a POST request with a "CancelMode" structure in the body to the "Cancel operation task" resource of the NS LCM operation occurrence that is to be cancelled.
- 2) The NFVO returns a "202 Accepted" response.
- 3) The NFVO cancels the ongoing LCM operation. This can take some time.
- 4) The NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate an intermediate error (cancelled) of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs).

**Postcondition:** The NS lifecycle management operation occurrence resource is in FAILED\_TEMP state.

**Error handling:** The operation is rejected in case the NS lifecycle operation occurrence is in any other state than PROCESSING or ROLLING\_BACK, or in case Cancel is not supported for the particular NS LCM operation for the particular NS.

## 6.3.12 Flow of the query VNF snapshot operation

This clause describes a sequence for querying/reading information about one or more VNF snapshots.



**Figure 6.3.12-1: Flow of VNF snapshot query/read**

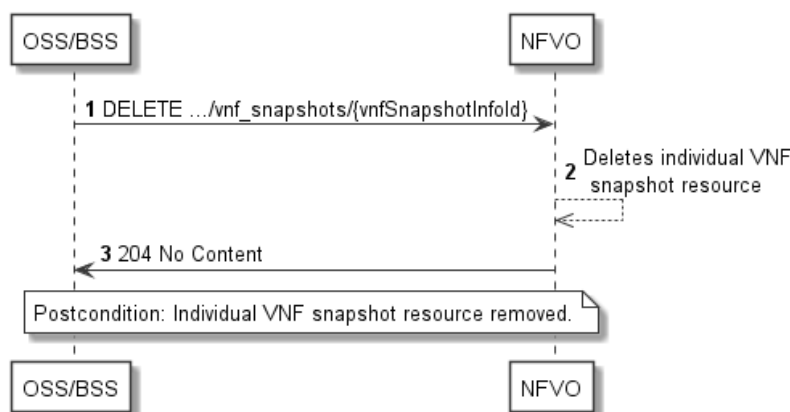
VNF snapshot query/read, as illustrated in figure 6.3.12-1, consists of the following steps:

- 1) If the OSS/BSS intends to query all snapshots, it sends a GET request to the "VNF snapshots" resource.
- 2) The NFVO returns a "200 OK" response to the OSS/BSS and includes zero or more data structures of type "VnfSnapshotInfo" in the payload body.
- 3) If the OSS/BSS intends to read information about a particular snapshot, it sends a GET request to the "Individual VNF snapshot" resource, addressed by the appropriate snapshot information identifier in its resource URI.
- 4) The NFVO returns a "200 OK" response to the OSS/BSS and includes one data structure of type "VnfSnapshotInfo" in the payload body.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 6.3.13 Flow of the deletion of a VNF snapshot resource

This clause describes the procedure for the deletion of a VNF snapshot resource.



**Figure 6.3.13-1: Flow of the deletion of a VNF snapshot resource**

The procedure consists of the following steps as illustrated in figure 6.3.13-1:

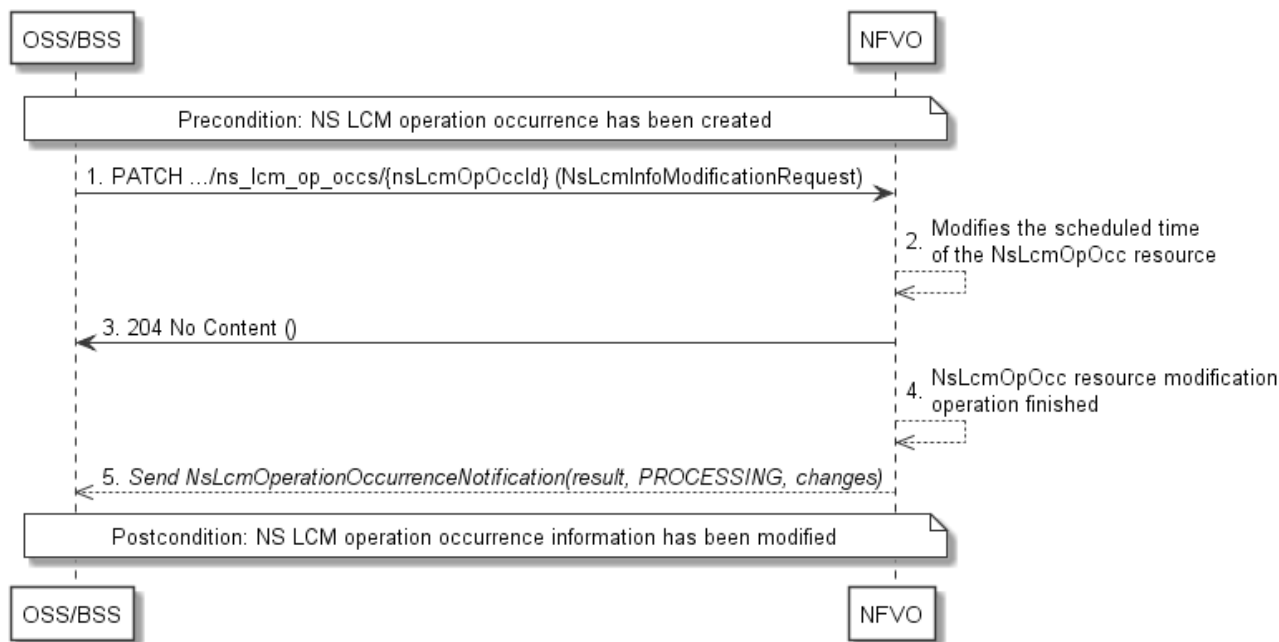
- 1) OSS/BSS sends a DELETE request to the "Individual VNF snapshot" resource.
- 2) The NFVO deletes the VNF snapshot resource and the associated VNF snapshot interacting with the corresponding VNF.
- 3) The NFVO returns a "204 No Content" response with an empty payload body.

**Postcondition:** The resource representing the VNF snapshot has been removed from the list of VNF snapshot resources, and the VNF snapshot has been deleted.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 6.3.14 Flow of the Modify an Individual NS LCM Information operation

This clause describes a sequence for updating information about an "Individual NS LCM operation occurrence" resource.



**Figure 6.3.14-1: Flow of the modification of NS LCM information**

**NOTE:** Due to possible race conditions, the 204 response and the NsLcmOperationOccurrenceNotification can arrive in any order at the OSS/BSS.

**Precondition:** The resource representing the NS LCM operation occurrence has been created.

Updating an "Individual NS LCM operation occurrence" resource information, as illustrated in figure 6.3.14-1, consists of the following steps:

- 1) The OSS/BSS sends a PATCH request to the "Individual NS LCM operation occurrence" resource, addressed by the appropriate NS LCM operation occurrence identifier in its resource URI that is to be operated to update the scheduled time, and includes in the payload body a data structure of type "NsLcmInfoModificationRequest".
- 2) The NFVO modifies the scheduled time (i.e. the timer defined by the "startTime" attribute) of the "Individual NsLcmOpOcc" resource for the request.
- 3) The NFVO returns a "204 No Content" response code to the OSS/BSS with an empty payload body for successfully changing the information of the "Individual NsLcmOpOcc" resource. See note.
- 4) The NFVO has finished the "Individual NS LCM operation occurrence" resource modification operation.
- 5) The NFVO sends a lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the completion of the modification operation, with "operationState" = "PROCESSING" and the performed changes. See note.

**Postcondition:** Upon successful completion, the information of the "Individual NS LCM operation occurrence" is updated.

**Error handling:** If the updating of NS LCM operation occurrence information fails, error information is provided in the notification message that reports the erroneous completion of the procedure.

## 6.4 Resources

### 6.4.1 Introduction

#### 6.4.1.1 Overview

This clause defines all the resources and methods provided by the NS lifecycle management interface.

#### 6.4.1.2 Task resources that trigger NS LCM operations

A number of resources are defined as task resources to trigger NS LCM operations that are potentially long-running (e.g. Instantiate NS, Scale NS). To represent each occurrence of such an NS LCM operation, an "Individual NS LCM operation occurrence" resource is created as defined in clause 6.3.3.

When successfully executing the POST method on a task resource that triggers an NS LCM operation, asynchronous processing of the request is started, which shall include the following:

- 1) Before returning the "202 Accepted" response to the POST method, a new "Individual NS LCM operation occurrence" resource as defined in clause 6.4.10 shall be created, which represents the underlying NS LCM operation occurrence that is executed by the NFVO. The NFVO shall set the "operationState" in the representation of the "Individual NS LCM operation occurrence" resource to "PROCESSING".
- 2) Notifications of type "NsLcmOperationOccurrenceNotification" shall be triggered as part of executing the underlying NS LCM operation occurrence as defined in clauses 6.5.2.5 and 6.6.2.
- 3) If the NFVO has successfully completed the underlying NS LCM operation occurrence:
  - a) It shall update the representation of the "Individual NS instance" resource which has been changed by the LCM operation to reflect the result of the operation. For individual operations, specific additional conditions can be specified in the following clauses, if applicable.
  - b) It shall set the "operationState" attribute in the representation of the aforementioned "Individual NS LCM operation occurrence" resource to the value "COMPLETED", and shall reflect the changes performed during the LCM operation in the representation of that resource.
  - c) To indicate success, the NFVO shall send a notification of type "NsLcmOperationOccurrenceNotification" with the "operationState" attribute set to "COMPLETED" as defined in clause 6.6.2.
- 4) If executing the underlying NS LCM operation occurrence by the NFVO has failed in the "PROCESSING" phase, which NFVO knows that the NS instance can be brought into a consistent state by immediately rolling back the operation, the NFVO shall send a notification of type "NsLcmOperationOccurrenceNotification" with the "operationState" attribute set to "ROLLING\_BACK" as defined in clause 6.6.2. It shall also set the "operationState" attribute in the representation of the aforementioned "Individual NS LCM operation occurrence" resource to the value "ROLLING\_BACK".
- 5) If executing the underlying NS LCM operation occurrence by the NFVO has failed with no option to recover, the NFVO shall send a notification of type "NsLcmOperationOccurrenceNotification" with the "operationState" attribute set to "FAILED" as defined in clause 6.6.2. It shall also set the "operationState" attribute in the representation of the aforementioned "Individual NS LCM operation occurrence" resource to the value "FAILED", and shall reflect, at its best knowledge, the changes performed during the LCM operation.

- 6) If executing the underlying NS LCM operation occurrence by the NFVO has failed temporarily, the NFVO shall send a notification of type "NsLcmOperationOccurrenceNotification" with the "operationState" attribute set to "FAILED\_TEMP" as defined in clause 6.6.2. It shall also set the "operationState" attribute in the representation of the aforementioned "Individual NS LCM operation occurrence" resource to the value "FAILED\_TEMP", and shall reflect, at its best knowledge, the changes performed so far during the LCM operation in the representation of that resource.
- 7) If executing the underlying NS LCM operation occurrence by the NFVO has failed, which the NFVO knows that the failure is acceptable and continues the NS LCM till it finishes, the NFVO shall send a notification of type "NsLcmOperationOccurrenceNotification" with the "operationState" attribute set to "PARTIALLY\_COMPLETED" as defined in clause 6.6.2. It shall also set the "operationState" attribute in the representation of the aforementioned "Individual NS LCM operation occurrence" resource to the value "PARTIALLY\_COMPLETED", and shall reflect, at its best knowledge, the changes performed so far during the LCM operation in the representation of that resource has been partially completed with acceptable errors.

The preconditions and postconditions for a successful execution of each of the NS lifecycle management operations triggered by the corresponding task resources shall be as defined in table 6.4.1.2-1.

**Table 6.4.1.2-1: Preconditions, postconditions, and parameterization of the flow for different NS lifecycle management operations**

Operation	Precondition	Task	RequestStructure	Postcondition
Instantiate NS	NS instance created and in NOT_INSTANTIATED state	instantiate	InstantiateNsRequest	NS instance in INSTANTIATED state
Scale NS	NS instance in INSTANTIATED state	scale	ScaleNsRequest	NS instance still in INSTANTIATED state and NS has been scaled
Update NS	NS instance in INSTANTIATED state	update	UpdateNsRequest	NS instance still in INSTANTIATED state and NS has been updated
Heal NS	NS instance in INSTANTIATED state	heal	HealNsRequest	NS instance still in INSTANTIATED state
Terminate NS	NS instance in INSTANTIATED state	terminate	TerminateNsRequest	NS instance in NOT_INSTANTIATED state

### 6.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [16] are part of the NS lifecycle management interface.

## 6.4.2 Resource: NS Instances

### 6.4.2.1 Description

This resource represents NS instances. The API consumer can use this resource to create "Individual NS instance" resources, and to query NS instances.

### 6.4.2.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/ns\_instances**

This resource shall support the resource URI variables defined in table 6.4.2.2-1.

**Table 6.4.2.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.

### 6.4.2.3 Resource methods

#### 6.4.2.3.1 POST

The POST method creates a new NS instance resource.

This method shall follow the provisions specified in the tables 6.4.2.3.1-1 and 6.4.2.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual NS instance" resource as defined in clause 6.4.3 shall have been created, and the value of the "instantiationState" attribute in the representation of that resource shall be "NOT\_INSTANTIATED". A notification of type NsIdentifierCreationNotification shall be triggered as part of successfully executing this method as defined in clause 6.5.2.6.

**Table 6.4.2.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Remarks
n/a		

**Table 6.4.2.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Remarks	
	CreateNsRequest	1	The NS creation parameters, as defined in clause 6.5.2.9.	
Response body	Data type	Cardinality	Response Codes	Remarks
	NsInstance	1	201 Created	<p>Shall be returned when a new "Individual NS instance" resource and the associated NS instance identifier has been created successfully.</p> <p>The response body shall contain a representation of the created NS instance, as defined in clause 6.5.2.10.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created NS instance.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 6.4.2.3.2 GET

The GET method queries information about multiple NS instances.

This method shall follow the provisions specified in the tables 6.4.2.3.2-1 and 6.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>All attribute names that appear in the NsInstance and in data types referenced from it shall be supported by the NFVO in the filter expression.</p>
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_default	0..1	<p>Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.</p> <p>The following attributes shall be excluded from the NsInstance structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided:</p> <ul style="list-style-type: none"> <li>- vnflInstances</li> <li>- pnfInfo</li> <li>- virtualLinkInfo</li> <li>- vnffgInfo</li> <li>- sapInfo</li> <li>- nsScaleStatus</li> <li>- additionalAffinityOrAntiAffinityRules</li> <li>- wanConnectionInfo</li> </ul>
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 6.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
n/a				
Response body	Data type	Cardinality	Response Codes	Remarks
	NsInstance	0..N	200 OK	<p>Shall be returned when information about zero or more NS instances has been queried successfully.</p> <p>The response body shall contain in an array the representations of zero or more NS instances, as defined in clause 6.5.2.10.</p> <p>If the "filter" URI parameter or one of the "all_fields", "fields" (if supported), "exclude_fields" (if supported) or "exclude_default" URI parameters was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 and 5.3.2 of ETSI GS NFV-SOL 013 [16], respectively.</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute selector.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 6.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].



### 6.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 6.4.3 Resource: Individual NS Instance

### 6.4.3.1 Description

This resource represents an individual NS instance. The API consumer can use this resource to delete the underlying NS instance and to read information about the NS instance.

### 6.4.3.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/ns\_instances/{nsInstanceId}**

The base resource URI variables for this resource are defined in table 6.4.3.2-1.

**Table 6.4.3.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsInstanceId	Identifier of the NS instance

### 6.4.3.3 Resource methods

#### 6.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.3.3.2 GET

The GET method retrieves information about an NS instance by reading an "Individual NS instance" resource.

This method shall follow the provisions specified in the tables 6.4.3.3.2-1 and 6.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 6.4.3.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Remarks
none supported		

**Table 6.4.3.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response codes	Remarks
	NsInstance	1	200 OK	<p>Shall be returned when information about an individual NS instance has been read successfully.</p> <p>The response body shall contain a representation of the NS instance, as defined in clause 6.5.2.10.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

**6.4.3.3.3 PUT**

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

**6.4.3.3.4 PATCH**

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

**6.4.3.3.5 DELETE**

This method deletes an "Individual NS instance" resource.

This method shall follow the provisions specified in the tables 6.4.3.3.5-1 and 6.4.3.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual NS instance" resource shall not exist any longer. A notification of type "NsIdentifierDeletionNotification" shall be triggered as part of successfully executing this method as defined in clause 6.5.2.7.

**Table 6.4.3.3.5-1: URI query parameters supported by the DELETE method on this resource**

Name	Cardinality	Remarks
none supported		

Table 6.4.3.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		204 No Content	<p>Shall be returned when the "Individual NS instance" resource and the associated NS identifier have been deleted successfully.</p> <p>The response body shall be empty.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the NS instance resource is in INSTANTIATED state.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 6.4.4 Resource: Instantiate NS task

### 6.4.4.1 Description

This task resource represents the "Instantiate NS" operation. The API consumer can use this resource to instantiate an NS instance.

### 6.4.4.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/ns\_instances/{nsInstanceId}/instantiate**

This resource shall support the resource URI variables defined in table 6.4.4.2-1.

Table 6.4.4.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsInstanceId	Identifier of the NS instance to be instantiated.

### 6.4.4.3 Resource methods

#### 6.4.4.3.1 POST

The POST method instantiates an NS instance.

In case of the "feasibilityCheckReserve" attribute is present, the method runs the feasibility check procedure for an NS instantiation scheduled operation or runs the feasibility checking procedure followed by the execution of the subsequent (actual) NS instantiation operation.

This method shall follow the provisions specified in the tables 6.4.4.3.1-1 and 6.4.4.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 6.4.1.2.

Before the NFVO initiates the "Instantiate NS" operation it shall verify (if not already done previously when onboarding the NSD) that all descriptors referenced in the selected NS deployment flavour, or a respective overriding descriptor indicated in the operation, are already on-boarded.

In addition, once the NFVO has successfully completed the underlying NS LCM operation occurrence, it shall set the "nsState" attribute to the value "INSTANTIATED" in the representation of the "Individual NS instance" resource.

**Table 6.4.4.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
None supported		

**Table 6.4.4.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	InstantiateNsRequest	1	Parameters for the instantiate NS operation, as defined in clause 6.5.2.11.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the request has been accepted for processing, but the processing has not been completed.</p> <p>The response body shall be empty.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual NS LCM operation occurrence" resource corresponding to the operation.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the NS instance resource is in the INSTANTIATED state, or that another lifecycle management operation is ongoing.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

#### 6.4.4.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.5 Resource: Scale NS task

#### 6.4.5.1 Description

This task resource represents the "Scale NS" operation. The API consumer can use this resource to request scaling an NS instance.

Scaling an NS instance can be performed by explicitly adding/removing existing VNF instances to/from the NS instance, by leveraging on the abstraction mechanism provided by the NS scaling aspects and NS levels information elements declared in the NSD or by scaling individual VNF instances that are part of the NS itself. When adding VNFs and nested NSs - already existing or not - to the NS to be scaled, the NFVO shall follow the indications provided by the dependencies attribute, as specified in the corresponding NSD.

**NOTE:** In case the NS is a composite NS, it is also possible to scale directly its nested NS, as they are also NS and thus indirectly effectively scale the composite NS.

For the case of scaling individual VNF instances that are part of the NS instance, it depends on the VNF capabilities, and is declared in the corresponding VNFD, whether the scaling of the individual VNF instance is supported.

#### 6.4.5.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/ns\_instances/{nsInstanceId}/scale**

This resource shall support the resource URI variables defined in table 6.4.5.2-1.

**Table 6.4.5.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsInstanceId	Identifier of the NS instance to be scaled.

#### 6.4.5.3 Resource methods

##### 6.4.5.3.1 POST

The POST method requests to scale an NS instance resource.

This method shall follow the provisions specified in the tables 6.4.5.3.1-1 and 6.4.5.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 6.4.1.2.

In addition, once the NFVO has successfully completed the underlying NS LCM operation occurrence, it shall reflect the result of scaling the NS instance by updating the "nsScaleStatus" attribute in the representation of the "Individual NS instance" resource.

**Table 6.4.5.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

Table 6.4.5.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	ScaleNsRequest	1	Parameters for the scale NS operation, as defined in clause 6.5.2.14.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the request has been accepted for processing, but the processing has not been completed.</p> <p>The response body shall be empty.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual NS lifecycle operation occurrence" resource corresponding to the operation.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the NS instance resource is in NOT_INSTANTIATED state, or that another lifecycle management operation is ongoing.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

#### 6.4.5.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 6.4.6 Resource: Update NS task

### 6.4.6.1 Description

This task resource represents the "Update NS" operation. The API consumer can use this resource to update an NS instance.

For the cases of "change VNF deployment flavour" and "operate VNF" on individual VNF instances that are part of the NS instance, it depends on the VNF capabilities, and is declared in the VNFD, whether this operation is supported for a particular VNF. This operation may be service-disruptive.

### 6.4.6.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/ns\_instances/{nsInstanceId}/update**

This resource shall support the resource URI variables defined in table 6.4.6.2-1.

**Table 6.4.6.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsInstanceId	Identifier of the NS instance to be updated.

### 6.4.6.3 Resource methods

#### 6.4.6.3.1 POST

The POST method updates an NS instance.

In case of the "feasibilityCheckReserve" attribute is present, the NFVO runs the feasibility check procedure for an NS updating scheduled operation or runs the feasibility checking procedure followed by the execution of the subsequent (actual) NS updating operation.

This method shall follow the provisions specified in the tables 6.4.6.3.1-1 and 6.4.6.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 6.4.1.2.

If the "Update NS" operation involves adding a VNF instance, a nested NS instance or a PNF to the NS instance, changing an NS or VNF deployment flavour, or changing the associated NSD descriptor or a current VNF package, the NFVO shall, before it initiates the operation, verify (if not already done previously when onboarding the NSD) that all descriptors referenced in the selected NS deployment flavour, or a respective overriding descriptor indicated in the operation, are already on-boarded.

In addition, once the NFVO has successfully completed the underlying NS LCM operation occurrence, it shall reflect the result of the update of the NS instance by updating the representation of the "Individual NS instance" resource accordingly to the update.

**Table 6.4.6.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

Table 6.4.6.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	UpdateNsRequest	1	Parameters for the update NS operation, as defined in clause 6.5.2.12.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the request has been accepted for processing, but the processing has not been completed.</p> <p>The response body shall be empty.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual NS lifecycle operation occurrence" resource corresponding to the operation.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the NS instance resource is in NOT_INSTANTIATED state, or that another lifecycle management operation is ongoing.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

#### 6.4.6.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.6.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.6.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.7 Resource: Heal NS task

#### 6.4.7.1 Description

This task resource represents the "Heal NS" operation. The API consumer can use this resource to request healing an NS instance.

For the case of healing individual VNF instances that are part of the NS instance, it depends on VNF capabilities, and is declared in the VNFD, whether the healing of the individual VNF instance is supported.



## 6.4.7.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/ns\_instances/{nsInstanceId}/heal**

This resource shall support the resource URI variables defined in table 6.4.7.2-1.

**Table 6.4.7.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsInstanceId	Identifier of the NS instance to be healed.

## 6.4.7.3 Resource methods

### 6.4.7.3.1 POST

The POST method requests to heal an NS instance.

This method shall follow the provisions specified in the tables 6.4.7.3.1-1 and 6.4.7.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 6.4.1.2.

**Table 6.4.7.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 6.4.7.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	HealNsRequest	1	Parameters for the heal NS operation, as defined in clause 6.5.2.13.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the request has been accepted for processing, but the processing has not been completed.</p> <p>The response body shall be empty.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual NS lifecycle operation occurrence" resource corresponding to the operation.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the NS instance resource is in NOT_INSTANTIATED state, or that another lifecycle management operation is ongoing.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]		4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 6.4.7.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.7.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.7.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.7.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 6.4.8 Resource: Terminate NS task

### 6.4.8.1 Description

This task resource represents the "Terminate NS" operation. The API consumer can use this resource to terminate an NS instance.

### 6.4.8.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/ns\_instances/{nsInstanceId}/terminate**

This resource shall support the resource URI variables defined in table 6.4.8.2-1.

**Table 6.4.8.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsInstanceId	The identifier of the NS instance to be terminated.

### 6.4.8.3 Resource methods

#### 6.4.8.3.1 POST

The POST method terminates an NS instance.

This method shall follow the provisions specified in the tables 6.4.8.3.1-1 and 6.4.8.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 6.4.1.2.

In addition, once the NFVO has successfully completed the underlying NS LCM operation occurrence, it shall set the "nsState" attribute in the representation of the "Individual NS instance" resource to the value "NOT\_INSTANTIATED".

This method can only be used with an NS instance in the INSTANTIATED state. Terminating an NS instance does not delete the NS instance identifier, but rather transitions the NS into the NOT\_INSTANTIATED state.

Table 6.4.8.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Remarks
none supported		

Table 6.4.8.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	TerminateNsRequest	1	The terminate NS request parameters, as defined in clause 6.5.2.15.	
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a	1	202 Accepted	<p>Shall be returned when the request has been accepted for processing.</p> <p>The response body shall be empty.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual NS lifecycle operation occurrence" resource corresponding to the operation.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the NS instance resource is in NOT_INSTANTIATED state, or that another lifecycle management operation is ongoing.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

#### 6.4.8.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.8.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.8.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.8.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 6.4.9 Resource: NS LCM operation occurrences

### 6.4.9.1 Description

This resource represents NS lifecycle management operation occurrences. The API consumer can use this resource to query status information about multiple NS lifecycle management operation occurrences.

### 6.4.9.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/ns\_lcm\_op\_occs**

The base resource URI variables for this resource are defined in table 6.4.9.2-1.

**Table 6.4.9.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.

### 6.4.9.3 Resource methods

#### 6.4.9.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.9.3.2 GET

The API consumer can use this method to query status information about multiple NS lifecycle management operation occurrences.

This method shall follow the provisions specified in the tables 6.4.9.3.2-1 and 6.4.9.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 6.4.9.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].  The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.  All attribute names that appear in the NsLcmOpOcc and in data types referenced from it shall be supported by the NFVO in the filter expression.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.

Name	Cardinality	Description
exclude_default	0..1	<p>Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.</p> <p>The following attributes shall be excluded from the NsLcmOpOcc structure in the response body if this parameter is provided:</p> <ul style="list-style-type: none"> <li>- operationParams</li> <li>- changedVnfInfo</li> <li>- error</li> <li>- resourceChanges</li> <li>- lcmCoordinations</li> <li>- warnings</li> </ul>
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 6.4.9.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	NsLcmOpOcc	0..N	200 OK	<p>Shall be returned when status information for zero or more NS lifecycle management operation occurrences has been queried successfully.</p> <p>The response body shall contain in an array the status information about zero or more NS lifecycle operation occurrences, as defined in clause 6.5.2.3.</p> <p>If the "filter" URI parameter or one of the "all_fields", "fields", "exclude_fields" or "exclude_default" URI parameters was supplied in the request and is supported, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 and 5.3.2 of ETSI GS NFV-SOL 013 [16], respectively.</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute selector.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>

	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.
--	----------------	------------------------	---------	---

#### 6.4.9.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.9.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.9.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.10 Resource: Individual NS LCM operation occurrence

#### 6.4.10.1 Description

This resource represents an NS lifecycle management operation occurrence. The API consumer can use this resource to read status information about an individual NS lifecycle management operation occurrence. Further, the API consumer can use task resources which are children of this resource to request cancellation of an operation in progress, and to request the handling of operation errors via retrying the operation, rolling back the operation, or permanently failing the operation.

The NFVO may remove an "Individual NS LCM operation occurrence" resource sometime after it has reached one of the terminal states (i.e. the "operationState" attribute of its representation is equal to one of the values "COMPLETED", "FAILED" or "ROLLED\_BACK"). The minimum time how long the NFVO waits before deleting such a resource is defined by means outside the scope of the present document.

#### 6.4.10.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/ns\_lcm\_op\_occs/{nsLcmOpOccId}**

The base resource URI variables for this resource are defined in table 6.4.10.2-1.

**Table 6.4.10.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsLcmOpOccId	Identifier of an NS lifecycle management operation occurrence.

#### 6.4.10.3 Resource methods

##### 6.4.10.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.10.3.2 GET

The API consumer can use this method to retrieve status information about an NS lifecycle management operation occurrence by reading an individual "NS LCM operation occurrence" resource.

This method shall follow the provisions specified in the tables 6.4.10.3.2-1 and 6.4.10.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 6.4.10.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 6.4.10.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	NsLcmOpOcc	1	200 OK	<p>Shall be returned when information about an NS LCM operation occurrence has been read successfully.</p> <p>The response body shall contain status information about an NS lifecycle management operation occurrence (see clause 6.5.2.3).</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 6.4.10.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.10.3.4 PATCH

This method modifies an "Individual NS LCM operation occurrence" resource.

The NFVO shall apply the attributes in the "NsLcmInfoModificationRequest" data structure in the payload body to the existing attributes of the "Individual NsLcmOpOcc" resource data structure according to the rules of JSON Merge Patch (see IETF RFC 7396 [11]).

If the NS LCM operation is a scheduled operation, this operation enables to modify the scheduled time whereas it does not allow to change any other operational attributes. The scheduled time is represented by the "startTime" attribute of the "NsLcmOpOcc" data structure as set, upon creating the NS LCM operation occurrence resource, to the "startTime" attribute value of the related "InstantiateNsRequest", or "updateTime" attribute value of the related "UpdateNsRequest", or "scaleTime" attribute value of the related "ScaleNsRequest" or "terminationTime" attribute value of the related "TerminateNsRequest".

If the "Individual NS LCM operation occurrence" resource to be modified by this method represents a scheduled operation occurrence that did not start yet (i.e. for which the NFVO has an active start timer that has not elapsed yet) and the payload body of the PATCH request contains a modification of the scheduled start time of this operation occurrence, the NFVO shall stop the timer, shall replace the value of the "startTime" attribute of the NS LCM operation occurrence with the new value defined by the "scheduledTime" attribute in the payload body of the request and shall set a timer with that new value.

This method shall follow the provisions specified in tables 6.4.10.3.4-1 and 6.4.10.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.10.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.10.3.4-2: Details of the PATCH request/response on this resource

Request body	Data type	Cardinality	Description	
	NsLcmInfoModificationRequest	1	Parameters for the "Individual NS LCM operation occurrence" resource modification information, as defined in clause 6.5.2.20.  The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [11].	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the "Individual NS LCM operation occurrence" resource has been changed successfully.  The response body shall be empty.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the "Individual NS instance" resource.  Typically, this is due to an attempt to modify the scheduled time of an operation that has already started.  The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled.  Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity.  The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
ProblemDetails	See clause 6.4 of [16]		4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 6.4.10.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 6.4.11 Resource: Retry operation task

### 6.4.11.1 Description

This task resource represents the "Retry operation" operation. The API consumer can use this resource to initiate retrying an NS lifecycle management operation that is in a transient failure state. See also clause 6.6.2.3.



### 6.4.11.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/ns\_lcm\_op\_occs/{nsLcmOpOccId}/retry**

This resource shall support the resource URI variables defined in table 6.4.11.2-1.

**Table 6.4.11.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsLcmOpOccId	Identifier of an NS lifecycle management operation occurrence to be retried. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering an NS LCM operation. It can also be retrieved from the "nsLcmOpOccId" attribute in the NsLcmOperationOccurrenceNotification.

### 6.4.11.3 Resource methods

#### 6.4.11.3.1 POST

The POST method initiates retrying an NS lifecycle management operation if that operation has experienced a temporary failure, i.e. the related "NS LCM operation occurrence" is in "FAILED\_TEMP" state.

This method shall follow the provisions specified in the tables 6.4.11.3.1-1 and 6.4.11.3.1-2 for URI query parameters, request and response data structures, and response codes.

In case of success of processing the asynchronous request, the "operationState" attribute in the representation of the parent resource shall be changed to "PROCESSING" and the applicable "start" notification according to clause 6.6.2.2 shall be emitted to indicate that the underlying NS LCM operation occurrence proceeds.

**Table 6.4.11.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 6.4.11.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a		The POST request to this resource has an empty payload body.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing, but processing has not been completed. The response shall have an empty payload body.
	ProblemDetails	0..1	404 Not Found	Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists.  The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.  Specifically in case of this task resource, the response code 404 shall also be returned if the task is not supported for the NS LCM operation occurrence represented by the parent resource, which means that the task resource consequently does not exist.

				In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the NS LCM operation occurrence resource.</p> <p>Typically, this is due to the fact that the NS LCM operation occurrence is not in FAILED_TEMP state, or another error handling action is starting, such as rollback or fail.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 6.4.11.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.11.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.11.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.11.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 6.4.12 Resource: Rollback operation task

### 6.4.12.1 Description

This task resource represents the "Rollback operation" operation. The API consumer can use this resource to initiate rolling back an NS lifecycle management operation. See also clause 6.6.2.3.

### 6.4.12.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/ns\_lcm\_op\_occs/{nsLcmOpOccId}/rollback**

This resource shall support the resource URI variables defined in table 6.4.12.2-1.

**Table 6.4.12.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsLcmOpOcclId	Identifier of an NS lifecycle management operation occurrence to be rolled back. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering an NS LCM operation. It can also be retrieved from the "nsLcmOpOcclId" attribute in the NsLcmOperationOccurrenceNotification.

### 6.4.12.3 Resource methods

#### 6.4.12.3.1 POST

The POST method initiates rolling back an NS lifecycle operation if that operation has experienced a temporary failure, i.e. the related "NS LCM operation occurrence" is in "FAILED\_TEMP" state.

This method shall follow the provisions specified in the tables 6.4.12.3.1-1 and 6.4.12.3.1-2 for URI query parameters, request and response data structures, and response codes.

In case of success of processing the asynchronous request, the "operationState" attribute in the representation of the parent resource shall be changed to "ROLLING\_BACK" and the applicable "start" notification according to clause 6.6.2.2 shall be emitted to indicate that rollback of the underlying NS LCM operation occurrence is attempted.

**Table 6.4.12.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 6.4.12.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a		The POST request to this resource has an empty payload body.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the request has been accepted for processing, but processing has not been completed.</p> <p>The response shall have an empty payload body.</p>
	ProblemDetails	0..1	404 Not Found	<p>Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically, in case of this task resource, the response code 404 shall also be returned if the task is not supported for the NS LCM operation occurrence represented by the parent resource, which means that the task resource consequently does not exist.</p> <p>In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>

	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the NS LCM operation occurrence resource.</p> <p>Typically, this is due to the fact that the NS LCM operation occurrence is not in FAILED_TEMP state, or another error handling action is starting, such as retry or fail.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 6.4.12.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.12.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.12.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.12.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.13 Resource: Continue operation task

#### 6.4.13.1 Description

This task resource represents the "Continue operation" operation. The API consumer can use this resource to initiate continuing an NS lifecycle management operation. See also clause 6.6.2.3.

#### 6.4.13.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/ns\_lcm\_op\_occs/{nsLcmOpOccId}/continue**

This resource shall support the resource URI variables defined in table 6.4.13.2-1.

**Table 6.4.13.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsLcmOpOcclId	Identifier of an NS lifecycle management operation occurrence to be continued. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering an NS LCM operation. It can also be retrieved from the "nsLcmOpOcclId" attribute in the NsLcmOperationOccurrenceNotification.

### 6.4.13.3 Resource methods

#### 6.4.13.3.1 POST

The POST method initiates continuing an NS lifecycle operation if that operation has experienced a temporary failure, i.e. the related "NS LCM operation occurrence" is in "FAILED\_TEMP" state.

This method shall follow the provisions specified in the tables 6.4.13.3.1-1 and 6.4.13.3.1-2 for URI query parameters, request and response data structures, and response codes.

In case of success of processing the asynchronous request, the "operationState" attribute in the representation of the parent resource shall be changed to "PROCESSING" and the applicable "start" notification according to clause 6.6.2.2 shall be emitted to indicate that the underlying NS LCM operation occurrence continues.

**Table 6.4.13.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 6.4.13.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a		The POST request to this resource has an empty payload body.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the request has been accepted for processing, but processing has not been completed.</p> <p>The response shall have an empty payload body.</p>
	ProblemDetails	0..1	404 Not Found	<p>Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically, in case of this task resource, the response code 404 shall also be returned if the task is not supported for the NS LCM operation occurrence represented by the parent resource, which means that the task resource consequently does not exist.</p> <p>In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>

	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the NS LCM operation occurrence resource.</p> <p>Typically, this is due to the fact that the NS LCM operation occurrence is not in FAILED_TEMP state, or another error handling action is starting, such as retry or fail.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 6.4.13.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.13.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.13.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.13.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.14 Resource: Fail operation task

#### 6.4.14.1 Description

This task resource represents the "Fail operation" operation. The API consumer can use this resource to mark an NS lifecycle management operation occurrence as "finally failed", i.e. change the state of the related NS LCM operation occurrence resource to "FAILED", if it is not assumed that a subsequent retry or rollback will succeed. Once the operation is marked as "finally failed", it cannot be retried or rolled back anymore. See also clause 6.6.2.3.

#### 6.4.14.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/ns\_lcm\_op\_occs/{nsLcmOpOccId}/fail**

This resource shall support the resource URI variables defined in table 6.4.14.2-1.

**Table 6.4.14.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsLcmOpOcclId	Identifier of an NS lifecycle management operation occurrence to be marked as "failed". See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering an NS LCM operation. It can also be retrieved from the "nsLcmOpOcclId" attribute in the NsLcmOperationOccurrenceNotification.	

### 6.4.14.3 Resource methods

#### 6.4.14.3.1 POST

The POST method marks an NS lifecycle management operation occurrence as "finally failed" if that operation occurrence is in "FAILED\_TEMP" state.

This method shall follow the provisions specified in the tables 6.4.14.3.1-1 and 6.4.14.3.1-2 for URI query parameters, request and response data structures, and response codes.

In case of success, the "operationState" attribute in the representation of the parent resource shall be changed to "FAILED" and the applicable "result" notification according to clause 6.6.2.2 shall be emitted to indicate that the execution of the underlying NS LCM operation occurrence has finally and unrecoverable failed.

**Table 6.4.14.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
None supported		

Table 6.4.14.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
n/a			The POST request to this resource has an empty payload body.	
Response body	Data type	Cardinality	Response Codes	Description
	NsLcmOpOcc	1	200 OK	<p>Shall be returned when the state of the NS lifecycle management operation occurrence has been changed successfully.</p> <p>The response shall include a representation of the "Individual NS lifecycle management operation occurrence" resource.</p>
	ProblemDetails	0..1	404 Not Found	<p>Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this task resource, the response code 404 shall also be returned if the task is not supported for the NS LCM operation occurrence represented by the parent resource, which means that the task resource consequently does not exist.</p> <p>In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the NS LCM operation occurrence resource.</p> <p>Typically, this is due to the fact that the NS LCM operation occurrence is not in FAILED_TEMP state, or another error handling action is starting, such as retry or rollback.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 6.4.14.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.14.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.14.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].



### 6.4.14.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 6.4.15 Resource: Cancel operation task

### 6.4.15.1 Description

This task resource represents the "Cancel operation" operation. The API consumer can use this resource to cancel an ongoing NS lifecycle management operation. See also clause 6.6.2.3.

### 6.4.15.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/ns\_lcm\_op\_occs/{nsLcmOpOccId}/cancel**

This resource shall support the resource URI variables defined in table 6.4.15.2-1.

**Table 6.4.15.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsLcmOpOccId	Identifier of an NS lifecycle management operation occurrence to be cancelled. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering an NS LCM operation. It can also be retrieved from the "nsLcmOpOccId" attribute in the NsLcmOperationOccurrenceNotification.

### 6.4.15.3 Resource methods

#### 6.4.15.3.1 POST

The POST method initiates cancelling an ongoing NS lifecycle management operation while it is being executed or rolled back, i.e. the related "NS LCM operation occurrence" is either in "PROCESSING" or "ROLLING\_BACK" state.

This method shall follow the provisions specified in the tables 6.4.15.3.1-1 and 6.4.15.3.1-2 for URI query parameters, request and response data structures, and response codes.

Before returning the "202 Accepted" response, the NFVO shall update the "isCancelPending" and "cancelMode" attributes in the representation of the parent resource according to the provisions in clause 6.5.2.3.

In case of success of processing the asynchronous request:

- 1) If the request has been processed in "PROCESSING" or "ROLLING\_BACK" state, the "operationState" attribute in the representation of the parent resource shall be changed to "FAILED\_TEMP".

In both cases, the NFVO shall update the "isCancelPending" and "cancelMode" attributes in the representation of the parent resource according to the provisions in clause 6.5.2.3 to reflect the new status, and the applicable "result" notification according to clause 6.6.2.2 shall be emitted to indicate that the execution of the underlying NS LCM operation occurrence has temporarily failed.

Due to race conditions, the processing of the actual operation that is to be cancelled may eventually still succeed, in which case the "operationState" attribute in the representation of the parent resource shall represent the result of that operation, rather than the result of the cancellation.

Table 6.4.15.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
None supported		

Table 6.4.15.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	CancelMode	1	The POST request to this resource shall include a CancelMode structure in the payload body to choose between "graceful" and "forceful" cancellation.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the request has been accepted for processing, but processing has not been completed.</p> <p>The response shall have an empty entity body.</p>
	ProblemDetails	0..1	404 Not Found	<p>Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically, in case of this task resource, the response code 404 shall also be returned if the task is not supported for the NS LCM operation occurrence represented by the parent resource, which means that the task resource consequently does not exist.</p> <p>In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the NS LCM operation occurrence resource.</p> <p>Typically, this is due to the fact that the operation occurrence is not in STARTING, PROCESSING or ROLLING_BACK state.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

### 6.4.15.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.15.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.15.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.15.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.16 Resource: Subscriptions

#### 6.4.16.1 Description

This resource represents subscriptions. The API consumer can use this resource to subscribe to notifications related to NS lifecycle management, and to query its subscriptions.

#### 6.4.16.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/subscriptions**

This resource shall support the resource URI variables defined in table 6.4.16.2-1.

**Table 6.4.16.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.

#### 6.4.16.3 Resource methods

##### 6.4.16.3.1 POST

The POST method creates a new subscription.

This method shall follow the provisions specified in the tables 6.4.16.3.1-1 and 6.4.16.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual subscription" resource as defined in clause 6.4.17 shall have been created. This method shall not trigger any notification.

Creation of two subscription resources with the same callback URI and the same filter can result in performance degradation and will provide duplicates of notifications to the OSS, and might make sense only in very rare use cases. Consequently, the NFVO may either allow creating a subscription resource if another subscription resource with the same filter and callback URI already exists (in which case it shall return the "201 Created" response code), or may decide to not create a duplicate subscription resource (in which case it shall return a "303 See Other" response code referencing the existing subscription resource with the same filter and callback URI).

**Table 6.4.16.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Remarks
none supported		

Table 6.4.16.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	LccnSubscriptionRequest	1	Details of the subscription to be created, as defined in clause 6.5.2.2.	
Response body	Data type	Cardinality	Response Codes	Remarks
	LccnSubscription	1	201 Created	<p>Shall be returned when the subscription has been created successfully.</p> <p>The response body shall contain a representation of the created "Individual subscription" resource.</p> <p>The HTTP response shall include a "Location:" HTTP header that points to the created "Individual subscription" resource.</p>
	n/a		303 See Other	<p>Shall be returned if a subscription with the same callback URI and the same filter already exists and the policy of the NFVO is to not create redundant subscriptions.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource.</p> <p>The response body shall be empty.</p>
	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 6.4.18.3.2 and the test has failed.</p> <p>In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

### 6.4.16.3.2 GET

The GET method queries the list of active subscriptions of the functional block that invokes the method. It can be used e.g. for resynchronization after error situations.

This method shall follow the provisions specified in the tables 6.4.16.3.2-1 and 6.4.16.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.16.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Remarks
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].  The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.  All attribute names that appear in the LccnSubscription and in data types referenced from it shall be supported by the NFVO in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 6.4.16.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	LccnSubscription	0..N	200 OK	Shall be returned when the list of subscriptions has been queried successfully.  The response body shall contain in an array the representations of all active subscriptions of the functional block that invokes the method, i.e. zero or more representations of lifecycle change notification subscriptions as defined in clause 6.5.2.4.  If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [16].  If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression.  The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big.  If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 6.4.16.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.16.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.16.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.17 Resource: Individual subscription

#### 6.4.17.1 Description

This resource represents an individual subscription. The API consumer can use this resource to read and to terminate a subscription to notifications related to NS lifecycle management.

#### 6.4.17.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/subscriptions/{subscriptionId}**

This resource shall support the resource URI variables defined in table 6.4.17.2-1.

**Table 6.4.17.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
subscriptionId	Identifier of this subscription

#### 6.4.17.3 Resource methods

##### 6.4.17.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

##### 6.4.17.3.2 GET

The GET method retrieves information about a subscription by reading an "Individual subscription" resource.

This method shall follow the provisions specified in the tables 6.4.17.3.2-1 and 6.4.17.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 6.4.17.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Remarks
none supported		

Table 6.4.17.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	LccnSubscription	1	200 OK	<p>Shall be returned when information about an individual subscription has been read successfully.</p> <p>The response body shall contain a representation of the "Individual subscription" resource.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 6.4.17.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.17.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.17.3.5 DELETE

The DELETE method terminates an individual subscription.

This method shall follow the provisions specified in the tables 6.4.17.3.5-1 and 6.4.17.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual subscription" resource shall not exist any longer. This means that no notifications for that subscription shall be sent to the formerly-subscribed API consumer.

NOTE: Due to race conditions, some notifications might still be received by the formerly-subscribed API consumer for a certain time period after the deletion.

Table 6.4.17.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Remarks
none supported		

Table 6.4.17.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		204 No Content	<p>Shall be returned when the "Individual subscription" resource has been deleted successfully.</p> <p>The response body shall be empty.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 6.4.18 Resource: Notification endpoint

### 6.4.18.1 Description

This resource represents a notification endpoint. The API producer can use this resource to send notifications related to NS lifecycle changes to a subscribed API consumer, which has provided the URI of this resource during the subscription process.

### 6.4.18.2 Resource definition

The resource URI is provided by the API consumer when creating the subscription.

This resource shall support the resource URI variables defined in table 6.4.18.2-1.

**Table 6.4.18.2-1: Resource URI variables for this resource**

Name	Definition
none supported	

### 6.4.18.3 Resource methods

#### 6.4.18.3.1 POST

The POST method delivers a notification from the API producer to an API consumer. The API consumer shall have previously created an "Individual subscription" resource with a matching filter.

This method shall follow the provisions specified in the tables 6.4.18.3.1-1 and 6.4.18.3.1-2 for URI query parameters, request and response data structures, and response codes.

**Table 6.4.18.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Remarks
none supported		

Each notification request body shall include exactly one of the alternatives defined in table 6.4.18.3.1-2.

**Table 6.4.18.3.1-2: Details of the POST request/response on this resource**

	Data type	Cardinality	Remarks	
	Request body	NsLcmOperationOccurrenceNotification	1	A notification about lifecycle changes triggered by an NS LCM operation occurrence.
NsIdentifierCreationNotification		1	A notification about the creation of an NS identifier and the related NS instance resource.	
NsIdentifierDeletionNotification		1	A notification about the deletion of an NS identifier and the related NS instance resource.	
NsChangeNotification		1	A notification to inform the receiver of changes on the NS instance caused by the LCM operation occurrence.	
NsLcmCapacityShortageNotification		1	A notification about lifecycle capacity shortage triggered when there is a shortage condition.	
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		204 No Content	Shall be returned when the notification has been delivered successfully.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.



### 6.4.18.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during subscription.

This method shall follow the provisions specified in the tables 6.4.18.3.2-1 and 6.4.18.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 6.4.18.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 6.4.18.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned to indicate that the notification endpoint has been tested successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 6.4.18.3.3 PUT

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.18.3.4 PATCH

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.18.3.5 DELETE

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 6.4.19 Resource: VNF snapshots

### 6.4.19.1 Description

This resource represents VNF snapshots. The API consumer can use this resource to query information of the VNF snapshots.

### 6.4.19.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/vnf\_snapshots**

This resource shall support the resource URI variables defined in table 6.4.19.2-1.

**Table 6.4.19.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.

### 6.4.19.3 Resource methods

#### 6.4.19.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.19.3.2 GET

The GET method queries information about multiple VNF snapshots.

This method shall follow the provisions specified in the tables 6.4.19.3.2-1 and 6.4.19.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 6.4.19.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16]. The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter. All attribute names that appear in the VnfSnapshotInfo and in data types referenced from it shall be supported by the NFVO in the filter expression.
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter. The following attributes shall be excluded from the VnfSnapshot structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: - vnfInstance - vnfSnapshots
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 6.4.19.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	VnfSnapshotInfo	0..N	200 OK	<p>Shall be returned when information about zero or more VNF snapshots was queried successfully.</p> <p>The response body shall contain in an array the representations of zero or more "Individual VNF snapshot" resources, as defined in clause 6.5.2.17.</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute selector.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 6.4.19.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.19.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 6.4.19.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 6.4.20 Resource: Individual VNF snapshot

### 6.4.20.1 Description

This resource represents an individual VNF snapshot. The API consumer can use this resource to read information about the VNF snapshot, and to delete the VNF snapshot.

### 6.4.20.2 Resource definition

The resource URI is:

**{apiRoot}/nslcm/{apiMajorVersion}/vnf\_snapshots/{vnfSnapshotInfoId}**

The base resource URI variables for this resource are defined in table 6.4.20.2-1.

**Table 6.4.20.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1. of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
vnfSnapshotInfoId	Identifier of the "Individual VNF snapshot" resource. See note.
NOTE:	This identifier can be retrieved from the "id" attribute in the payload body of a response to a query to "VNF snapshots" resource.

### 6.4.20.3 Resource methods

#### 6.4.20.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.20.3.2 GET

The GET method retrieves information about a VNF snapshot by reading an "Individual VNF snapshot" resource.

This method shall follow the provisions specified in the tables 6.4.20.3.2-1 and 6.4.20.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 6.4.20.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 6.4.20.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response codes	Description
	VnfSnapshotInfo	1	200 OK	Shall be returned when information about an individual VNF snapshot was read successfully.  The response body shall contain a representation of the "Individual VNF snapshot" resource, as defined in clause 6.5.2.17.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 6.4.20.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.20.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 6.4.20.3.5 DELETE

This method deletes an "Individual VNF snapshot" resource and the associated VNF snapshot information managed by the NFVO and corresponding VNFM, and any resource associated to the VNF snapshot managed by the VIM.

As the result of successfully executing this method, the "Individual VNF snapshot" resource shall not exist any longer. In addition, the NFVO shall delete any references pointing to the "Individual VNF snapshot" resource from the "NsInstance" data structures representing the "Individual NS instance" resources.

For the successful execution of the operation, the NFVO shall request the corresponding VNFM the deletion of the specific VNF snapshot identified by the "vnfSnapshotInfoId" attribute in the URI, and obtain a successful confirmation from the VNFM of the request.

**NOTE:** The deletion of a VNF snapshot that is specifically associated to an NS instance can also be triggered by executing the "Update NS task" with the "updateType" equal to "DELETE\_VNF\_SNAPSHOT\_INFO" as specified in clause 6.5.2.12.

This method shall follow the provisions specified in the tables 6.4.20.3.5-1 and 6.4.20.3.5-2 for URI query parameters, request and response data structures, and response codes.

**Table 6.4.20.3.5-1: URI query parameters supported by the DELETE method on this resource**

Name	Cardinality	Description
none supported		

**Table 6.4.20.3.5-2: Details of the DELETE request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	<p>Shall be returned when the VNF snapshot resource and the associated VNF snapshot were deleted successfully.</p> <p>The response body shall be empty.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact the VNF snapshot is in use by some operation such as reverting a VNF instance to a VNF snapshot or creating a VNF snapshot package.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx		In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 6.5 Data model

### 6.5.1 Introduction

This clause defines the request and response data structures of the NS Lifecycle management interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error and may choose to ignore them.

### 6.5.2 Resource and notification data types

#### 6.5.2.1 Introduction

This clause defines the data structures to be used in resource representations and notifications.

#### 6.5.2.2 Type: LccnSubscriptionRequest

This type represents a subscription request related to notifications about NS lifecycle changes. It shall comply with the provisions defined in table 6.5.2.2-1.

**Table 6.5.2.2-1: Definition of the LccnSubscriptionRequest data type**

Attribute name	Data type	Cardinality	Description
filter	LifecycleChangeNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this subscription, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [16]. This attribute shall only be present if the subscriber requires authorization of notifications.
verbosity	LcmOpOccNotificationVerbosityType	0..1	This attribute signals the requested verbosity of LCM operation occurrence notifications. If it is not present, it shall default to the value "FULL".

#### 6.5.2.3 Type: NsLcmOpOcc

This type represents an NS lifecycle management operation occurrence. It shall comply with the provisions defined in table 6.5.2.3-1.

**Table 6.5.2.3-1: Definition of the NsLcmOpOcc data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this NS lifecycle operation occurrence.
operationState	NsLcmOperationStateType	1	The state of the NS LCM operation.
stateEnteredTime	DateTime	1	Date-time when the current state has been entered.
nsInstanceId	Identifier	1	Identifier of the NS instance to which the operation applies.
lcmOperationType	NsLcmOpType	1	Type of the actual LCM operation represented by this lcm operation occurrence.
startTime	DateTime	1	Date-time of the start of the operation.
isAutomaticInvocation	Boolean	1	Set to true if this NS LCM operation occurrence has been automatically triggered by the NFVO. This occurs in the case of auto-scaling, auto-healing and when a nested NS is modified as a result of an operation on its composite NS.  Set to false otherwise.

Attribute name	Data type	Cardinality	Description
operationParams	Object	0..1	<p>Input parameters of the LCM operation. This attribute shall be formatted according to the request data type of the related LCM operation. In addition, the provisions in clause 6.7 shall apply.</p> <p>The following mapping between lcmOperationType and the data type of this attribute shall apply:</p> <ul style="list-style-type: none"> <li>• INSTANTIATE: InstantiateNsRequest</li> <li>• SCALE: ScaleNsRequest</li> <li>• UPDATE: UpdateNsRequest</li> <li>• HEAL: HealNsRequest</li> <li>• TERMINATE: TerminateNsRequest</li> </ul> <p>This attribute shall be present if this data type is returned in a response to reading an individual resource, and may be present according to the chosen attribute selector parameter if this data type is returned in a response to a query of a container resource.</p>
isCancelPending	Boolean	1	If the LCM operation occurrence is in "PROCESSING" or "ROLLING_BACK" state and the operation is being cancelled, this attribute shall be set to true. Otherwise, it shall be set to false.
cancelMode	CancelModeType	0..1	The mode of an ongoing cancellation. Shall be present when isCancelPending=true, and shall be absent otherwise.
error	ProblemDetails	0..1	If "operationState" is "FAILED_TEMP" or "FAILED" or "operationState" is "PROCESSING" or "ROLLING_BACK" and previous value of "operationState" was "FAILED_TEMP", this attribute shall be present and contain error information, unless it has been requested to be excluded via an attribute selector.
resourceChanges	Structure (inlined)	0..1	This attribute contains information about the cumulative changes to virtualised resources that were performed so far by the LCM operation since its start, if applicable.
>affectedVnfs	AffectedVnf	0..N	Information about the VNF instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note 1.
>affectedPnfs	AffectedPnf	0..N	Information about the PNF instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note 1.
>affectedVls	AffectedVirtualLink	0..N	Information about the VL instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note 1.
>affectedVnffgs	AffectedVnffg	0..N	Information about the VNFFG instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note 1.
>affectedNss	AffectedNs	0..N	Information about the nested NS instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note 1.
>affectedSaps	AffectedSap	0..N	Information about the SAP instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note 1.
lcmCoordinations	Structure (inlined)	0..N	Information about LCM coordination actions (see clause 12) related to this LCM operation occurrence.

Attribute name	Data type	Cardinality	Description
>id	Identifier	1	Identifier of this coordination action. For a terminated coordination action, this attribute refers to the "id" attribute in the LcmCoord data structure (see clause 12.5.2.3). For a timed-out or ongoing coordination action, this attribute refers to the {coordinationId} URI variable in the "Location" header of the "202 Accepted" HTTP response to the POST request that has initiated the coordination action (see clause 12.4.2.3.1).
>coordinationActionName	Identifier	1	Indicator of the actual coordination action.
>coordinationResult	LcmCoordResultType	0..1	The result of executing the coordination action which also implies the action to be performed by the NFVO as the result of this coordination.  Shall be present if the coordination has been finished. Shall be absent if the coordination is ongoing or has timed out (see note 2).
>startTime	DateTime	1	The time when the NFVO has received the confirmation that the coordination action has been started.
>endTime	DateTime	0..1	The time when the NFVO has received the confirmation that the coordination action has finished or has been cancelled, or the time when a coordination action has timed out.  Shall be present for a coordination action that has finished or timed out (see note 2) and shall be absent if the coordination is ongoing.
>endpointType	Enum (inlined)	1	The endpoint type used by this coordination action.  Valid values: - MGMT: coordination with other operation supporting management systems (e.g. OSS/BSS)
>delay	DateTime	0..1	The end of the delay period. This attribute shall be present if the last known HTTP response related to this coordination has contained a "Retry-After" header, and shall be absent otherwise.
rejectedLcmCoordinations	Structure (inlined)	0..N	Information about LCM coordination actions (see clause 12) that were rejected by 503 error which means they can be tried again after a delay. See note 3.
>coordinationActionName	Identifier	1	Indicator of the actual coordination action.
>rejectionTime	DateTime	1	The time when the NFVO has received the 503 response that rejects the actual coordination.
>endpointType	Enum (inlined)	1	The endpoint type used by this coordination action. Valid values: - MGMT: coordination with other operation supporting management systems (e.g. OSS/BSS)
>delay	DateTime	1	The end of the delay period, as calculated from the startTime and "Retry-After" header.
warnings	String	0..N	Warning messages that were generated while the operation was executing.  If the operation has included VNF LCM operations or NS LCM coordination actions and these have resulted in warnings, such warnings should be added to this attribute.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>nsInstance	Link	1	Link to the NS instance that the operation applies to.



Attribute name	Data type	Cardinality	Description
>cancel	Link	0..1	Link to the task resource that represents the "cancel" operation for this LCM operation occurrence, if cancelling is currently allowed.
>retry	Link	0..1	Link to the task resource that represents the "retry" operation for this LCM operation occurrence, if retrying is currently allowed.
>rollback	Link	0..1	Link to the task resource that represents the "rollback" operation for this LCM operation occurrence, if rolling back is currently allowed.
>continue	Link	0..1	Link to the task resource that represents the "continue" operation for this LCM operation occurrence, if continuing is currently allowed.
>fail	Link	0..1	Link to the task resource that represents the "fail" operation for this LCM operation occurrence, if declaring as failed is currently allowed.
NOTE 1: This allows the OSS/BSS to obtain a copy of the latest "result" notification if it has not received it due to an error. If the notification represents the successful result of a lifecycle operation, at least an affectedVnf, or affectedPnf, or affectedVI, or affectedVnfg or affectedNs, or affectedSap shall be present.			
NOTE 2: A coordination action has timed out if the NFVO has not been able to read the "Individual coordination action" resource within a timeout interval after requesting the coordination to be started or to be cancelled. The length of the timeout interval is defined by means outside the scope of the present document.			
NOTE 3: The list of rejected coordinations may be garbage collected if the LCM operation occurrence has reached a terminal state, i.e. one of "COMPLETED", "FAILED", "PARTIALLY COMPLETED" and "ROLLED_BACK".			

#### 6.5.2.4 Type: LccnSubscription

This type represents a subscription related to notifications about NS lifecycle changes. It shall comply with the provisions defined in table 6.5.2.4-1.

**Table 6.5.2.4-1: Definition of the LccnSubscription data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this subscription resource.
filter	LifecycleChangeNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
verbosity	LcmOpOccNotificationVerbosityType	1	This attribute signals the verbosity of LCM operation occurrence notifications.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.

#### 6.5.2.5 Type: NsLcmOperationOccurrenceNotification

This type represents an NS lifecycle management operation occurrence notification, which informs the receiver of changes in the NS lifecycle caused by an NS LCM operation occurrence. The NS LCM operation occurrence may be triggered by the OSS/BSS or automatically triggered by the NFVO. The automatic trigger occurs in case of auto-scaling, auto-healing and when a nested NS is modified as a result of an operation on its composite NS.

It shall comply with the provisions defined in table 6.5.2.5-1. The support of the notification is mandatory.

This notification shall be triggered by the NFVO when there is a change in the NS lifecycle caused by an NS LCM operation occurrence that changes the NS lifecycle, including:

- Instantiation of the NS (start and result, including feasibility check)
- Scaling of the NS (start and result, including the auto-scaling)
- Update of the NS (start and result, including feasibility check)

- Termination of the NS (start and result)
- Healing of the NS (start and result, including the auto-healing)
- When a nested NS is modified as a result of an operation on its composite NS
- Modifying the "scheduledTime" attribute of a scheduled NS LCM operation (Instantiate, Update, Scale and Terminate)

Clause 6.6.2 defines the states and state transition of an NS LCM operation occurrence, and also specifies details of the notifications to be emitted at each state transition.

If this is the initial notification about the start of an NS LCM operation occurrence, it is assumed that the notification is sent by the NFVO before any action is taken as part of the LCM operation. Due to possible race conditions, the "start" notification and the LCM operation acknowledgment (i.e. the "202 Accepted" response) can arrive in any order at the OSS/BSS, and the OSS/BSS shall be able to handle such a situation.

If this is a notification about a final or intermediate result state of an NS LCM operation occurrence, the notification shall be sent after all related actions of the LCM operation that led to this state have been executed.

The new state shall be set in the "Individual NS LCM operation occurrence" resource before the notification about the state change is sent.

The amount of information provided in the LCM operation occurrence notifications to be issued by the NFVO when a particular subscription matches can be controlled by the API consumer using the "verbosity" attribute in the subscription request (see clause 6.5.2.2). The "verbosity" setting in a particular individual subscription shall only apply to the LCM operation occurrence notifications triggered by that subscription. However, it shall not affect the amount of information in the "NsLcmOpOcc" structure (see clause 6.5.2.3) which represents the "Individual LCM operation occurrence" resource associated with each of the notifications.

If this is a notification about the result of successful feasibility check of an NS LCM operation, the notification shall include the "feasibilityCheckResult" attribute.

In case this is a notification about the result of failure(s) observed during the feasibility checking procedure of an NS LCM operation, the notification shall include appropriate information about the failures experienced in performing the NS LCM operation as part of the feasibility check.

See clause 6.6.2.2 for further provisions regarding sending this notification, including in cases of handling LCM operation errors.

**Table 6.5.2.5-1: Definition of the NsLcmOperationOccurrenceNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
nsInstanceld	Identifier	1	The identifier of the NS instance affected.
nsLcmOpOcclld	Identifier	1	The identifier of the NS lifecycle operation occurrence associated to the notification.
operation	NsLcmOpType	1	The lifecycle operation.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsLcmOperationOccurrenceNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timestamp	DateTime	1	Date-time of the generation of the notification.

Attribute name	Data type	Cardinality	Description
notificationStatus	Enum (inlined)	1	Indicates whether this notification reports about the start of an NS lifecycle operation or the result of an NS lifecycle operation.  Permitted values: - START: Informs about the start of the NS LCM operation occurrence. - RESULT: Informs about the final or intermediate result of the NS LCM operation occurrence.
operationState	NsLcmOperationStateType	1	The state of the NS lifecycle operation occurrence.
isAutomaticInvocation	Boolean	1	Set to true if this NS LCM operation occurrence has been automatically triggered by the NFVO. This occurs in case of auto-scaling, auto-healing and when a nested NS is modified as a result of an operation on its composite NS.  Set to false otherwise.
verbosity	LcmOpOccNotificationVerbosityType	0..1	This attribute signals the verbosity of the notification. If it is not present, it shall default to the value "FULL".  If the value is "SHORT", full change details can be obtained by performing a GET request on the "Individual LCM operation occurrence" resource that is signalled by the "nsLcmOpOcc" child attribute of the "_links" attribute.
affectedVnf	AffectedVnf	0..N	Information about the VNF instances that were affected during the lifecycle operation. See note 1.
affectedPnf	AffectedPnf	0..N	Information about the PNF instances that were affected during the lifecycle operation. See note 1.
affectedVl	AffectedVirtualLink	0..N	Information about the VL instances that were affected during the lifecycle operation. See note 1.
affectedVnffg	AffectedVnffg	0..N	Information about the VNFFG instances that were affected during the lifecycle operation. See note 1.
affectedNs	AffectedNs	0..N	Information about the NS instances that were affected during the lifecycle operation. See note 1.
affectedSap	AffectedSap	0..N	Information about the SAP instances that were affected during the lifecycle operation. See note 1.
feasibilityCheckResult	FeasibilityCheckResultType	0..1	Information about the feasibility check result. The attribute shall be provided if the notification is sent to indicate the result of feasibility check. See note 2.
error	ProblemDetails	0..1	Details of the latest error, if one has occurred during executing the LCM operation (see clause 6.3 of ETSI GS NFV-SOL 013 [16]). Shall be present if "operationState" is "FAILED_TEMP" or "FAILED", and shall be absent otherwise. See note 2.
_links	LccnLinks	1	Links to resources related to this notification.
NOTE 1: Shall be present if the "notificationStatus" is set to "RESULT", the "verbosity" attribute is set to "FULL" and the operation has performed any resource modification. Shall be absent otherwise. This attribute contains information about the cumulative changes to virtualised resources that were performed so far by the NS LCM operation occurrence and by any of the error handling procedures for that operation occurrence. NOTE 2: In case of failure in feasibility checking, additional information on the failure shall be reported in the "error" attribute of the NsLcmOperationOccurrenceNotification structure along with the result.			

### 6.5.2.6 Type: NsIdentifierCreationNotification

This type represents an NS identifier creation notification, which informs the receiver of the creation of a new "Individual NS instance" resource and the associated NS instance identifier. It shall comply with the provisions defined in table 6.5.2.6-1. The support of the notification is mandatory.

This notification shall be triggered by the NFVO when it has created an "Individual NS instance" resource and the associated NS instance identifier.

**Table 6.5.2.6-1: Definition of the NsIdentifierCreationNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsIdentifierCreationNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timestamp	DateTime	1	Date-time of the generation of the notification.
nsInstanceid	Identifier	1	The created NS instance identifier. Shall be set to the same "id" attribute value of the associated "NsInstance" representation of the "Individual NS instance" resource.
links	LccnLinks	1	Links to resources related to this notification.

### 6.5.2.7 Type: NsIdentifierDeletionNotification

This type represents an NS identifier deletion notification, which informs the receiver of the deletion of a new "Individual NS instance" resource and the associated NS instance identifier. It shall comply with the provisions defined in table 6.5.2.7-1. The support of the notification is mandatory.

This notification shall be triggered by the NFVO when it has deleted an "Individual NS instance" resource and the associated NS instance identifier.

**Table 6.5.2.7-1: Definition of the NsInstanceDeletionNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsIdentifierDeletionNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
nsInstanceid	Identifier	1	The deleted NS instance identifier. Shall be set to the same "id" attribute value of the associated "NsInstance" representation of the "Individual NS instance" resource.
links	LccnLinks	1	Links to resources related to this notification.

### 6.5.2.8 Type: NsChangeNotification

This type represents an NS change notification, which informs the receiver of changes on an NS instance caused by an LCM operation occurrence, which directly or indirectly impacts its NS component and is triggered without any context of this NS instance. In other words, this notification is triggered by an LCM operation occurrence on one of the components of the NS instance where aforementioned LCM operation occurrence is not associated to an NS LCM operation occurrence on the NS instance itself. Examples for such operations are a VNF LCM operation on a constituent VNF instance e.g. requested by an EM or automatically triggered by a VNFM, or an NS LCM operation on a constituent NS instance executed by either the same NFVO or another NFVO than the one managing the current NS instance.

This notification is different from the NsLcmOperationOccurrenceNotification (see clause 6.5.2.5), which is triggered by the NS LCM operation occurrence on the current NS instance itself. It shall comply with the provisions defined in table 6.5.2.8-1. The support of the notification is mandatory.

The trigger conditions include:

- 1) NS LCM operation occurrence which directly or indirectly impacts the NS component (start and result).

If this is a notification about the start of an LCM operation occurrence impacting the NS component, the notification shall be sent as soon as the impact on the NS component is identified.

If this is a notification about a final result state of an NS LCM operation occurrence impacting the NS component, the notification shall be sent after the impact on the NS component has been executed.

**Table 6.5.2.8-1: Definition of the NsChangeNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
nsInstancelid	Identifier	1	The identifier of the NS instance affected.
nsComponentType	NsComponentType	1	Indicates the impacted NS component type.
nsComponentId	Identifier	1	The identifier of the impacted NS component.
lcmOpOccIdImpactingNsComponent	Identifier	1	The identifier of the lifecycle operation occurrence which is associated to the notification and impacts the NS component directly or indirectly.
lcmOpNameImpactingNsComponent	LcmOpNameForChangeNotificationType	1	Indicates the name of the lifecycle operation occurrence which is associated to the notification and impacts the NS component directly or indirectly.
lcmOpOccStatusImpactingNsComponent	LcmOpOccStatusForChangeNotificationType	1	Indicates this status of the lifecycle operation occurrence which is associated to the notification and impacts the NS component directly or indirectly.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsChangeNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
error	ProblemDetails	0..1	Details of the latest error, if one has occurred during executing the LCM operation (see clause 6.3 of ETSI GS NFV-SOL 013 [16]). Shall be present if lcmOpOccStatusImpactingNsComponent is "PARTIALLY_COMPLETED" or "FAILED", and shall be absent otherwise.
_links	LccnLinks	1	Links to resources related to this notification.

### 6.5.2.9 Type: CreateNsRequest

This type represents request parameters for the "Create NS identifier" operation. It shall comply with the provisions defined in table 6.5.2.9-1.

Table 6.5.2.9-1: Definition of the CreateNsRequest data type

Attribute name	Data type	Cardinality	Description
nsdId	Identifier	1	Identifier of the NSD that defines the NS instance to be created.
nsName	String	1	Human-readable name of the NS instance to be created.
nsDescription	String	1	Human-readable description of the NS instance to be created.

### 6.5.2.10 Type: NsInstance

This type represents an NS instance. It shall comply with the provisions defined in table 6.5.2.10-1.

Table 6.5.2.10-1: Definition of the NsInstance data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the NS instance.
nsInstanceName	String	1	Human readable name of the NS instance.
nsInstanceDescription	String	1	Human readable description of the NS instance.
nsdId	Identifier	1	Identifier of the NSD on which the NS instance is based. The value is copied from the NSD.
versionDependency	VersionDependency	0..N	Describes version dependencies currently valid for the nested NS instance. Identifies versions of descriptors of other constituents in the NSD upon which the nested NS depends. The dependencies may be described for the NSD referenced in this NsInfo with nsdId and for NSDs with the same NsdExtInvariantId.  There shall not be more than one versionDependency present with the same dependentConstituentId.  It may be present for the NsInfo of a nested NS. It shall not be present otherwise.
nsdInfoId	Identifier	1	Identifier of the NSD information object on which the NS instance is based. This identifier has been allocated by the NFVO.
flavourId	IdentifierInNsd	0..1	Identifier of the NS deployment flavour applied to the NS instance. This attribute shall be present if the nsState attribute value is INSTANTIATED.
priority	Number	0..1	Specifies the priority level of the NS instance.  See note 2 and note 3.
vnfInstance	VnfInstance	0..N	Information on constituent VNF(s) of the NS instance.
pnfInfo	PnfInfo	0..N	Information on the PNF(s) that are part of the NS instance.
virtualLinkInfo	NsVirtualLinkInfo	0..N	Information on the VL(s) of the NS instance. This attribute shall be present if the nsState attribute value is INSTANTIATED and if the NS instance has specified connectivity.
vnffgInfo	VnffgInfo	0..N	Information on the VNFFG(s) of the NS instance.
sapInfo	SapInfo	0..N	Information on the SAP(s) of the NS instance.
nestedNsInstanceId	Identifier	0..N	Identifier of the nested NS(s) of the NS instance.

Attribute name	Data type	Cardinality	Description
vnfSnapshotInfolds	Identifier	0..N	Identifier of information on VNF snapshots of VNF instances that are part of this NS instance.
nsState	Enum (inlined)	1	The state of the NS instance.  Permitted values: <ul style="list-style-type: none"> <li>- NOT_INSTANTIATED: The NS instance is terminated or not instantiated.</li> <li>- INSTANTIATED: The NS instance is instantiated.</li> </ul>
monitoringParameter	NsMonitoringParameter	0..N	Performance metrics tracked by the NFVO (e.g. for auto-scaling purposes) as identified by the NS designer in the NSD.
nsScaleStatus	NsScaleInfo	0..N	Status of each NS scaling aspect declared in the applicable DF, how "big" the NS instance has been scaled w.r.t. that aspect. This attribute shall be present if the nsState attribute value is INSTANTIATED.
additionalAffinityOrAntiAffinityRule	AffinityOrAntiAffinityRule	0..N	Information on the additional affinity or anti-affinity rule from NS instantiation operation. Shall not conflict with rules already specified in the NSD.
wanConnectionInfo	WanConnectionInfo	0..N	Information about WAN related connectivity enabling multi-site VLs.
dataFlowMirroringInfo	MirroringInfo	0..N	Information related to Data Flow Mirroring associated to this NS instance.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>nestedNsInstances	Link	0..N	Links to the nested NS instances of the present NS instance.
>vnfSnapshotInfos	Link	0..N	Links to the VNF snapshots associated to VNF instances which are part of this NS instance.
>instantiate	Link	0..1	Link to the "instantiate" task resource, if the related operation is possible based on the current status of this NS instance resource (i.e. NS instance in NOT_INSTANTIATED state).
>terminate	Link	0..1	Link to the "terminate" task resource, if the related operation is possible based on the current status of this NS instance resource (i.e. NS instance is in INSTANTIATED state).
>update	Link	0..1	Link to the "update" task resource, if the related operation is possible based on the current status of this NS instance resource (i.e. NS instance is in INSTANTIATED state).
>scale	Link	0..1	Link to the "scale" task resource, if the related operation is supported for this NS instance, and is possible based on the current status of this NS instance resource (i.e. NS instance is in INSTANTIATED state).
>heal	Link	0..1	Link to the "heal" task resource, if the related operation is supported for this NS instance, and is possible based on the current status of this NS instance resource (i.e. NS instance is in INSTANTIATED state).
NOTE 1: Void.			
NOTE 2: The "priority" attribute of the NS instance is configured in the NSD in the NsDf structure. The mapping from application-specific priority values to a value in the NsDf is under OSS/BSS responsibility. The "zero" value expresses the highest priority and the fact that the NS instance based on this DF cannot be pre-empted during resource allocation.			
NOTE 3: Examples for the usage of priority include conflict resolution in case of resource shortage.			

## 6.5.2.11 Type: InstantiateNsRequest

This type represents request parameters for the "Instantiate NS" operation. It shall comply with the provisions defined in table 6.5.2.11-1.

Table 6.5.2.11-1: Definition of the InstantiateNsRequest data type

Attribute name	Data type	Cardinality	Description
nsFlavourId	IdentifierInNsd	1	Identifier of the NS deployment flavour to be instantiated.
sapData	SapData	0..N	Create data concerning the SAPs of this NS.
addpnfData	AddPnfData	0..N	Information on the PNF(s) that are part of this NS.
vnfInstanceData	VnfInstanceData	0..N	Specify an existing VNF instance to be used in the NS. If needed, the VNF Profile to be used for this VNF instance is also provided. See note 1.
nestedNsInstanceData	NestedNsInstanceData	0..N	Specify an existing NS instance to be used as a nested NS within the NS. If needed, the NS Profile to be used for this nested NS instance is also provided. See note 2 and note 3.
locationConstraints	VnfLocationConstraint	0..N	Defines the location constraints for the VNF to be instantiated as part of the NS instantiation. An example can be a constraint for the VNF to be in a specific geographic location.
nestedNsLocationConstraints	NestedNsLocationConstraint	0..N	Defines the location constraints for the nested NS to be instantiated as part of the NS instantiation.  An example can be a constraint for the nested NS to be in a specific geographic location.
additionalParamsForNs	KeyValuePairs	0..1	Allows the OSS/BSS to provide additional parameter(s) at the composite NS level (as opposed to the VNF level, which is covered in additionalParamsForVnf), and as opposed to the nested NS level, which is covered in additionalParamsForNestedNs.
additionalParamsForNestedNs	ParamsForNestedNs	0..N	Allows the OSS/BSS to provide additional parameter(s) per nested NS instance (as opposed to the composite NS level, which is covered in additionalParamsForNs, and as opposed to the VNF level, which is covered in additionalParamsForVnf). This is for nested NS instances that are to be created by the NFVO as part of the NS instantiation and not for existing nested NS instances that are referenced for reuse.
additionalParamsForVnf	ParamsForVnf	0..N	Allows the OSS/BSS to provide additional parameter(s) per VNF instance (as opposed to the composite NS level, which is covered in additionalParamsForNs), and as opposed to the nested NS level, which is covered in additionalParamsForNestedNs). This is for VNFs that are to be created by the NFVO as part of the NS instantiation and not for existing VNF that are referenced for reuse.
startTime	DateTime	0..1	Timestamp indicating the earliest time to instantiate the NS. Cardinality "0" indicates the NS instantiation takes place immediately.  If the "Instantiate NS" is a scheduled operation, it shall be possible to modify the schedule time, whereas it shall not be possible to change any other operational parameters.  This attribute can be modified with the PATCH method. See note 7.
nsInstantiationLevelId	IdentifierInNsd	0..1	Identifies one of the NS instantiation levels declared in the DF applicable to this NS instance. See note 5.
targetNsScaleLevelInfo	NsScaleInfo	0..N	This attribute is applicable for NS target scale level instantiation.  This attribute defines the target NS scale level to which the NS instance is to be instantiated for each NS scaling aspect of the DF applicable to this NS instance. See notes 5 and 6.



Attribute name	Data type	Cardinality	Description
wanConnectionData	WanConnectionData	0..N	Information for connecting VNs to the WAN when VNs are deployed across a WAN. See note 4.
additionalAffinityOrAntiAffinityRule	AffinityOrAntiAffinityRule	0..N	Specifies additional affinity or anti-affinity constraint for the VNF instances to be instantiated as part of the NS instantiation. Shall not conflict with rules already specified in the NSD.
feasibilityCheckReserve	FeasibilityCheckReserveType	0..1	Specifies the feasibility check and reserve option for the NS instantiation operation.  Default option is NO_FEASIBILITY_CHECK.  If not present, default value applies. See notes 7, 8, 9, 10 and 11.
dataFlowMirroringData	DataFlowMirroringData	0..N	Information for the data flow mirroring to be associated to this NS instance.
<p>NOTE 1: The DF of the VNF instance shall match the VNF DF present in the associated VNF Profile. The VNFD of the VNF instance shall match the one in the associated VNF Profile if no overridingVnfDId is provided in the interface as part of vnfInstanceData, or the overridingVnfDId provided in the interface if it is provided.</p> <p>NOTE 2: The NS DF of each nested NS shall be one of the allowed flavours in the associated NSD (as referenced in the nestedNsd attribute of the NSD of the NS to be instantiated).</p> <p>NOTE 3: The NSD of each referenced NSs (i.e. each nestedInstanceDId) shall match the one of the nested NSD in the composite NSD if no overridingNsDId is provided in the interface as part of NestedNsInstanceData, or the overridingNsDId provided in the interface if it is provided.</p> <p>NOTE 4: When the NS is deployed over several sites, the VNs of this NS will include VNs in each site connected over the WAN. In this case, the "wanConnectionData" provides the needed information required to connect each VN to the WAN. Annex E provides additional information and guidelines about the usage of the "wanConnectionData" attribute.</p> <p>NOTE 5: The target size for NS instantiation may be specified in either nsInstantiationLevelId or targetNsScaleLevelInfo, but not both. If none of the two parameters (nsInstantiationLevelId or targetNsScaleLevelInfo) are present, the default NS instantiation level as declared in the NSD shall be used.</p> <p>NOTE 6: If targetNsScaleLevelInfo is specified, information provided in targetNsScaleLevelInfo shall be used to calculate the number of instances of the VNFs/nested NSs referred in the NSD. For VNFs or nested NSs that are not scalable based on targetNsScaleLevelInfo, the default NS instantiation level as declared in the NSD shall be used to calculate the number of instances those VNFs/nested NSs.</p> <p>NOTE 7: When "feasibilityCheckReserve" is set to FEASIBILITY_CHECK_ONLY, the "startTime" attribute shall be ignored.</p> <p>NOTE 8: When feasibilityCheckReserve is set to NO_FEASIBILITY_CHECK, the feasibility checking procedure is not performed. Resources might be reserved as triggered by means outside the present operation request (e.g. configuration or policies), e.g. if it is a scheduled NS instantiation operation. See clause A.2.3 in ETSI GS NFV-IFA 010 [2].</p> <p>NOTE 9: When feasibilityCheckReserve is set to FEASIBILITY_CHECK_ONLY, only feasibility checking procedure is performed for the operation, no resources are reserved.</p> <p>NOTE 10: When feasibilityCheckReserve is set to FEASIBILITY_CHECK_WITH_OPERATION, the feasibility checking procedure is performed and based on the procedure result, the actual NS instantiation operation is performed at the time indicated by the "startTime" attribute, if provided. Resources might be reserved as triggered by means outside the present operation request (e.g. configuration or policies), e.g. if it is a scheduled NS instantiation operation. See clause A.2.3 in ETSI GS NFV-IFA 010 [2].</p> <p>NOTE 11: When feasibilityCheckReserve is set to FEASIBILITY_CHECK_WITH_RESERVATION_AND_OPERATION, the feasibility checking is performed and based on the procedure result, resources are reserved as part of the feasibility check, and the actual NS instantiation operation is performed using the resources reserved during the feasibility checking procedure execution, and at the time indicated by the "startTime" attribute, if provided.</p>			

### 6.5.2.12 Type: UpdateNsRequest

This type represents request parameters for the "Update NS" operation. It shall comply with the provisions defined in table 6.5.2.12-1.

Table 6.5.2.12-1: Definition of the UpdateNsRequest data type

Attribute name	Data type	Cardinality	Description
updateType	Enum (inlined)	1	<p>The type of update. It determines also which one of the following parameters is present in the operation.</p> <p>Possible values include:</p> <ul style="list-style-type: none"> <li>- ADD_VNF: Adding existing VNF instance(s)</li> <li>- REMOVE_VNF: Removing VNF instance(s)</li> <li>- INSTANTIATE_VNF: Instantiating new VNF(s)</li> <li>- CHANGE_VNF_DF: Changing VNF DF</li> <li>- OPERATE_VNF: Changing VNF state</li> <li>- MODIFY_VNF_INFORMATION: Modifying VNF information and/or the configurable properties of VNF instance(s)</li> <li>- CHANGE_EXTERNAL_VNF_CONNECTIVITY: Changing the external connectivity of VNF instance(s)</li> <li>- CHANGE_VNFPKG: Changing the VNF package(s) on which (a) VNF instance(s) is/are based (see note 5)</li> <li>- ADD_SAP: Adding SAP(s)</li> <li>- REMOVE_SAP: Removing SAP(s)</li> <li>- ADD_NESTED_NS: Adding existing NS instance(s) as nested NS(s)</li> <li>- REMOVE_NESTED_NS: Removing existing nested NS instance(s)</li> <li>- ASSOC_NEW_NSD_VERSION: Associating a new NSD version to the NS instance</li> <li>- MOVE_VNF: Moving VNF instance(s) from one origin NS instance to another target NS instance</li> <li>- ADD_VNFFG: Adding VNFFG(s)</li> <li>- REMOVE_VNFFG: Removing VNFFG(s)</li> <li>- UPDATE_VNFFG: Updating VNFFG(s)</li> <li>- CHANGE_NS_DF: Changing NS DF</li> <li>- ADD_PNF: Adding PNF</li> <li>- MODIFY_PNF: Modifying PNF</li> <li>- REMOVE_PNF: Removing PNF</li> <li>- CREATE_VNF_SNAPSHOT: Creating VNF Snapshots of VNF instances belonging to the NS instance. See note 2</li> <li>- REVERT_VNF_TO_SNAPSHOT: Reverting a VNF instance belonging to the NS instance to a VNF Snapshot. See note 2 and note 3</li> <li>- DELETE_VNF_SNAPSHOT_INFO: Deleting available VNF Snapshot information for a VNF instance belonging to the NS instance. See note 2</li> <li>- MODIFY_WAN_CONNECTION_INFO: Modify WAN related connectivity information</li> <li>- CREATE_NS_VIRTUAL_LINK: Create an NsVirtualLink instance</li> <li>- DELETE_NS_VIRTUAL_LINK: Delete an NsVirtualLink instance</li> <li>- CREATE_DATA_FLOW_MIRRORING: Create a new data flow mirroring</li> <li>- DELETE_DATA_FLOW_MIRRORING: Delete an existing data flow mirroring</li> <li>- UPDATE_DATA_FLOW_MIRRORING: Update an existing data flow mirroring</li> </ul>
addVnfInstance	VnfInstanceData	0..N	Identifies an existing VNF instance to be added to the NS instance. It shall be present only if updateType = "ADD_VNF".
removeVnfInstanceId	Identifier	0..N	Identifies an existing VNF instance to be removed from the NS instance. It contains the identifier(s) of the VNF instances to be removed. It shall be present only if updateType = "REMOVE_VNF". See note 1.

Attribute name	Data type	Cardinality	Description
instantiateVnfData	InstantiateVnfData	0..N	Identifies the new VNF to be instantiated. It can be used e.g. for the bottom-up NS creation. It shall be present only if updateType = "INSTANTIATE_VNF".
terminateVnfData	TerminateVnfData	0..N	Specifies the details to terminate VNF instance(s). It shall be present only if updateType = "REMOVE_VNF" and if the VNF instance(s) is(are) to be terminated as part of this operation. See notes 1 and 4.
changeVnfFlavourData	ChangeVnfFlavourData	0..N	Identifies the new DF of the VNF instance to be changed to. It shall be present only if updateType = "CHANGE_VNF_DF".
operateVnfData	OperateVnfData	0..N	Identifies the state of the VNF instance to be changed. It shall be present only if updateType = "OPERATE_VNF".
modifyVnfInfoData	ModifyVnfInfoData	0..N	Identifies the VNF information parameters and/or the configurable properties of VNF instance to be modified. It shall be present only if updateType = "MODIFY_VNF_INFORMATION".
changeExtVnfConnectivityData	ChangeExtVnfConnectivityData	0..N	Specifies the new external connectivity data of the VNF instance to be changed. It shall be present only if updateType = "CHANGE_EXTERNAL_VNF_CONNECTIVITY".
changeVnfPackageData	ChangeVnfPackageData	0..N	Specifies the details to change the VNF package on which the VNF instance is based. It shall be present only if updateType = "CHANGE_VNFPKG".
addSap	SapData	0..N	Identifies a new SAP to be added to the NS instance. It shall be present only if updateType = "ADD_SAP."
removeSapId	Identifier	0..N	The identifier an existing SAP to be removed from the NS instance. It shall be present only if updateType = "REMOVE_SAP."
addNestedNsData	NestedNsInstanceData	0..N	The identifier of an existing nested NS instance to be added to (nested within) the NS instance. It shall be present only if updateType = "ADD_NESTED_NS".
removeNestedNsId	IdentifierInNs	0..N	The identifier of an existing nested NS instance to be removed from the NS instance. It shall be present only if updateType = "REMOVE_NESTED_NS".
assocNewNsdVersionData	AssocNewNsdVersionData	0..1	Specify the new NSD to be used for the NS instance. It shall be present only if updateType = "ASSOC_NEW_NSD_VERSION".
moveVnfInstanceData	MoveVnfInstanceData	0..N	Specify existing VNF instance to be moved from one NS instance to another NS instance. It shall be present only if updateType = "MOVE_VNF".
addVnffg	AddVnffgData	0..N	Specify the new VNFFG to be created to the NS Instance. It shall be present only if updateType = "ADD_VNFFG".
removeVnffgId	Identifier	0..N	Identifier of an existing VNFFG to be removed from the NS Instance. It shall be present only if updateType = "REMOVE_VNFFG".
updateVnffg	UpdateVnffgData	0..N	Specify the new VNFFG Information data to be updated for a VNFFG of the NS Instance. It shall be present only if updateType = "UPDATE_VNFFG".
changeNsFlavourData	ChangeNsFlavourData	0..1	Specifies the new DF to be applied to the NS instance. It shall be present only if updateType = "CHANGE_NS_DF".
addPnfData	AddPnfData	0..N	Specifies the PNF to be added into the NS instance. It shall be present only if updateType = "ADD_PNF".
modifyPnfData	ModifyPnfData	0..N	Specifies the PNF to be modified in the NS instance. It shall be present only if updateType = "MODIFY_PNF".
removePnfId	Identifier	0..N	Identifier of the PNF to be deleted from the NS instance. It shall be present only if updateType = "REMOVE_PNF".
modifyWanConnectionInfoData	ModifyWanConnectionInfoData	0..N	Specifies the data to modify about WAN related connectivity information. It shall be present only if updateType = "MODIFY_WAN_CONNECTION_INFO".
updateTime	DateTime	0..1	Timestamp indicating the update time of the NS, i.e. the NS will be updated at this timestamp. Cardinality "0" indicates the NS update takes place immediately.

Attribute name	Data type	Cardinality	Description
			<p>If the "Update NS" is a scheduled operation, it shall be possible to modify the schedule time, whereas it shall not be possible to change any other operational parameters.</p> <p>This attribute can be modified with the PATCH method. See note 6.</p>
createSnapshotData	CreateVnfSnapshotData	0..1	Specify the VNF instance to be snapshotted. It shall be present only if updateType = "CREATE_VNF_SNAPSHOT".
revertVnfToSnapshotData	RevertVnfToSnapshotData	0..1	Specify the VNF instance to be reverted and the VNF Snapshot to be reverted to. It shall be present only if updateType = "REVERT_VNF_TO_SNAPSHOT".
deleteVnfSnapshotData	DeleteVnfSnapshotData	0..1	Specify the VNF Snapshot info to be deleted and the related VNF instance. It shall be present only if updateType = "DELETE_VNF_SNAPSHOT_INFO".
addNsVirtualLinkData	AddNsVirtualLinkData	0..N	Specify data to be used to create a new NsVirtualLink instance. This parameter shall be present only if updateType = "CREATE_NS_VIRTUAL_LINK".
deleteNsVirtualLinkId	Identifier	0..N	Identify an existing NsVirtualLink instance to be deleted. The parameter shall be present only if updateType = "DELETE_NS_VIRTUAL_LINK".
feasibilityCheckReserve	FeasibilityCheckReserveType	0..1	<p>Specifies the feasibility check and reserve option for the Update NS operation.</p> <p>Default option is NO_FEASIBILITY_CHECK.</p> <p>If not present, default value applies. See notes 6, 7, 8, 9 and 10.</p>
createDataFlowMirroring	CreateDataFlowMirroring	0..N	Information of the Data Flow Mirroring that are associated with the NS instance. This parameter shall be present only if updateType = CREATE_DATA_FLOW_MIRRORING.
deleteDataFlowMirroringId	Identifier	0..N	Identifier of the "DataFlowMirroringData" structure that are associated with the NS instance to be deleted. This parameter shall be present only if updateType = DELETE_DATA_FLOW_MIRRORING.
updateDataFlowMirroring	UpdateDataFlowMirroring	0..N	Information on the Data Flow Mirroring associated to the NS instance to be updated. This parameter shall be present only if updateType = UPDATE_DATA_FLOW_MIRRORING.
<p>NOTE 1: If a VNF instance is removed from an NS and this NS was the last one for which this VNF instance was a part, the VNF instance is terminated by the NFVO.</p> <p>NOTE 2: It depends on the VNF capabilities, and is declared in the VNFD whether the operation is supported for a particular VNF.</p> <p>NOTE 3: The operation might be service-disruptive.</p> <p>NOTE 4: For each of the referred vnfInstanceId in the terminateVnfData, there shall be a corresponding value in the removeVnfInstanceId.</p> <p>NOTE 5: A change of VNFD version of a constituent VNF where the source and target VNFD versions have the same vnfExtInvariantId can also be performed without a change in the NSD by a sequence of two Update NS operations, the first one of type RemoveVnf, the second one of type AddVnf or of type InstantiateVnf; or with the Scale NS operation, removing the existing VNF instance and adding a VNF instance with the target VNFD.</p> <p>NOTE 6: When feasibilityCheckReserve is set to FEASIBILITY_CHECK_ONLY, the "updateTime" attribute shall be ignored.</p> <p>NOTE 7: When feasibilityCheckReserve is set to NO_FEASIBILITY_CHECK, the feasibility checking procedure is not performed. Resources might be reserved as triggered by means outside of the present operation request (e.g. configuration or policies), e.g. if it is a scheduled Update NS operation. See clause A.2.3 in ETSI GS NFV-IFA 010 [2].</p> <p>NOTE 8: When feasibilityCheckReserve is set to FEASIBILITY_CHECK_ONLY, only feasibility checking procedure is performed for the operation, no resources are reserved.</p> <p>NOTE 9: When feasibilityCheckReserve is set to FEASIBILITY_CHECK_WITH_OPERATION, the feasibility checking procedure is performed and based on the procedure result, the actual Update NS operation is performed at the time indicated by the "updateTime" attribute, if provided. Resources might be reserved as triggered by means outside of the present operation request (e.g. configuration or policies), e.g. if it is a scheduled Update NS operation. See clause A.2.3 in ETSI GS NFV-IFA 010 [2].</p>			

Attribute name	Data type	Cardinality	Description
NOTE 10: When feasibilityCheckReserve is set to FEASIBILITY_CHECK_WITH_RESERVATION_AND_OPERATION, the feasibility checking is performed and based on the procedure result, resources are reserved as part of the feasibility check, and the actual Update NS operation is performed using the resources reserved during the feasibility checking procedure execution, and at the time indicated by the "updateTime" attribute, if provided.			

### 6.5.2.13 Type: HealNsRequest

This type represents request parameters for the "Heal NS" operation. This operation supports the healing of an NS instance, either by healing the complete NS instance or by healing one of more of the VNF instances that are part of this NS. It shall comply with the provisions defined in table 6.5.2.13-1.

**Table 6.5.2.13-1: Definition of the HealNsRequest data type**

Attribute name	Data type	Cardinality	Description
healNsData	HealNsData	0..1	Provides the information needed to heal an NS. See note.
healVnfData	HealVnfData	0..N	Provides the information needed to heal a VNF. See note.
NOTE: Either the parameter healNsData or the parameter healVnfData, but not both shall be provided.			

### 6.5.2.14 Type: ScaleNsRequest

This type represents request parameters for the "Scale NS" operation. It shall comply with the provisions defined in table 6.5.2.14-1.

**Table 6.5.2.14-1: Definition of the ScaleNsRequest data type**

Attribute name	Data type	Cardinality	Description
scaleType	Enum (inlined)	1	Indicates the type of scaling to be performed. Possible values: - SCALE_NS - SCALE_VNF
scaleNsData	ScaleNsData	0..1	The necessary information to scale the referenced NS instance. It shall be present when scaleType = SCALE_NS. See note.
scaleVnfData	ScaleVnfData	0..N	The necessary information to scale the referenced NS instance. It shall be present when scaleType = SCALE_VNF. See note.
scaleTime	DateTime	0..1	Timestamp indicating the scale time of the NS, i.e. the NS will be scaled at this timestamp. Cardinality "0" indicates the NS scaling takes place immediately.  If the "Scale NS" is a scheduled operation, it shall be possible to modify the schedule time, whereas it shall not be possible to change any other operational parameters.  This attribute can be modified with the PATCH method.
NOTE: Either the parameter scaleNsData or the parameter scaleVnfData, but not both shall be provided.			

### 6.5.2.15 Type: TerminateNsRequest

This type represents request parameters for the "Terminate NS" operation. It shall comply with the provisions defined in table 6.5.2.15-1.

**Table 6.5.2.15-1: Definition of the TerminateNsRequest data type**

Attribute name	Data type	Cardinality	Description
terminationTime	DateTime	0..1	Timestamp indicating the end time of the NS, i.e. the NS will be terminated automatically at this timestamp. Cardinality "0" indicates the NS termination takes place immediately.  If the "Terminate NS" is a scheduled operation, it shall be possible to modify the schedule time, whereas it shall not be possible to change any other operational parameters.  This attribute can be modified with the PATCH method.
terminateNsData	TerminateNsData	0..1	Provides additional parameters to the termination process at the NS level. See note 1.
terminateVnfData	TerminateVnfData	0..N	Provides the information to terminate VNF instance(s). See notes 1 and 2.
NOTE 1: Information needed for terminating specific VNF instances shall only be specified in the "terminateVnfData" attribute, and not in the "terminateNsData" attribute.			
NOTE 2: VNF instance(s) part of this NS instance is(are) terminated as part of Terminate NS operation only if the instance(s) is(are) not used by any other NS instance.			

### 6.5.2.16 Type: CancelMode

This type represents a parameter to select the mode of cancelling an ongoing NS LCM operation occurrence. It shall comply with the provisions defined in table 6.5.2.16-1.

**Table 6.5.2.16-1: Definition of the CancelMode data type**

Attribute name	Data type	Cardinality	Description
cancelMode	CancelModeType	1	Cancellation mode to apply.

### 6.5.2.17 Type: VnfSnapshotInfo

This type represents an "Individual VNF snapshot" resource. The "id" attributed is used by the NFVO to index and identify the VNF snapshots information resources that are accessible via the NFVO. The identifier is still generated by the VNFM and copied into the present "VnfSnapshotInfo" representing the "Individual VNF snapshot" resource hold by the NFVO.

It shall comply with the provisions defined in table 6.5.2.17-1.

**Table 6.5.2.17-1: Definition of the VnfSnapshotInfo data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the "Individual VNF snapshot" resource. This identifier is allocated by the VNFM.
vnfSnapshotPkgId	Identifier	0..1	Identifier of the VNF snapshot package information held by the NFVO. Shall be present when the "Individual VNF snapshot" resource is created from a VNF snapshot package extraction.
vnfSnapshot	VnfSnapshot	0..1	Information about the VNF snapshot, content and/or references to its content. Shall be present when the "Individual VNF snapshot" resource is associated to a VNF snapshot on a VNFM, either created via the corresponding "Create VNF Snapshot" task resource or extracted from a VNF snapshot package.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.

### 6.5.2.18 Type: VnfSnapshot

This type represents a VNF Snapshot. It shall comply with the provisions defined in table 6.5.2.18-1.

**Table 6.5.2.18-1: Definition of the VnfSnapshot data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the VNF snapshot. This identifier is allocated by the VNFM.
vnfInstancelid	Identifier	1	Identifier of the snapshotted VNF instance.
creationStartedAt	DateTime	1	Timestamp indicating when the VNF Snapshot creation has been started by the VNFM.
creationFinishedAt	DateTime	1	Timestamp indicating when the VNF Snapshot has been completed by the VNFM.
vnfdId	Identifier	1	Identifier of the VNFD in use at the time the snapshot of the VNF instance has been created.
vnfInstance	VnfInstance	1	VNF instance information of the snapshotted VNF instance. This is a copy of the "Individual VNF instance" resource.
vnfcSnapshots	VnfcSnapshotInfo	1..N	Information about VNFC Snapshots constituting this VNF Snapshot.
userDefinedData	KeyValuePairs	0..1	User defined data for the VNF Snapshot.

### 6.5.2.19 Type: NsLcmCapacityShortageNotification

This type represents an NS LCM capacity shortage notification, which informs the receiver about resource shortage conditions during the execution of NS LCM operations. The notifications are triggered by the NFVO when a capacity shortage condition occurs during the execution of an NS LCM operation, which fails due to the resource shortage, or which succeeds despite the resource shortage because the NFVO has reduced the resource consumption of other NSs by requesting these NSs to be scaled in or terminated.

The notification shall comply with the provisions defined in table 6.5.2.19-1. The support of the notification is mandatory.

This notification shall be triggered by the NFVO when there is a capacity shortage condition during the execution of an NS LCM operation which will cause the LCM operation to be not successfully completed, or which will trigger the automatic executing of an LCM operation to reduce the resource consumption of one or more NS instances to resolve a resource shortage situation. The shortage conditions include:

- Necessary resources could not be allocated during an LCM operation because of resource shortage which causes the LCM operation to fail.
- An LCM operation on an NS instance with higher priority pre-empted an LCM operation on NS instance with lower priority because of resource shortage.
- An LCM operation on an NS instance with higher priority pre-empted an existing NS instance. Resources were de-allocated from the lower priority NS instance to allow the LCM operation on a higher priority NS instance.
- The resource capacity shortage situation has ended, and it can be expected that an LCM operation that had failed could succeed now if retried.

NOTE: ETSI GS NFV-IFA 013 [3] defines further shortage situations. These are not supported by the present version of the present document.

This notification shall also be triggered by the NFVO when a shortage condition has ended that has previously led to NS LCM operation occurrences failing. The notification shall be sent to all API consumers (OSS/BSS) that have subscribed to notifications related to capacity shortage and meeting the filter conditions of all pre-empted (low priority) and all pre-empting (high priority NS instance(s)). See ETSI GS NFV-IFA 010 [2], clause D.2 for the use cases.

The notification about the result of an unsuccessful LCM operation occurrence shall include appropriate information about the resource shortage when the cause for failure is a resource shortage.

The notification where a pre-emption occurred due to e.g. a higher priority LCM operation during resource shortage shall include appropriate information about the pre-emption.

**Table 6.5.2.19-1: Definition of the NsLcmCapacityShortageNotification data type**

Attribute	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsLcmCapacityShortageNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timestamp	DateTime	1	Date-time of the generation of the notification.
preemptingNsLcmOpOcclId	Identifier	0..1	Identifier of the LCM operation occurrence that has triggered a pre-emption.  Shall be absent when the instantiation of a lower priority NS instance has failed because all resources are allocated to higher priority NS instance(s). In this case the current instantiate operation is represented in "affectedCondition" with the value "PRE_EMPTED".
highPrioNsInstancelId	Identifier	0..1	Identifier of the higher priority NS instance affected by the lifecycle operation represented by the "preemptingNsLcmOpOcclId" attribute.  Shall be present if "preemptingNsLcmOpOcclId" is present and shall be absent otherwise.
status	Enum (Inlined)	1	Indicates the situation of capacity shortage.  Permitted values: <ul style="list-style-type: none"> <li>- LCM_RESOURCES_NOT_AVAILABLE: the lifecycle operation identified by the nsLcmOpOcclId attribute could not be completed because necessary resources were not available.</li> <li>- LCM_SHORTAGE_END: the shortage situation which has caused the lifecycle management operation identified by the nsLcmOpOcclId attribute to fail has ended.</li> </ul> See note 2.
shortageType	Enum (inlined)	0..1	Indicates whether this notification reports about a resource shortage or NFV-MANO capacity or performance shortage.  Permitted values: <ul style="list-style-type: none"> <li>- RESOURCE_SHORTAGE: the notification reports a resource shortage.</li> </ul> Shall be present when a resources shortage situation has been identified starts and the notification is sent with "status" = "LCM_RESOURCES_NOT_AVAILABLE" and shall be absent otherwise.
affectedOpOccs	Structure (inlined)	0..N	List of NS LCM operation occurrence(s) that were affected by the resource shortage.
>affectedNsId	Identifier	1	Identifier of the NS instance related to the operation occurrence that was affected by the shortage.
>affectedOpOcclId	Identifier	1	Identifier of the NS LCM operation occurrence that was affected by the shortage.



Attribute	Data type	Cardinality	Description
>affectedCondition	Enum (inlined)	1	This flag indicates in what condition (pre-empted or triggered) the operation occurrence(s) were affected by the resource shortage.  Permitted values: <ul style="list-style-type: none"> <li>- PRE_EMPTED: the operation was pre-empted (not executed due to resource shortage) and the "operationState" attribute of the NsLcmOpOcc structure identified by "affectedOpOcclId" attribute has the value "FAILED_TEMP" or "ROLLED_BACK".</li> <li>- TRIGGERED: the operation was triggered by NFVO and the "operationState" attribute of the NsLcmOpOcc structure identified by "affectedOpOcclId" attribute has the value "COMPLETED".</li> </ul> See note 1.
>_links	Structure(inlined)	1	Links related to NS resources affected by the shortage of this operation occurrence.
>>affectedLcmOpOcc	Link	1	Link related to the NS lifecycle management operation occurrence.
>>affectedNs	Link	1	Link related to the NS instance that was affected by the shortage.
capacityInformation	Structure(inlined)	0..N	References to NFVI capacity information related to the shortage.
>vimId	Identifier	1	Identifier of the VIM where the capacity shortage occurs.
>_link	Link	1	Link to the related "Individual VIM's capacity information" resource.
_links	Structure (inlined)	1	Links to resources related to this notification.
>preemptingNsLcmOpOcc	Link	0..1	Link to the resource representing the pre-empting LCM operation occurrence identified by preemptingNsLcmOpOcclId. Shall be present if preemptingNsLcmOpOcclId is present and shall be absent otherwise.
>highPrioNsInstance	Link	0..1	Link to the resource representing the high-priority NS instance identified by highPrioNsInstanceld. Shall be present if highPrioNsInstanceld is present and shall be absent otherwise.
>subscription	Link	1	Link to the subscription that triggered this notification.
NOTE 1: Not all operation occurrences that are in "FAILED_TEMP" have been pre-empted by a resource shortage. When the operation occurrences were pre-empted, the NS instances affected by the resource shortage end ("status" = "LCM_SHORTAGE_END") which are pointed by "affectedNsId" in the list of "affectedOpOccs" structures can have the operations in "FAILED_TEMP" state retried again (which needs a request by the OSS/BSS).			
NOTE 2: The present version of the present document supports only the resource shortage status enumeration values "LCM_RESOURCES_NOT_AVAILABLE" and "LCM_SHORTAGE_END" which represent a subset of the trigger conditions defined in clause 8.3.5.2 of ETSI GS NFV-IFA 013 [3].			

### 6.5.2.20 Type: NsLcmInfoModificationRequest

This type represents attribute modifications for an "Individual NS LCM operation occurrence" resource, i.e. modifications to a resource representation based on the "NsLcmOpOcc" data structure.

In case the NS LCM operation is a scheduled operation, it shall be possible to modify the schedule time, whereas it shall not be possible to change any other operational parameters. The attributes of the "NsLcmOpOcc" that can be modified are included in the "NsLcmInfoModificationRequest" data type.

The "NsLcmInfoModificationRequest" data type shall comply with the provisions defined in table 6.5.2.20-1.

**Table 6.5.2.20-1: Definition of the NsLcmInfoModificationRequest data type**

Attribute name	Data type	Cardinality	Description
scheduledTime	DateTime	1	This attribute specifies the modifications to the scheduled time of the specific NS LCM operation. See note.
NOTE: The "scheduledTime" attribute represents the new value of the "startTime" attribute of the "NsLcmOpOcc" data structure to start the NS LCM operation.			

## 6.5.3 Referenced structured data types

### 6.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but can neither be resource representations nor bound to any subscribe/notify mechanism.

### 6.5.3.2 Type: AffectedVnf

This type provides information about added, deleted and modified VNFs. It shall comply with the provisions in table 6.5.3.2-1.

**Table 6.5.3.2-1: Definition of the AffectedVnf data type**

Attribute name	Data type	Cardinality	Description
vnfInstanceid	Identifier	1	Identifier of the VNF instance.
vnfdid	Identifier	1	Identifier of the VNFD of the VNF Instance.
vnfProfileid	IdentifierInNsd	1	Identifier of the VNF profile of the NSD.
vnfName	String	1	Name of the VNF Instance.
changeType	Enum (inlined)	1	Signals the type of change.  Permitted values: <ul style="list-style-type: none"> <li>- ADD</li> <li>- REMOVE</li> <li>- INSTANTIATE</li> <li>- TERMINATE</li> <li>- SCALE</li> <li>- CHANGE_FLAVOUR</li> <li>- HEAL</li> <li>- OPERATE</li> <li>- MODIFY_INFORMATION</li> <li>- CHANGE_EXTERNAL_VNF_CONNECTIVITY</li> <li>- CHANGE_VNFPKG</li> </ul>
changeResult	Enum (inlined)	1	Signals the result of change identified by the "changeType" attribute.  Permitted values: <ul style="list-style-type: none"> <li>- COMPLETED</li> <li>- ROLLED_BACK</li> <li>- FAILED</li> </ul>
changedInfo	Structure (inlined)	0..1	Information about the changed VNF instance information, including VNF configurable properties, if applicable.
>changedVnfInfo	ModifyVnfInfoData	0..1	Information about the changed VNF instance information, including configurable properties, if applicable. See note.
>changedExtConnectivity	ExtVirtualLinkInfo	0..N	Information about changed external connectivity, if applicable. Only information about external VL instances that have been added or modified shall be provided. See note.

Attribute name	Data type	Cardinality	Description
>modificationsTriggeredByVnfPkgChange	ModificationsTriggeredByVnfPkgChange	0..1	Information about performed changes of "VnfInstance" attributes triggered by changing the current VNF package, if applicable. Shall be absent if the "operation" attribute is different from "CHANGE_VNFPKG". See note.
NOTE: At least one of the attributes "changedVnfInfo", "changedExtConnectivity" or "modificationsTriggeredByVnfPkgChange" shall be present. Not more than one of "changedVnfInfo" and "modificationsTriggeredByVnfPkgChange" shall be present.			

### 6.5.3.3 Type: AffectedPnf

This type provides information about added, deleted and modified PNFs. It shall comply with the provisions in table 6.5.3.3-1.

**Table 6.5.3.3-1: Definition of the AffectedPnf data type**

Attribute name	Data type	Cardinality	Description
pnfId	Identifier	1	Identifier of the affected PNF. This identifier is allocated by the OSS/BSS.
pnfDId	IdentifierInNsd	1	Identifier of the PNFD on which the PNF is based.
pnfProfileId	IdentifierInNsd	1	Identifier of the PNF profile of the NSD.
pnfName	String	1	Name of the PNF.
cpInstanceld	IdentifierInPnf	1..N	Identifier of the CP in the scope of the PNF.
changeType	Enum (inlined)	1	Signals the type of change.  Permitted values: - ADD - REMOVE - MODIFY
changeResult	Enum (inlined)	1	Signals the result of change identified by the "changeType" attribute.  Permitted values: - COMPLETED - ROLLED_BACK - FAILED

### 6.5.3.4 Type: AffectedVirtualLink

This type provides information about added, deleted and modified VLs. It shall comply with the provisions in table 6.5.3.4-1.

**Table 6.5.3.4-1: Definition of the AffectedVirtualLink data type**

Attribute name	Data type	Cardinality	Description
nsVirtualLinkInstanceld	IdentifierInNs	1	Identifier of the VL Instance.
nsVirtualLinkDesclD	IdentifierInNsd	1	Identifier of the VLD in the NSD for this VL.
vlProfileId	IdentifierInNsd	1	Name of the VL profile.
changeType	Enum (inlined)	1	Signals the type of change.  Permitted values: - ADD - DELETE - MODIFY - ADD_LINK_PORT - REMOVE_LINK_PORT

Attribute name	Data type	Cardinality	Description
linkPortIds	IdentifierInNs	0..N	Identifiers of the link ports of the affected VL related to the change. Each identifier references an "NsLinkPortInfo" structure.  Shall be set when changeType is equal to "ADD_LINK_PORT" or "REMOVE_LINK_PORT", and the related "NsLinkPortInfo" structures are present (case "add") or have been present (case "remove") in the "NsVirtualLinkInfo" structure that is represented by the "virtualLinkInfo" attribute in the "NsInstance" structure. See note.
changeResult	Enum (inlined)	1	Signals the result of change identified by the "changeType" attribute.  Permitted values: - COMPLETED - ROLLED_BACK - FAILED
NOTE: The resource handles of the affected NS link ports can be found by dereferencing the identifiers in the "linkPortIds" attribute.			

### 6.5.3.5 Type: AffectedVnffg

This type provides information about added, deleted and modified VNFFG instances. It shall comply with the provisions in table 6.5.3.5-1.

**Table 6.5.3.5-1: Definition of the AffectedVnffg data type**

Attribute name	Data type	Cardinality	Description
vnffgInstancelId	IdentifierInNs	1	Identifier of the VNFFG instance.
vnffgdId	IdentifierInNsd	1	Identifier of the VNFFGD of the VNFFG instance.
changeType	Enum (inlined)	1	Signals the type of lifecycle change.  Permitted values: - ADD - REMOVE - MODIFY
changeResult	Enum (inlined)	1	Signals the result of change identified by the "changeType" attribute.  Permitted values: - COMPLETED - ROLLED_BACK - FAILED

### 6.5.3.6 Type: AffectedNs

This type provides information about added, deleted and modified nested NSs. It shall comply with the provisions in table 6.5.3.6-1.

**Table 6.5.3.6-1: Definition of the AffectedNs data type**

Attribute name	Data type	Cardinality	Description
nsInstancelId	Identifier	1	Identifier of the nested NS instance.
nsdId	Identifier	1	Identifier of the NSD of the nested NS instance.
changeType	Enum (inlined)	1	Signals the type of lifecycle change.  Permitted values: - ADD - REMOVE - INSTANTIATE - INSTANTIATE - SCALE - UPDATE - HEAL - TERMINATE
changeResult	Enum (inlined)	1	Signals the result of change identified by the "changeType" attribute.  Permitted values: - COMPLETED - ROLLED_BACK - FAILED - PARTIALLY_COMPLETED
changedInfo	Structure (inlined)	0..1	Information about the changed NS instance information, if applicable.
>wanConnectionInfoModifications	WanConnectionInfoModification	0..N	Information about the modified WAN related connectivity information, if applicable.

### 6.5.3.7 Type: AffectedSap

This type provides information about added, deleted and modified SAP of an NS. It shall comply with the provisions in table 6.5.3.7-1.

**Table 6.5.3.7-1: Definition of the AffectedSap data type**

Attribute name	Data type	Cardinality	Description
sapInstancelId	IdentifierInNs	1	Identifier of the SAP instance.
sapId	IdentifierInNsd	1	Identifier of the SAPD for this SAP.
sapName	String	1	Human readable name for the SAP.
changeType	Enum (inlined)	1	Signals the type of lifecycle change.  Permitted values: - ADD - REMOVE - MODIFY
changeResult	Enum (inlined)	1	Signals the result of change identified by the "changeType" attribute.  Permitted values: - COMPLETED - ROLLED_BACK - FAILED

### 6.5.3.8 Type: LifecycleChangeNotificationsFilter

This type represents a subscription filter related to notifications about NS lifecycle changes. It shall comply with the provisions defined in table 6.5.3.8-1.

At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).

Table 6.5.3.8-1: Definition of the LifecycleChangeNotificationsFilter data type

Attribute name	Data type	Cardinality	Description
nsInstanceSubscriptionFilter	NsInstanceSubscriptionFilter	0..1	Filter criteria to select NS instances about which to notify.
notificationTypes	Enum (inlined)	0..N	Match particular notification types.  Permitted values: <ul style="list-style-type: none"> <li>- NsLcmOperationOccurrenceNotification</li> <li>- NsIdentifierCreationNotification</li> <li>- NsIdentifierDeletionNotification</li> <li>- NsLcmCapacityShortageNotification</li> <li>- NsChangeNotification</li> </ul> See note.
operationTypes	NsLcmOpType	0..N	Match particular NS lifecycle operation types for the notification of type NsLcmOperationOccurrenceNotification.  May be present if the "notificationTypes" attribute contains the value "NsLcmOperationOccurrenceNotification", and shall be absent otherwise.
operationStates	NsLcmOperationStateType	0..N	Match particular LCM operation state values as reported in notifications of type NsLcmOperationOccurrenceNotification.  May be present if the "notificationTypes" attribute contains the value "NsLcmOperationOccurrenceNotification", and shall be absent otherwise.
affectedNsInstanceIds	Identifier	0..N	Match particular identifiers of the NS instance(s) related to the operation occurrence that were affected by the shortage as reported in notifications of type NsLcmCapacityShortageNotification.  May be present if the "notificationTypes" attribute contains the value "NsLcmCapacityShortageNotification", and shall be absent otherwise.
nsComponentTypes	NsComponentType	0..N	Match particular NS component types for the notification of type NsChangeNotification.  May be present if the "notificationTypes" attribute contains the value "NsChangeNotification", and shall be absent otherwise.
lcmOpNameImpactingNsComponent	LcmOpNameForChangeNotificationType	0..N	Match particular LCM operation names for the notification of type NsChangeNotification.  May be present if the "notificationTypes" attribute contains the value "NsChangeNotification", and shall be absent otherwise.
lcmOpOccStatusImpactingNsComponent	LcmOpOccStatusForChangeNotificationType	0..N	Match particular LCM operation status values as reported in notifications of type NsChangeNotification.  May be present if the "notificationTypes" attribute contains the value "NsChangeNotification", and shall be absent otherwise.
NOTE: The permitted values of the "notificationTypes" attribute are spelled exactly as the names of the notification types to facilitate automated code generation systems.			

### 6.5.3.9 Type: LccnLinks

This type represents the links to resources that a notification can contain. It shall comply with the provisions defined in table 6.5.3.9-1.

**Table 6.5.3.9-1: Definition of the LccnLinks data type**

Attribute name	Data type	Cardinality	Description
nsInstance	NotificationLink	1	Link to the resource representing the NS instance to which the notified change applies.
subscription	NotificationLink	1	Link to the subscription that triggered this notification.
nsLcmOpOcc	NotificationLink	0..1	Link to the lifecycle operation occurrence that this notification is related to. Shall be present if there is a related lifecycle operation occurrence.

### 6.5.3.10 Type: SapData

This type represents the information related to a SAP of an NS. It shall comply with the provisions defined in table 6.5.3.10-1.

**Table 6.5.3.10-1: Definition of the SapData data type**

Attribute name	Data type	Cardinality	Description
sapId	IdentifierInNsd	1	Reference to the SAPD for this SAP.
sapName	String	1	Human readable name for the SAP.
description	String	1	Human readable description for the SAP.
sapProtocolData	CpProtocolData	0..N	Parameters for configuring the network protocols on the SAP.

### 6.5.3.11 Type: CpProtocolData

This type represents network protocol data. It shall comply with the provisions defined in table 6.5.3.11-1.

**Table 6.5.3.11-1: Definition of the CpProtocolData data type**

Attribute name	Data type	Cardinality	Description
layerProtocol	Enum (inlined)	1	Identifier of layer(s) and protocol(s).  Permitted values: - IP_OVER_ETHERNET - IP_FOR_VIRTUAL_CP  See note.
ipOverEthernet	IpOverEthernetAddressData	0..1	Network address data for IP over Ethernet to assign to the extCP instance. Shall be present if layerProtocol is equal to "IP_OVER_ETHERNET", and shall be absent otherwise.
virtualCpAddress	VirtualCpAddressData	0..1	IP address data to assign to an external CP instance exposing a virtual CP. It shall be present if layerProtocol is equal to "IP_FOR_VIRTUAL_CP" and the external CP instance exposes a virtual CP and shall not be present otherwise.
NOTE:	This attribute allows to signal the addition of further types of layer and protocol in future versions of the present document in a backwards-compatible way. In the current version of the present document, only IP over Ethernet is supported.		

### 6.5.3.12 Type: IpOverEthernetAddressData

This type represents network address data for IP over Ethernet. It shall comply with the provisions defined in table 6.5.3.12-1.

**Table 6.5.3.12-1: Definition of the IpOverEthernetAddressData data type**

Attribute name	Data type	Cardinality	Description
macAddress	MacAddress	0..1	MAC address. If this attribute is not present, it shall be chosen by the NFV MANO. See note 1.
segmentationType	Enum	0..1	Specifies the encapsulation type for the traffics coming in and out of the trunk subport.  Permitted values are: <ul style="list-style-type: none"> <li>- VLAN: The subport uses VLAN as encapsulation type.</li> <li>- INHERIT: The subport gets its segmentation type from the network it is connected to.</li> </ul> This attribute may be present for CP instances that represent subports in a trunk and shall be absent otherwise. If this attribute is not present for a subport CP instance, default value VLAN shall be used.
segmentationId	String	0..1	Identification of the network segment to which the CP instance connects to. See note 3 and note 4.
ipAddresses	Structure (inlined)	0..N	List of IP addresses to assign to the extCP instance. Each entry represents IP address data for fixed or dynamic IP address assignment per subnet.  If this attribute is not present, no IP address shall be assigned. See note 1.
>type	Enum (inlined)	1	The type of the IP addresses.  Permitted values: <ul style="list-style-type: none"> <li>- IPV4</li> <li>- IPV6</li> </ul>
>fixedAddresses	IpAddress	0..N	Fixed addresses to assign (from the subnet defined by "subnetId" if provided). See note 2.
>numDynamicAddresses	Integer	0..1	Number of dynamic addresses to assign (from the subnet defined by "subnetId" if provided). See note 2.
>addressRange	Structure (inlined)	0..1	An IP address range to be used, e.g. in case of egress connections.  In case this attribute is present, IP addresses from the range will be used. See note 2.
>>minAddress	IpAddress	1	Lowest IP address belonging to the range.
>>maxAddress	IpAddress	1	Highest IP address belonging to the range.
>subnetId	IdentifierInVim	0..1	Subnet defined by the identifier of the subnet resource in the VIM.  In case this attribute is present, IP addresses from that subnet will be assigned; otherwise, IP addresses not bound to a subnet will be assigned.



Attribute name	Data type	Cardinality	Description
NOTE 1: At least one of "macAddress" or "ipAddresses" shall be present.			
NOTE 2: Exactly one of "fixedAddresses", "numDynamicAddresses" or "ipAddressRange" shall be present.			
NOTE 3: If the CP instance represents a subport in a trunk, "segmentationId" shall be present. Otherwise it shall not be present.			
NOTE 4: Depending on the NFVI networking infrastructure, the "segmentationId" may indicate the actual network segment value (e.g. vlan Id, Vxlan segmentation id, etc.) used in the transport header of the packets or it may be an identifier used between the application and the NFVI networking infrastructure to identify the network sub-interface of the trunk port in question. In the latter case the NFVI infrastructure will map this local "segmentationId" to whatever "segmentationId" is actually used by the NFVI's transport technology.			

### 6.5.3.12a Type: VirtualCpAddressData

This type represents network address data for a virtual CP. It shall comply with the provisions defined in table 6.5.3.12a-1.

**Table 6.5.3.12a-1: Definition of the VirtualCpAddressData data type**

Attribute name	Data type	Cardinality	Description
type	Enum (inlined)	1	The type of the IP addresses.  Permitted values: - IPV4 - IPV6
loadBalancerIp	IpAddress	0..1	Fixed address to assign to an external load balancer. See notes 1 and 2.
NOTE 1: If the container cluster is set up to be able to configure an external load balancer this address will be used, otherwise it will be ignored by the CISM.			
NOTE 2: In case the cluster can configure an external load balancer but no loadBalancerIp is provided the container cluster will assign an IP address.			

### 6.5.3.13 Type: PnfInfo

This type represents the information about a PNF that is part of an NS instance. It shall comply with the provisions defined in table 6.5.3.13-1.

**Table 6.5.3.13-1: Definition of the PnfInfo data type**

Attribute name	Data type	Cardinality	Description
pnfId	Identifier	1	Identifier of the PNF. This identifier is allocated by the OSS/BSS.
pnfName	String	1	Name of the PNF.
pnfDId	Identifier	1	Identifier of the PNFD on which the PNF is based.
versionDependency	VersionDependency	0..N	Describes version dependencies currently valid for the PNF. Identifies versions of descriptors of other constituents in the NSD upon which the PNF depends. The dependencies may be described for the PNFD referenced in this PnfInfo with pnfDId and for PNFDs with the same PnfExtInvariantId.  There shall not be more than one versionDependency present with the same dependentConstituentId.
pnfDInfoId	Identifier	1	Identifier of the PNFD information object related to this PNF. This identifier is allocated by the NFVO.
pnfProfileId	IdentifierInNsd	1	Identifier of the related PnfProfile in the NSD on which the PNF is based.
cpInfo	PnfExtCpInfo	1..N	Information on the external CP of the PNF.

### 6.5.3.14 Type: AddPnfData

This type specifies an PNF to be added to the NS instance and the PNF Profile to use for this PNF. It shall comply with the provisions defined in table 6.5.3.14-1.

**Table 6.5.3.14-1: Definition of the AddPnfData data type**

Attribute name	Data type	Cardinality	Description
pnfId	Identifier	1	Identifier of the PNF. This identifier is allocated by the OSS/BSS.
pnfName	String	1	Name of the PNF.
pnfdId	Identifier	1	Identifier of the PNFD on which the PNF is based. If different to the one indicated in the referenced PnfProfile it replaces it when adding the PNF to an NS. The PNFD referenced by this attribute shall have the same pnfExtInvariantId as the one indicated in the PnfProfile. If the PnfProfile does not contain a pnfExtInvariantId this attribute shall be ignored. See note 1.
pnfProfileId	IdentifierInNsd	1	Identifier of related PnfProfile in the NSD on which the PNF is based.
cpData	PnfExtCpData	0..N	Address assigned for the PNF external CP(s).
overridingVersionDependency	OverridingVersionDependency	0..N	If present, information in each overridingVersionDependency replaces the versionDependency in the runtime information that the NFVO keeps about the VnfProfile, NsProfile or PnfProfile indicated in the OverridingVersionDependency.  Only the versionDependency in the VnfProfile, NsProfile or PnfProfile with the same dependentConstituentId as in the overridingVersionDependency is replaced. See note 2.  If no versionDependency with the dependentConstituentId indicated in the overridingVersionDependency exists in the VnfProfile, NsProfile or PnfProfile, the new versionDependency is added to the runtime information that the NFVO keeps about the profile.
NOTE 1: This attribute allows the use of a PNFD different from the one specified in the NSD when adding the PNF to the NS, provided they have the same pnfExtInvariantId.			
NOTE 2: A VnfProfile, NsProfile or PnfProfile may contain multiple VersionDependencies as it may describe the version dependencies of the descriptor referenced in the profile or of other descriptors with the same external invariant identifier.			

### 6.5.3.15 Type: ModifyPnfData

This type specifies an PNF to be modified in the NS instance. It shall comply with the provisions defined in table 6.5.3.15-1.

**Table 6.5.3.15-1: Definition of the ModifyPnfData data type**

Attribute name	Data type	Cardinality	Description
pnfId	Identifier	1	Identifier of the PNF. This identifier is allocated by the OSS/BSS.
pnfName	String	0..1	Name of the PNF. See note.
cpData	PnfExtCpData	0..N	Address assigned for the PNF external CP(s). See note.
NOTE: At least one attribute shall be present.			

### 6.5.3.16 Type: PnfExtCpData

This type represents the configuration data on the external CP of the PNF. It shall comply with the provisions defined in table 6.5.3.16-1.

**Table 6.5.3.16-1: Definition of the PnfExtCpData data type**

Attribute name	Data type	Cardinality	Description
cpInstancelId	IdentifierInPnf	0..1	Identifier of the CP. Shall be present for existing CP.
cpdId	IdentifierInNsd	0..1	Identifier of the Connection Point Descriptor (CPD) for this CP. Shall be present for new CP.
cpProtocolData	CpProtocolData	1..N	Address assigned for this CP.

### 6.5.3.17 Type: PnfExtCpInfo

This type represents the information about the external CP of the PNF. It shall comply with the provisions defined in table 6.5.3.17-1.

**Table 6.5.3.17-1: Definition of the PnfExtCpInfo data type**

Attribute name	Data type	Cardinality	Description
cpInstancelId	IdentifierInPnf	1	Identifier of the CP in the scope of the PNF.
cpdId	IdentifierInNsd	1	Identifier of (reference to) the Connection Point Descriptor (CPD) for this CP.
cpProtocolData	cpProtocolData	1..N	Parameters for configuring the network protocols on the CP.

### 6.5.3.18 Type: IpOverEthernetAddressInfo

This type represents information about a network address that has been assigned. It shall comply with the provisions defined in table 6.5.3.18-1.

**Table 6.5.3.18-1: Definition of the IpOverEthernetAddressInfo data type**

Attribute name	Data type	Cardinality	Description
macAddress	MacAddress	0..1	Assigned MAC address. See note 1.
segmentationId	String	0..1	Identification of the network segment to which the Cp instance connects to. See note 3 and note 4.
ipAddresses	Structure (inlined)	0..N	Addresses assigned to the CP or SAP instance. Each entry represents IP addresses assigned by fixed or dynamic IP address assignment per subnet. See note 1.
>type	Enum (inlined)	1	The type of the IP addresses.  Permitted values: - IPV4 - IPV6
>addresses	IpAddress	0..N	Fixed addresses assigned (from the subnet defined by "subnetId" if provided). See note 2.
>isDynamic	Boolean	0..1	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.
>addressRange	Structure (inlined)	0..1	An IP address range used, e.g. in case of egress connections. See note 2.
>>minAddress	IpAddress	1	Lowest IP address belonging to the range.
>>maxAddress	IpAddress	1	Highest IP address belonging to the range.
>subnetId	IdentifierInVim	0..1	Subnet defined by the identifier of the subnet resource in the VIM.  In case this attribute is present, IP addresses are bound to that subnet.

Attribute name	Data type	Cardinality	Description
NOTE 1:	At least one of "macAddress" or "ipAddresses" shall be present.		
NOTE 2:	Exactly one of "addresses" or "addressRange" shall be present.		
NOTE 3:	If the Cp instance represents a subport in a trunk, "segmentationId" shall be present. Otherwise it shall not be present.		
NOTE 4:	Depending on the NFVI networking infrastructure, the "segmentationId" may indicate the actual network segment value (e.g. vlan Id, Vxlan segmentation id, etc.) used in the transport header of the packets or it may be an identifier used between the application and the NFVI networking infrastructure to identify the network sub-interface of the trunk port in question. In the latter case the NFVI infrastructure will map this local "segmentationId" to whatever "segmentationId" is actually used by the NFVI's transport technology.		

### 6.5.3.18a Type: VirtualCpAddressInfo

This type represents information about a network address that has been assigned to a virtual CP. It shall comply with the provisions defined in table 6.5.3.18a-1.

**Table 6.5.3.18a-1: Definition of the VirtualCpAddressInfo data type**

Attribute name	Data type	Cardinality	Description
type	Enum (inlined)	1	The type of the IP addresses.  Permitted values: - IPV4 - IPV6
loadBalancerIp	IpAddress	0..1	Fixed addresses assigned to an external load balancer.

### 6.5.3.19 Type: VnfInstanceData

This type specifies an existing VNF instance to be used in the NS instance and if needed, the VNF Profile to use for this VNF instance. It shall comply with the provisions defined in table 6.5.3.19-1.

**Table 6.5.3.19-1: Definition of the VnfInstanceData data type**

Attribute name	Data type	Cardinality	Description
vnfInstanceid	Identifier	1	Identifier of the existing VNF instance to be used in the NS.
vnfProfileid	IdentifierInNsd	0..1	Identifier of (Reference to) a vnfProfile defined in the NSD which the existing VNF instance shall be matched with. If not present, the NFVO will select the VnfProfile matching the information in the VNF instance. See note 1.
overridingVnfId	Identifier	0..1	Reference to a VNFD. It replaces the vnfId indicated in the vnfProfileid. The VNFD referenced by this attribute shall have the same vnfExtInvariantId as the one indicated in the VnfProfile. If the VnfProfile does not contain a vnfExtInvariantId this attribute shall be ignored. If the VNFD of the existing VNF instance referenced by vnfInstanceid does not match the VNFD indicated by this attribute the VNF instance shall not be added to the NS instance. See notes 1, 2 and 4.

Attribute name	Data type	Cardinality	Description
overridingVersionDependency	OverridingVersionDependency	0..N	<p>If present, information in each overridingVersionDependency replaces the versionDependency in the runtime information that the NFVO keeps about the VnfProfile, NsProfile or PnfProfile indicated in the OverridingVersionDependency.</p> <p>Only the versionDependency in the VnfProfile, NsProfile or PnfProfile with the same dependentConstituentId as in the overridingVersionDependency is replaced. See note 3.</p> <p>If no versionDependency with the dependentConstituentId indicated in the overridingVersionDependency exists in the VnfProfile, NsProfile or PnfProfile, the new versionDependency is added to the runtime information that the NFVO keeps about the profile.</p> <p>See note 4.</p>
<p>NOTE 1: If the overridingVnfId attribute is present the vnfProfileId attribute shall also be present.</p> <p>NOTE 2: This attribute allows to use an existing VNF instance based on a different VNFD to the one specified in the NSD with vnfProfileId, provided both have the same vnfExtInvariantId.</p> <p>NOTE 3: A VnfProfile, NsProfile or PnfProfile may contain multiple VersionDependencies as it may describe the version dependencies of the descriptor referenced in the profile or of other descriptors with the same external invariant identifier.</p> <p>NOTE 4: The overridingVersionDependency attribute may only be present if the overridingVnfId attribute is present.</p>			

### 6.5.3.19a Type: NestedNsInstanceData

This type specifies an existing nested NS instance to be used in the NS instance and if needed, the NsProfile to use for this nested NS instance. It shall comply with the provisions defined in table 6.5.3.19a-1.

**Table 6.5.3.19a-1: Definition of the NestedNsInstanceData data type**

Attribute name	Data type	Cardinality	Description
nestedNsInstanceid	Identifier	1	Identifier of the existing nested NS instance to be used in the NS.
nsProfileid	IdentifierInNsd	0..1	Identifier of an NsProfile defined in the NSD which the existing nested NS instance shall be matched with. If not present, the NFVO will select the NsProfile matching the information in the nested NS instance. See note 1.
overridingNsId	Identifier	0..1	Reference to an NSD. It replaces the nsId indicated in the nsProfileid. The NSD referenced by this attribute shall have the same nsdExtInvariantId as the one indicated in the NsProfile. If the NsProfile does not contain an nsdExtInvariantId this attribute shall be ignored. If the NSD of the existing NS instance referenced by nestedNsInstanceid does not match the NSD indicated by this attribute the NS instance shall not be added as nested NS to the composite NS instance. See notes 1, 2 and 4.

Attribute name	Data type	Cardinality	Description
overridingVersionDependency	OverridingVersionDependency	0..N	<p>If present, information in each overridingVersionDependency replaces the versionDependency in the runtime information that the NFVO keeps about the VnfProfile, NsProfile or PnfProfile indicated in the OverridingVersionDependency.</p> <p>Only the versionDependency in the VnfProfile, NsProfile or PnfProfile with the same dependentConstituentId as in the overridingVersionDependency is replaced. See note 3.</p> <p>If no versionDependency with the dependentConstituentId indicated in the overridingVersionDependency exists in the VnfProfile, NsProfile or PnfProfile, the new versionDependency is added to the runtime information that the NFVO keeps about the profile.</p> <p>See note 4.</p>
<p>NOTE 1: If the overridingNsdlId attribute is present the nsProfileId attribute shall also be present.</p> <p>NOTE 2: This attribute allows to use as nested NS an existing NS instance based on a different NSD to the one specified in the composite NSD with nsProfileId, provided they have the same nsdExtInvariantId.</p> <p>NOTE 3: A VnfProfile, NsProfile or PnfProfile may contain multiple VersionDependencies as it may describe the version dependencies of the descriptor referenced in the profile or of other descriptors with the same external invariant identifier.</p> <p>NOTE 4: The overridingVersionDependency attribute may only be present if the overridingNsdlId attribute is present.</p>			

### 6.5.3.20 Type: VnfLocationConstraint

This type represents the association of location constraints to a VNF instance to be created according to a specific VNF profile. It shall comply with the provisions defined in table 6.5.3.20-1.

**Table 6.5.3.20-1: Definition of the VnfLocationConstraint data type**

Attribute name	Data type	Cardinality	Description
vnfProfileId	IdentifierInNsd	1	Identifier (reference to) of a VnfProfile in the NSD used to manage the lifecycle of the VNF instance.
locationConstraints	LocationConstraints	1	<p>Defines the location constraints for the VNF instance to be created based on the VNF profile.</p> <p>See note.</p>
<p>NOTE: These constraints are typically determined by the OSS/BSS from service requirements (e.g. latency requirements, regulatory requirements). The NFVO can map such location constraints to eligible NFVI-PoPs/resource zones where the VNF instance is to be created.</p>			

### 6.5.3.21 Type: LocationConstraints

This type represents location constraints for a VNF to be instantiated. The location constraints can be represented as follows:

- as a country code;
- as a civic address combined with a country code;
- as an area, conditionally combined with a country code.

The LocationConstraints data type shall comply with the provisions defined in table 6.5.3.21-1.

**Table 6.5.3.21-1: Definition of the LocationConstraints data type**

Attribute name	Data type	Cardinality	Description
countryCode	String	0..1	The two-letter ISO 3166 [13] country code in capital letters. Shall be present in case the "area" attribute is absent. May be absent if the "area" attribute is present (see note).
civicAddressElement	Structure (inlined)	0..N	Zero or more elements comprising the civic address. Shall be absent if the "area" attribute is present.
>caType	Integer	1	Describe the content type of caValue. The value of caType shall comply with section 3.4 of IETF RFC 4776 [8].
>caValue	String	1	Content of civic address element corresponding to the caType. The format caValue shall comply with section 3.4 of IETF RFC 4776 [8].
area	Polygon (see [20])	0..1	Geographic area. Shall be absent if the "civicAddressElement" attribute is present. The content of this attribute shall follow the provisions for the "Polygon" geometry object as defined in IETF RFC 7946 [20], for which the "type" member shall be set to the value "Polygon". See note.
NOTE:	If both "countryCode" and "area" are present, no conflicts should exist between the values of these two attributes. In case of conflicts, the API producer (i.e. the NFVO) shall disregard parts of the geographic area signalled by "area" that are outside the boundaries of the country signalled by "countryCode". If "countryCode" is absent, it is solely the "area" attribute that defines the location constraint.		

### 6.5.3.21a Type: ParamsForNestedNs

This type specifies additional parameters on a per-nested NS instance basis. It shall comply with the provisions defined in table 6.5.3.21a-1.

**Table 6.5.3.21a-1: Definition of the ParamsForNestedNs data type**

Attribute name	Data type	Cardinality	Description
nsProfileId	IdentifierInNsd	1	Identifier of an NSProfile to which the additional parameters apply.
additionalParam	KeyValuePairs	0..N	Additional parameters that are to be applied on a per nested NS instance.
overridingNsdId	Identifier	0..1	Reference to an NSD. If present it replaces the nsdId indicated in the referenced NsProfile at the instantiation of the nested NS. The NSD referenced by this attribute shall have the same nsdExtInvariantId as the one indicated in the NsProfile. If the NsProfile does not contain an nsdExtInvariantId this attribute shall be ignored. See notes 1 and 3.
overridingVersionDependency	OverridingVersionDependency	0..N	If present, information in each overridingVersionDependency replaces the versionDependency in the runtime information that the NFVO keeps about the VnfProfile, NsProfile or PnfProfile indicated in the OverridingVersionDependency.  Only the versionDependency in the VnfProfile, NsProfile or PnfProfile with the same dependentConstituentId as in the overridingVersionDependency is replaced. See note 2.  If no versionDependency with the dependentConstituentId indicated in the overridingVersionDependency exists in the VnfProfile, NsProfile or PnfProfile, the new versionDependency is added to the runtime information that the NFVO keeps about the profile.  See note 3.

Attribute name	Data type	Cardinality	Description
NOTE 1: This attribute allows for a nested NS at instantiation of the composite NS the use of an NSD different from the one specified in the NSD of the composite NS with nsProfileId, provided they have the same nsdExtInvariantId.			
NOTE 2: A VnfProfile, NsProfile or PnfProfile may contain multiple VersionDependencies as it may describe the version dependencies of the descriptor referenced in the profile or of other descriptors with the same external invariant identifier.			
NOTE 3: The overridingVersionDependency attribute may only be present if the overridingNsId attribute is present.			

### 6.5.3.22 Type: ParamsForVnf

This type defines the additional parameters for the VNF instance to be created associated with an NS instance. It shall comply with the provisions defined in table 6.5.3.22-1.

**Table 6.5.3.22-1: Definition of the ParamsForVnf data type**

Attribute name	Data type	Cardinality	Description
vnfProfileId	IdentifierInNsd	1	Identifier of the vnfProfile applicable to the VNF instance to which the additional parameters apply.
overridingVnfId	Identifier	0..1	Reference to a VNFD. If present it replaces the vnfId indicated in the referenced VnfProfile at the VNF instantiation. The VNFD referenced by this attribute shall have the same vnfExtInvariantId as the one indicated in the VnfProfile. If the VnfProfile does not contain a vnfExtInvariantId this attribute shall be ignored. See notes 1 and 3.
overridingVersionDependency	OverridingVersionDependency	0..N	If present, information in each overridingVersionDependency replaces the versionDependency in the runtime information that the NFVO keeps about the VnfProfile, NsProfile or PnfProfile indicated in the OverridingVersionDependency.  Only the versionDependency in the VnfProfile, NsProfile or PnfProfile with the same dependentConstituentId as in the overridingVersionDependency is replaced. See note 2.  If no versionDependency with the dependentConstituentId indicated in the overridingVersionDependency exists in the VnfProfile, NsProfile or PnfProfile, the new versionDependency is added to the runtime information that the NFVO keeps about the profile.  See note 3.
vnfInstanceName	String	0..1	Human-readable name of the VNF instance to be created.
vnfInstanceDescription	String	0..1	Human-readable description of the VNF instance to be created.
vnfConfigurableProperties	KeyValuePairs	0..1	If present, this attribute provides values for the configurable properties declared in the VNFD. These values will override the default values if default values are also declared in the VNFD.  It provides values for the "vnfConfigurableProperties" input parameter of the Instantiate VNF operation defined in ETSI GS NFV-SOL 003 [4].
metadata	KeyValuePairs	0..1	If present, this attribute provides values for metadata Metadata can but need not be declared in the VNFD. These values will override the default values if default values are also declared in the VNFD.  It provides the values for the "metadata" input parameter of the Create VNF Identifier operation defined in ETSI GS NFV-SOL 003 [4].



Attribute name	Data type	Cardinality	Description
extensions	KeyValuePairs	0..1	If present, this attribute provides values for the extensions declared in the VNFD. These values will override the default values if default values are also declared in the VNFD.  It provides the values for the "extensions" input parameter of the Instantiate VNF operation defined in ETSI GS NFV-SOL 003 [4].
additionalParams	KeyValuePairs	0..1	Additional input parameters for the instantiation process, specific to the VNF being instantiated, as declared in the VNFD as part of "InstantiateVnfOpConfig" defined in ETSI GS NFV-IFA 011 [29].
NOTE 1: This attribute allows at VNF instantiation the use of a VNFD different from the one specified in the NSD with vnfProfileId provided the two VNFDs refer to the same vnfExtInvariantId.			
NOTE 2: A VnfProfile, NsProfile or PnfProfile may contain multiple VersionDependencies as it may describe the version dependencies of the descriptor referenced in the profile or of other descriptors with the same external invariant identifier.			
NOTE 3: The overridingVersionDependency attribute may only be present if the overridingVnfId attribute is present.			

### 6.5.3.23 Type: AffinityOrAntiAffinityRule

This type describes the additional affinity or anti-affinity rule applicable between the VNF instances to be instantiated in the NS instantiation operation request or between the VNF instances to be instantiated in the NS instantiation operation request and the existing VNF instances. It shall comply with the provisions defined in table 6.5.3.23-1.

**Table 6.5.3.23-1: Definition of the AffinityOrAntiAffinityRule data type**

Attribute name	Data type	Cardinality	Description
vnfdId	Identifier	0..N	Reference to a VNFD. When the VNFD which is not used to instantiate VNF, it presents all VNF instances of this type as the subjects of the affinity or anti-affinity rule. The VNF instance which the VNFD presents is not necessary as a part of the NS to be instantiated.
vnfProfileId	IdentifierInNsd	1..N	Reference to a vnfProfile defined in the NSD. At least one VnfProfile which is used to instantiate VNF for the NS to be instantiated as the subject of the affinity or anti-affinity rule shall be present. When the VnfProfile which is not used to instantiate VNF, it presents all VNF instances of this type as the subjects of the affinity or anti-affinity rule. The VNF instance which the VnfProfile presents is not necessary as a part of the NS to be instantiated.
vnfInstanceId	Identifier	0..N	Reference to the existing VNF instance as the subject of the affinity or anti-affinity rule. The existing VNF instance is not necessary as a part of the NS to be instantiated.
affinityOrAntiAffinity	Enum (inlined)	1	The type of the constraint.  Permitted values: - AFFINITY - ANTI_AFFINITY
scope	Enum (inlined)	1	Specifies the scope of the rule where the placement constraint applies.  Permitted values: - NFVI_POP - ZONE - ZONE_GROUP - NFVI_NODE

### 6.5.3.24 Type: InstantiateVnfData

This type represents the information that are needed for VNF instantiation. This information element is used for the bottom-up NS creation when the OSS/BSS explicitly requests VNF instantiation for a given NS. When the NFVO invokes the Instantiate VNF update operation, a set of these parameters are then passed by the NFVO to the VNFM. It shall comply with the provisions defined in table 6.5.3.24-1.

**Table 6.5.3.24-1: Definition of the InstantiateVnfData data type**

Attribute name	Data type	Cardinality	Description
vnfdId	Identifier	0..1	Information sufficient to identify the VNFD which defines the VNF to be instantiated. See notes 2 and 3.
vnfFlavourId	IdentifierInVnfd	0..1	Identifier of the VNF deployment flavour to be instantiated. See notes 2 and 3.
vnfInstantiationLevelId	IdentifierInVnfd	0..1	Identifier of the instantiation level of the deployment flavour to be instantiated. See notes 2, 3 and 4.
targetScaleLevelInfo	VnfScaleInfo	0..N	This attribute is applicable if VNF supports target scale level instantiation.  For each scaling aspect of the current deployment flavour, the attribute specifies the scale level of VNF constituents (e.g. VDU level) to be instantiated. See notes 4 and 5.
vnfProfileId	IdentifierInNsd	0..1	Identifier of (Reference to) a vnfProfile defined in the NSD which is used for instantiating the VNF. See notes 3 and 6.
vnfInstanceName	String	0..1	Human-readable name of the VNF instance to be created.
vnfInstanceDescription	String	0..1	Human-readable description of the VNF instance to be created.
extVirtualLinks	ExtVirtualLinkData	0..N	Information about external VLs to connect the VNF to.
extManagedVirtualLinks	ExtManagedVirtualLinkData	0..N	Information about internal VLs that are managed by other entities than the VNFM. See note 1.
localizationLanguage	String	0..1	Localization language of the VNF to be instantiated. The value shall comply with the format defined in IETF RFC 5646 [9].
vnfConfigurableProperties	KeyValuePairs	0..1	If present, this attribute provides values for the configurable properties declared in the VNFD. These values will override the default values if default values are also declared in the VNFD.  It provides values for the "vnfConfigurableProperties" input parameter of the Instantiate VNF operation defined in ETSI GS NFV-SOL 003 [4].
additionalParams	KeyValuePairs	0..1	Additional input parameters for the instantiation process, specific to the VNF being instantiated, as declared in the VNFD as part of "InstantiateVnfOpConfig" defined in ETSI GS NFV-IFA 011 [29].

Attribute name	Data type	Cardinality	Description
metadata	KeyValuePairs	0..1	<p>If present, this attribute provides values for metadata. Metadata can but need not be declared in the VNFD. These values will override the default values if default values are also declared in the VNFD.</p> <p>This attribute provides values for the "metadata" input parameter of the Create VNF Identifier operation defined in ETSI GS NFV-SOL 003 [4].</p>
extensions	KeyValuePairs	0..1	<p>If present, this attribute provides values for extensions declared in the VNFD. These values will override the default values if default values are also declared in the VNFD.</p> <p>This attribute provides values for the "extensions" input parameter of the Instantiate VNF operation defined in ETSI GS NFV-SOL 003 [4].</p>
locationConstraints	VnfLocationConstraint	0..1	<p>Defines the location constraints for the VNF to be instantiated as part of the NS Update. An example can be a constraint for the VNF to be in a specific geographic location.</p>
overridingVnfldd	Identifier	0..1	<p>Reference to a VNFD.</p> <p>If present the vnfProfileId shall also be present. It replaces the vnfldd indicated in the vnfProfileId. The VNFD referenced by this attribute shall have the same vnfExtInvariantId as the one indicated in the VnfProfile. If the VnfProfile does not contain a vnfExtInvariantId this attribute shall be ignored.</p> <p>See notes 6, 7 and 9.</p>
overridingVersionDependency	OverridingVersionDependency	0..N	<p>If present, information in each overridingVersionDependency replaces the versionDependency in the runtime information that the NFVO keeps about the VnfProfile, NsProfile or PnfProfile indicated in the OverridingVersionDependency.</p> <p>Only the versionDependency in the VnfProfile, NsProfile or PnfProfile with the same dependentConstituentId as in the overridingVersionDependency is replaced. See note 8.</p> <p>If no versionDependency with the dependentConstituentId indicated in the overridingVersionDependency exists in the VnfProfile, NsProfile or PnfProfile, the new version Dependency is added to the runtime information that the NFVO keeps about the profile.</p> <p>See note 9.</p>

Attribute name	Data type	Cardinality	Description
NOTE 1:	It is possible to have several ExtManagedVirtualLinkData for the same VNF internal VL in case of a multi-site VNF spanning several VIMs. The set of ExtManagedVirtualLinkData corresponding to the same VNF internal VL shall indicate so by referencing to the same VnfVirtualLinkDesc and externally-managed multi-site VL instance (refer to clause 6.5.3.27).		
NOTE 2:	If vnfId and vnfFlavourId (and vnfInstantiationLevelId, if provided) are present, there should be only one vnfProfile that matches the vnfId and vnfFlavourId (and vnfInstantiationLevelId, if present) in the NS deployment flavour specified in the NSD associated to the NS instance to which the present operation is triggered. In the case there is more than one matching vnfProfile, the NFVO may select a matching vnfProfile based on other information, such as external VL.		
NOTE 3:	Either the attribute triple "vnfId, vnfFlavourId and vnfInstantiationLevelId (if provided)" or the attribute "vnProfileId" shall be present, but not both.		
NOTE 4:	The target size for VNF instantiation may be specified in either vnfInstantiationLevelId or targetScaleLevelInfo, but not both. If none of the two attributes (vnfInstantiationLevelId or targetScaleLevelInfo) are present, the default instantiation level as declared in the VNFD shall be used.		
NOTE 5:	If targetScaleLevelInfo is specified, information provided in targetScaleLevelInfo shall be used for instantiating scalable constituents of the VNF (e.g. VDUs/VLs). For scaling aspects not specified in targetScaleLevelInfo or for the VNF constituents (e.g. VDUs/VLs) that are not scalable, the default instantiation level as declared in the VNFD shall be used for instantiation.		
NOTE 6:	If the overridingVnfId attribute is present the vnfProfileId attribute shall also be present.		
NOTE 7:	This attribute allows at VNF instantiation the use of a VNFD different from the one specified in the NSD with vnfProfileId provided the two VNFDs have the same vnfExtInvariantId.		
NOTE 8:	A VnfProfile, NsProfile or PnfProfile may contain multiple VersionDependencies as it may describe the version dependencies of the descriptor referenced in the profile or of other descriptors with the same external invariant identifier.		
NOTE 9:	The overridingVersionDependency attribute may only be present if the overridingVnfId attribute is present.		

### 6.5.3.25 Type: ChangeVnfFlavourData

The type represents the information that is requested to be changed deployment flavour for an existing VNF instance. It shall comply with the provisions defined in table 6.5.3.25-1.

**Table 6.5.3.25-1: Definition of the ChangeVnfFlavourData data type**

Attribute name	Data type	Cardinality	Description
vnfInstanceid	Identifier	1	Identifier of the VNF instance to be modified.
newFlavourId	IdentifierInVnfd	1	Identifier of the VNF deployment flavour to be instantiated.
instantiationLevelId	IdentifierInVnfd	0..1	Identifier of the instantiation level of the deployment flavour to be instantiated. See note 3.
targetScaleLevelInfo	VnfScaleInfo	0..N	This attribute is applicable if VNF supports target scale level instantiation.  For each scaling aspect of the current deployment flavour, the attribute specifies the scale level of VNF constituents (e.g. VDU level) to be instantiated. See notes 3 and 4.
extVirtualLinks	ExtVirtualLinkData	0..N	Information about external VLs to connect the VNF to. Entries in the list of external VLs that are unchanged need not be supplied.
extManagedVirtualLinks	ExtManagedVirtualLinkData	0..N	Information about internal VLs that are managed by NFVO. See note 1 and note 2.
additionalParams	KeyValuePairs	0..1	Additional input parameters for the flavour change process, specific to the VNF being modified, as declared in the VNFD as part of "ChangeVnfFlavourOpConfig" defined in ETSI GS NFV-IFA 011 [29].
extensions	KeyValuePairs	0..1	This attribute provides values for the "extensions" input parameter of the Change VNF Flavour operation defined in ETSI GS NFV-SOL 003 [4].
vnfConfigurableProperties	KeyValuePairs	0..1	This attribute provides values for the "vnfConfigurableProperties" input parameter of the Change VNF Flavour operation defined in ETSI GS NFV-SOL 003 [4].

Attribute name	Data type	Cardinality	Description
NOTE 1:	The indication of externally-managed internal VLs is needed in case networks have been pre-configured for use with certain VNFs, for instance to ensure that these networks have certain properties such as security or acceleration features, or to address particular network topologies. The present document assumes that externally-managed internal VLs are managed by the NFVO and created towards the VIM.		
NOTE 2:	It is possible to have several ExtManagedVirtualLinkData for the same VNF internal VL in case of a multi-site VNF spanning several VIMs. The set of ExtManagedVirtualLinkData corresponding to the same VNF internal VL shall indicate so by referencing to the same VnfVirtualLinkDesc and externally-managed multi-site VL instance (refer to clause 6.5.3.27).		
NOTE 3:	The target size for VNF instantiation may be specified in either instantiationLevelId or targetScaleLevelInfo, but not both. If none of the two attributes (instantiationLevelId or targetScaleLevelInfo) are present, the default instantiation level as declared in the VNFD shall be used.		
NOTE 4:	If targetScaleLevelInfo is specified, information provided in targetScaleLevelInfo shall be used for instantiating scalable constituents of the VNF (e.g. VDUs/VLs). For scaling aspects not specified in targetScaleLevelInfo or for the VNF constituents (e.g. VDUs/VLs) that are not scalable, the default instantiation level as declared in the VNFD shall be used for instantiation.		

### 6.5.3.26 Type: ExtVirtualLinkData

This type represents an external VL. It shall comply with the provisions defined in table 6.5.3.26-1.

**Table 6.5.3.26-1: Definition of the ExtVirtualLinkData data type**

Attribute name	Data type	Cardinality	Description
extVirtualLinkId	Identifier	0..1	The identifier of the external VL instance, if provided.
vimId	Identifier	0..1	Identifier of the VIM that manages this resource. This attribute shall only be supported and present if VNF-related resource management in direct mode is applicable.
resourceProviderId	Identifier	0..1	Identifies the entity responsible for the management of this resource. This attribute shall only be supported and present if VNF-related resource management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
resourceId	IdentifierInVim	1	The identifier of the resource in the scope of the VIM or the resource provider.
extCps	VnfExtCpData	1..N	External CPs of the VNF to be connected to this external VL.
extLinkPorts	ExtLinkPortData	0..N	Externally provided link ports to be used to connect external connection points to this external VL unless the extCp exposes a VIP CP and a link port is not needed for it based on the conditions defined below. See note 1.
extNetAttDefResourceData	NetAttDefResourceData	0..N	Externally provided network attachment definition resource(s) that provides the specification of the interface to attach external CPs to this external VL. See note 2.  It is only applicable if the external VL is realized by a secondary container cluster network. It shall not be present otherwise.
NOTE 1:	A link port is not needed for an external CP instance that exposes a VIP CP in the following cases: 1) For a VIP CP directly exposed as extCP: 1.1) no dedicated IP address is allocated as VIP address, as indicated in the VNFD; 1.2) a dedicated IP address is allocated as VIP address, but the NFVO indicates that no port is needed (createExtLinkPort in VnfExtCpConfig set to false). 2) For a VIP CP exposed as extCP via a floating IP address: 2.1) no dedicated IP address is allocated as VIP address, as indicated in the VNFD, and the VNFC CP associated to the VIP CP is also exposed via a floating IP address.		
NOTE 2:	An example of the network attachment definition resource when the container infrastructure service is a Kubernetes® instance is a Network Attachment Definition (NAD).		

### 6.5.3.27 Type: ExtManagedVirtualLinkData

This type represents an externally-managed internal VL. It shall comply with the provisions defined in table 6.5.3.27-1.

**Table 6.5.3.27-1: Definition of the ExtManagedVirtualLinkData data type**

Attribute name	Data type	Cardinality	Description
extManagedVirtualLinkId	Identifier	0..1	The identifier of the externally-managed internal VL instance, if provided.
vnfVirtualLinkDescId	IdentifierInVnfd	1	The identifier of the VLD in the VNFD for this VL.
vimId	Identifier	0..1	Identifier of the VIM that manage this resource. This attribute shall only be supported and present if VNF-related resource management in direct mode is applicable.
resourceProviderId	Identifier	0..1	Identifies the entity responsible for the management of this resource. This attribute shall only be supported and present if VNF-related resource management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
resourceId	IdentifierInVim	1	The identifier of the resource in the scope of the VIM or the resource provider.
vnfLinkPort	VnfLinkPortData	0..N	Externally provided link ports to be used to connect VNFC connection points to this externally-managed VL on this network resource. If this attribute is not present, the NFVO or the VNFM shall create the link ports on the externally-managed VL.
netAttDefResourceData	NetAttDefResourceData	0..N	Externally provided network attachment definition resource(s) that provides the specification of the interface to attach VNFC connection points to this externally-managed VL. If this attribute is not present, the NFVO shall create the network attachment definition resource(s) for the externally-managed VL. See notes 1 and 2.
intCp	IntVnfCpData	0..N	Internal CPs of the VNF to be connected to this externally-managed VL. See note 1.  This attribute may only be present if the "netAttDefResourceData" is also present.
extManagedMultisiteVirtualLinkId	Identifier	0..1	Identifier of the externally-managed multi-site VL instance. The identifier is assigned by the NFV-MANO entity that manages the externally managed multi-site VL instance. It shall be present when the present externally-managed internal VL (indicated by extManagedVirtualLinkId) is part of a multi-site VL, e.g. in support of multi-site VNF spanning several VIMs. All externally-managed internal VL instances corresponding to an internal VL created based on the same virtualLinkDescId shall refer to the same extManagedMultisiteVirtualLinkId.
NOTE 1: It is only applicable if the externally-managed VL is realized by a secondary container cluster network. It shall not be present otherwise.			
NOTE 2: An example of the network attachment definition resource when the container infrastructure service is a Kubernetes® instance is a Network Attachment Definition (NAD).			

### 6.5.3.28 Type: ExtLinkPortData

This type represents an externally provided link port to be used to connect a VNF external connection point to an external VL. It shall comply with the provisions defined in table 6.5.3.28-1.

**Table 6.5.3.28-1: Definition of the ExtLinkPortData data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this link port as provided by the entity that has created the link port.
resourceHandle	ResourceHandle	1	Reference to the virtualised resource realizing this link port.
trunkResourceId	IdentifierInVim	0..1	Identifier of the trunk resource in the VIM. Shall be present if the present link port corresponds to the parent port that the trunk resource is associated with. See note.

NOTE: The value of "trunkResourceId" is scoped by the value of "vimConnectionId" in the "resourceHandle" attribute.

### 6.5.3.29 Type: VnfExtCpData

This type represents configuration information for external CPs created from a CPD. It shall comply with the provisions defined in table 6.5.3.29-1.

**Table 6.5.3.29-1: Definition of the VnfExtCpData data type**

Attribute name	Data type	Cardinality	Description
cpdId	IdentifierInVnfd	1	The identifier of the CPD in the VNFD. See note 1.
cpConfig	map(VnfExtCpConfig)	1..N	Map of instance data that need to be configured on the CP instances created from the respective CPD. The key of the map which identifies the individual VnfExtCpConfig entries is of type "IdentifierInVnf" and is managed by the API consumer. The entries shall be applied by the VNFM according to the rules of JSON Merge Patch (see IETF RFC 7396 [11]). See notes 2, 3 and 4.

NOTE 1: In case this identifier refers to a CPD with trunking enabled, the external CP instances created from this CPD will represent ports in a trunk.

NOTE 2: The map entry value shall be set to "null" in order to delete a "VnfExtCpConfig" entry identified by a particular key value from the map, i.e. for the disconnection of an existing external CP instance addressed by cpInstanceId in the deleted map entry from a particular external virtual link, and deletion of that instance in case it represents a subport. Deleting the last key from the map removes the affected instance of the "VnfExtCpData" structure from its parent data structure.

NOTE 3: Within one VNF instance, all VNFC instances created from a particular VDU have the same external connectivity. Thus, given a particular value of the "cpdId" attribute, there shall be one "cpConfig" entry for each VNFC instance that has been or can be created from a VDU which includes a CPD identified by the "cpdId" attribute. If the cpConfig represents a subport in a trunk, all "cpConfig" entries in this list shall have the same segmentationId, which means they are connected to the same set of external VLS via the trunk.

NOTE 4: If, as defined by the input parameters of a "ChangeVnfFlavour", "ChangeExtVnfConnectivity" or "ChangeCurrentVnfPkg" operation, a cpConfig map entry identified by a particular map key value is moved into another "ExtVirtualLinkData" or "VnfExtCpData" structure, this particular cpConfig map entry may be used by an external CP instance different than the one that has used it before the operation, or by no external CP instance at all. Renaming a CPD identifier during the "changeCurrentVnfPkg" operation does not count as moving the related "cpConfig" map entries to a new "extCpData" structure.

### 6.5.3.30 Type: VnfExtCpConfig

This type represents an externally provided link port, or a network attachment definition resource of secondary container cluster network, or network address information per instance of a VNF external connection point.

In the case of VM-based deployment of the VNFC exposing the external CP:

- In case a link port is provided, the NFVO shall use that link port when connecting the VNF external CP to the external VL.
- In case a link port is not provided, the NFVO or VNFM shall create a link port on the external VL, and use that link port to connect the VNF external CP to the external VL.

In the case of container-based deployment of the VNFC exposing the external CP, the NFVO and VNFM shall use the network attachment definition resource of secondary container cluster network when connecting the CP to the external VL.

This type shall comply with the provisions defined in table 6.5.3.30-1.

**Table 6.5.3.30-1: Definition of the VnfExtCpConfig data type**

Attribute name	Data type	Cardinality	Description
parentCpConfigId	IdentifierInVnf	0..1	Value of the key that identifies the "VnfExtCpConfig" map entry which corresponds to the parent port of the trunk. Only present in "VnfExtCpConfig" structures that provide configuration information for a CP which represents a subport in a trunk, and if parent ports are supported. See note 2.
linkPortId	Identifier	0..1	Identifier of a pre-configured link port to which the external CP will be associated. See notes 1 and 5.
createExtLinkPort	Boolean	0..1	Indicates the need to create a dedicated link port for the external CP.  If set to True, a link port is created. If set to False, no link port is created.  This attribute is only applicable for external CP instances without a floating IP address that expose a VIP CP instance for which a dedicated IP address is allocated. It shall be present in that case and shall be absent otherwise.
cpProtocolData	CpProtocolData	0..N	Parameters for configuring the network protocols on the link port that connects the CP to a VL. See notes 1 and 3.
netAttDefResourceId	IdentifierInVnf	0..N	Identifier of the "NetAttDefResourceData" structure that provides the specification of the interface to attach the external CP to a secondary container cluster network.  It is only applicable if the external CP is connected or to be connected to a secondary container cluster network. It shall not be present if the external CP is related to a virtual network not categorized as secondary container cluster network.  See notes 3, 4 and 5.
<p>NOTE 1: The following conditions apply to the attributes "linkPortId" and "cpProtocolData" for an external CP instance connected or to be connected to a virtual network not categorized as secondary container cluster network:</p> <ul style="list-style-type: none"> <li>– At least one of the "linkPortId" and "cpProtocolData" attributes shall be present for an external CP instance representing a subport that is to be created, or an external CP instance that is to be created by creating the corresponding VNFC or VNF instance during the current or a subsequent LCM operation, or for an existing external CP instance that is to be re-configured or added to a particular external virtual link.</li> <li>– If the "cpProtocolData" attribute is absent, the "linkPortId" attribute shall be provided referencing a pre-created link port with pre-configured address information.</li> <li>– If both "cpProtocolData" and "linkPortId" are provided, the API consumer shall ensure that the "cpProtocolData" can be used with the pre-created link port referenced by "linkPortId".</li> </ul> <p>NOTE 2: In case the NFVO manages its own identifier space, the NFVO may remap this identifier when communicating with the VNFM. If the NFVO knows that there can be an identifier collision when communicating with the VNFM by using the identifier from the OSS/BSS, the NFVO shall remap it.</p> <p>NOTE 3: The following conditions apply to the attributes "netAttDefResourceId" and "cpProtocolData" for an external CP instance connected or to be connected to a secondary container cluster network:</p> <ul style="list-style-type: none"> <li>– The "netAttDefResourceId" and "cpProtocolData" attributes shall both be absent for the deletion of an existing external CP instance addressed by "cpInstancelId".</li> <li>– At least one of these attributes shall be present for a to-be-created external CP instance or an existing external CP instance.</li> </ul>			



Attribute name	Data type	Cardinality	Description
NOTE 4: Cardinality greater than 1 is only applicable for specific cases where more than one network attachment definition resource is needed to fulfil the connectivity requirements of the external CP, e.g. to build a link redundant mated pair in SR-IOV cases. When more than one "netAttDefResourceId" is indicated, all shall belong to the same namespace as defined by the corresponding "netAttDefResourceNamespace" attribute in the "NetAttDefResourceData".			
NOTE 5: Either "linkPortId" or "netAttDefResourceId" may be included, but not both.			

### 6.5.3.31 Type: OperateVnfData

This type represents a VNF instance for which the operational state needs to be changed and the requested new state. It shall comply with the provisions defined in table 6.5.3.31-1.

**Table 6.5.3.31-1: Definition of the OperateVnfData data type**

Attribute name	Data type	Cardinality	Description
vnfInstanceId	Identifier	1	Identifier of the VNF instance.
changeStateTo	OperationalStates	1	The desired operational state (i.e. started or stopped) to change the VNF to.
stopType	StopType	0..1	It signals whether forceful or graceful stop is requested. See note.
gracefulStopTimeout	Integer	0..1	The time interval (in seconds) to wait for the VNF to be taken out of service during graceful stop, before stopping the VNF. See note.
additionalParam	KeyValuePairs	0..1	Additional parameters passed by the OSS/BSS as input to the Operate VNF operation, specific to the VNF being operated.
NOTE: The "stopType" and "gracefulStopTimeout" attributes shall be absent, when the "changeStateTo" attribute is equal to "STARTED". The "gracefulStopTimeout" attribute shall be present, when the "changeStateTo" attribute is equal to "STOPPED" and the "stopType" attribute is equal to "GRACEFUL". The "gracefulStopTimeout" attribute shall be absent, when the "changeStateTo" attribute is equal to "STOPPED" and the "stopType" attribute is equal to "FORCEFUL". The request shall be treated as if the "stopType" attribute was set to "FORCEFUL", when the "changeStateTo" attribute is equal to "STOPPED" and the "stopType" attribute is absent.			

### 6.5.3.32 Type: ModifyVnfInfoData

This type represents the information that is requested to be modified for a VNF instance. The information to be modified shall comply with the associated VNFD. This type shall comply with the provisions defined in table 6.5.3.32-1.

**Table 6.5.3.32-1: Definition of the ModifyVnfInfoData data type**

Attribute name	Data type	Cardinality	Description
vnfInstanceId	Identifier	1	Identifier of the VNF instance.
vnfInstanceName	String	0..1	New value of the "vnfInstanceName" attribute in "VnfInstance", or "null" to remove the attribute.
vnfInstanceDescription	String	0..1	New value of the "vnfInstanceDescription" attribute in "VnfInstance", or "null" to remove the attribute.
vnfId	Identifier	0..1	New value of the "vnfId" attribute in "VnfInstance". The value "null" is not permitted.
vnfConfigurableProperties	KeyValuePairs	0..1	Modifications to entries in the "vnfConfigurableProperties" attribute in "VnfInstance", as defined in clause 6.5.3.57. In addition, the provisions in clause 6.7 shall apply.
metadata	KeyValuePairs	0..1	Modifications to entries in the "metadata" attribute in "VnfInstance", as defined in clause 6.5.3.57.
extensions	KeyValuePairs	0..1	Modifications to entries in the "extensions" attribute in "VnfInstance", as defined in clause 6.5.3.57. In addition, the provisions in clause 6.7 shall apply.

### 6.5.3.33 Type: ChangeExtVnfConnectivityData

This type describes the information invoked by the NFVO to change the external VNF connectivity information maintained by the VNFM. The types of changes that this operation supports are:

- 1) Disconnect external CPs that are connected to a particular external VL, and connect them to a different external VL.
- 2) Disconnect and delete external CPs that are connected to a particular external VL and that represent subports in a trunk, i.e. CP instances that are created from external CPDs that have trunk mode configured according to clause 7.1.6.3 in ETSI GS NFV-IFA 011 [29]. If the parent port is exposed as an "extCp", the VNFM shall ensure that the parent port is not deleted. If the parent port is exposed as an "extCp" and there are other subports connected, the VNFM shall ensure that the parent port is not disconnected, unless it is reconnected to a different external VL in the same operation.
- 3) Change the connectivity parameters of existing external CPs, including changing addresses.

NOTE: Depending on the capabilities of the underlying VIM resources, certain changes (e.g. modifying the IP address assignment) might not be supported without deleting the resource and creating another one with the modified configuration.

- 4) Create new CPs that represent subports in a trunk, i.e. CP instances that are created from external CPDs that have trunk mode configured according to clause 7.1.6.3 in ETSI GS NFV-IFA 011 [29], and connect them to a particular external VL. Creation of the parent port with this operation is not supported.

This type shall comply with the provisions defined in table 6.5.3.33-1.

**Table 6.5.3.33-1: Definition of the ChangeExtVnfConnectivityData data type**

Attribute name	Data type	Cardinality	Description
vnfInstanceId	Identifier	1	Identifier of the VNF instance.
extVirtualLinks	ExtVirtualLinkData	1..N	Information about external VLs to change (e.g. connect the VNF to). Entries in the list of external VLs that are unchanged need not be supplied.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the OSS/BSS as input to the external connectivity change process, specific to the VNF instance being changed.

### 6.5.3.34 Type: AssocNewNsdVersionData

This type specifies a new NSD version that is associated to the NS instance. After issuing the Update NS operation with updateType = "AssocNewNsdVersion", the NFVO shall use the referred NSD as a basis for the given NS instance. Different versions of the same NSD have same nsdInvariantId, but different nsdId attributes, therefore if the nsdInvariantId of the NSD version that is to be associated to this NS instance is different from the one used before, the NFVO shall reject the request. Only new versions of the same NSD can be associated to an existing NS instance. This data type shall comply with the provisions defined in table 6.5.3.34-1.

**Table 6.5.3.34-1: Definition of the AssocNewNsdVersionData data type**

Attribute name	Data type	Cardinality	Description
newNsdId	Identifier	1	Identifier of the new NSD version that is to be associated to the NS instance.
sync	Boolean	0..1	Specify whether the NS instance shall be automatically synchronized to the new NSD by the NFVO (in case of true value) or the NFVO shall not do any action (in case of a false value) and wait for further guidance from OSS/BSS (i.e. waiting for OSS/BSS to issue NS lifecycle management operation to explicitly add/remove VNFs and modify information of VNF instances according to the new NSD). The synchronization to the new NSD means e.g. instantiating/adding those VNFs whose VNFD is referenced by the new NSD version but not referenced by the old one, terminating/removing those VNFs whose VNFD is referenced by the old NSD version but not referenced by the new NSD version, modifying information of VNF instances to the new applicable VNFD provided in the new NSD version. See note. A cardinality of 0 indicates that synchronization shall not be done.
NOTE: The replacement of VNF instances, nested NS instances or PNF instances by instances whose descriptor is referenced by the new NSD is applicable even if the instance with a descriptor not referenced from the new NSD has the same VnfdExtInvariantId, NsdExtInvariantId or PnfdExtInvariantId as the new one, i.e. the synchronization procedure is based on descriptor identities of NSD constituents, and does not consider VnfdExtInvariantId, NsdExtInvariantId or PnfdExtInvariantId values.			

### 6.5.3.35 Type: MoveVnflInstanceData

This type specifies existing VNF instances to be moved from one NS instance (source) to another NS instance (destination). The NS instance defined in the Update NS operation indicates the source NS instance and the destination NS instance is specified in this data type (referred to targetNsInstanceId). It shall comply with the provisions defined in table 6.5.3.35-1.

**Table 6.5.3.35-1: Definition of the MoveVnflInstanceData data type**

Attribute name	Data type	Cardinality	Description
targetNsInstanceId	Identifier	1	Specify the target NS instance where the VNF instances are moved to.
vnflInstanceId	Identifier	1..N	Specify the VNF instance that is moved.

### 6.5.3.36 Type: AddVnffgData

This type specifies the parameters used for the creation of a new VNFFG instance. It shall comply with the provisions defined in table 6.5.3.36-1.

**Table 6.5.3.36-1: Definition of the AddVnffgData data type**

Attribute name	Data type	Cardinality	Description
vnffgId	IdentifierInNsd	1	Identifier of the VNFFGD used to create this VNFFG instance.
vnffgName	String	1	Human readable name for the VNFFG.
description	String	1	Human readable description for the VNFFG.

### 6.5.3.37 Type: UpdateVnffgData

This type specifies the parameters used for the update of an existing VNFFG instance. It shall comply with the provisions defined in table 6.5.3.37-1.

**Table 6.5.3.37-1: Definition of the UpdateVnffgData data type**

Attribute name	Data type	Cardinality	Description
vnffgInfold	IdentifierInNs	1	Identifier of an existing VNFFG to be updated for the NS Instance.
nfp	NfpData	0..N	Indicate the desired new NFP(s) for a given VNFFG after the operations of addition/removal of NS components (e.g. VNFs, VLs, etc.) have been completed, or indicate the updated or newly created NFP classification and selection rule which applied to an existing NFP.
nfpInfold	IdentifierInNs	0..N	Identifier(s) of the NFP to be deleted from a given VNFFG.

### 6.5.3.38 Type: NfpData

This type contains information used to create or modify NFP instance parameters for the update of an existing VNFFG instance. It shall comply with the provisions defined in table 6.5.3.38-1.

**Table 6.5.3.38-1: Definition of the NfpData data type**

Attribute name	Data type	Cardinality	Description
nfpInfold	IdentifierInNs	0..1	Identifier of the NFP to be modified. It shall be present for modified NFPs and shall be absent for the new NFP. See note 1.
nfpName	String	0..1	Human readable name for the NFP. It shall be present for the new NFP, and it may be present otherwise. See note 2.
description	String	0..1	Human readable description for the NFP. It shall be present for the new NFP, and it may be present otherwise. See note 2.
cpGroup	CpGroupInfo	0..N	Group(s) of CPs and/or SAPs which the NFP passes by. Cardinality can be 0 if only updated or newly created NFP classification and selection rule which applied to an existing NFP is provided. See note 3 and note 4.
nfpRule	NfpRule	0..1	NFP classification and selection rule. See note 1.
NOTE 1: It shall be present for modified NFPs and shall be absent for the new NFP.			
NOTE 2: It shall be present for the new NFP, and it may be present otherwise.			
NOTE 3: At least a CP or an nfpRule shall be present.			
NOTE 4: When multiple identifiers are included, the position of the identifier in the cpGroup value specifies the position of the group in the path.			

### 6.5.3.39 Type: ChangeNsFlavourData

This type specifies an existing NS instance for which the DF needs to be changed. This specifies the new DF, the instantiationLevel of the new DF that may be used. It shall comply with the provisions defined in table 6.5.3.39-1.

**Table 6.5.3.39-1: Definition of the ChangeNsFlavourData data type**

Attribute name	Data type	Cardinality	Description
newNsFlavourId	IdentifierInNsd	1	Identifier of the new NS DF to apply to this NS instance.
instantiationLevelId	IdentifierInNsd	0..1	Identifier of the instantiation level of the deployment flavour to be instantiated. If not present, the default instantiation level as declared in the NSD is instantiated.

### 6.5.3.40 Type: NfpRule

The NfpRule data type is an expression of the conditions that shall be met in order for the NFP to be applicable to the packet. The condition acts as a flow classifier and it is met only if all the values expressed in the condition are matched by those in the packet. It shall comply with the provisions defined in table 6.5.3.40-1.

**Table 6.5.3.40-1: Definition of the NfpRule data type**

Attribute name	Data type	Cardinality	Description
etherDestinationAddress	MacAddress	0..1	Indicates a destination Mac address See note.
etherSourceAddress	MacAddress	0..1	Indicates a source Mac address See note.
etherType	Enum (inlined)	0..1	Indicates the protocol carried over the Ethernet layer.  Permitted values - IPV4 - IPV6 See note.
vlanTag	String	0..N	Indicates a VLAN identifier in an IEEE 802.1Q™-2018 tag [6] Multiple tags can be included for QinQ stacking. See note.
protocol	Enum (inlined)	0..1	Indicates the L4 protocol, For IPv4 [7] this corresponds to the field called "Protocol" to identify the next level protocol. For IPv6 [12] this corresponds to the field is called the "Next Header" field.  Permitted values: Any keyword defined in the IANA protocol registry [1], e.g.: - TCP - UDP - ICMP See note.
dscp	String	0..1	For IPv4 [7] a string of "0" and "1" digits that corresponds to the 6-bit Differentiated Services Code Point (DSCP) field of the IP header. For IPv6 [12] a string of "0" and "1" digits that corresponds to the 6 differentiated services bits of the traffic class header field. See note.
sourcePortRange	PortRange	0..1	Indicates a range of source ports. See note.
destinationPortRange	PortRange	0..1	Indicates a range of destination ports. See note.
sourceIpAddressPrefix	IpAddressPrefix	0..1	Indicates the source IP address range in CIDR format. See note.
destinationIpAddressPrefix	IpAddressPrefix	0..1	Indicates the destination IP address range in CIDR format. See note.
extendedCriteria	Mask	0..N	Indicates values of specific bits in a frame. See note.
NOTE: At least one attribute shall be present. If multiple attributes are present, a logical "AND" operation shall be applied to those attributes when matching packets against the rule.			

#### 6.5.3.41 Type: Mask

The Mask data type identifies the value to be matched for a sequence of bits at a particular location in a frame. It shall comply with the provisions defined in table 6.5.3.41-1.

**Table 6.5.3.41-1: Definition of the Mask data type**

Attribute name	Data type	Cardinality	Description
startingPoint	Integer	1	Indicates the offset between the last bit of the source mac address and the first bit of the sequence of bits to be matched.
length	Integer	1	Indicates the number of bits to be matched.
value	String	1	Provide the sequence of bit values to be matched.

### 6.5.3.42 Type: PortRange

The PortRange data type provides the lower and upper bounds of a range of Internet ports. It shall comply with the provisions defined in table 6.5.3.42-1.

**Table 6.5.3.42-1: Definition of the PortRange data type**

Attribute name	Data type	Cardinality	Description
lowerPort	Integer	1	Identifies the lower bound of the port range.
upperPort	Integer	1	Identifies the upper bound of the port range.

### 6.5.3.43 Type: HealNsData

This type represents the information used to heal an NS. It shall comply with the provisions defined in table 6.5.3.43-1.

**Table 6.5.3.43-1: Definition of the HealNsData data type**

Attribute name	Data type	Cardinality	Description
degreeHealing	Enum (inlined)	1	Indicates the degree of healing. Possible values include: <ul style="list-style-type: none"> <li>- HEAL_RESTORE: Complete the healing of the NS restoring the state of the NS before the failure occurred</li> <li>- HEAL_QOS: Complete the healing of the NS based on the newest QoS values</li> <li>- HEAL_RESET: Complete the healing of the NS resetting to the original instantiation state of the NS</li> <li>- PARTIAL_HEALING</li> </ul>
actionsHealing	String	0..N	Used to specify dedicated healing actions in a particular order (e.g. as a script). The actionsHealing attribute can be used to provide a specific script whose content and actions might only be possible to be derived during runtime. See note.
healScript	IdentifierInNsd	0..1	Reference to a script from the NSD that shall be used to execute dedicated healing actions in a particular order. The healScript, since it refers to a script in the NSD, can be used to execute healing actions which are defined during NS design time. See note.
additionalParamsforNs	KeyValuePairs	0..1	Allows the OSS/BSS to provide additional parameter(s) to the healing process at the NS level.
NOTE: Either the actionsHealing or healScript attribute shall be present, not both attributes.			

### 6.5.3.44 Type: HealVnfData

This type represents the information to heal a VNF that is part of an NS. The NFVO shall then invoke the Heal VNF operation towards the appropriate VNFM. It shall comply with the provisions defined in table 6.5.3.44-1.

**Table 6.5.3.44-1: Definition of the HealVnfData data type**

Attribute name	Data type	Cardinality	Description
vnfInstanceid	Identifier	1	Identifies the VNF instance, part of the NS, requiring a healing action.
cause	String	0..1	Indicates the reason why a healing procedure is required.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the NFVO as input to the healing process, specific to the VNF being healed.  EXAMPLE: Input parameters to VNF-specific healing procedures.

### 6.5.3.45 Type: ScaleNsData

This type represents the information to scale an NS. It shall comply with the provisions defined in table 6.5.3.45-1.

**Table 6.5.3.45-1: Definition of the ScaleNsData data type**

Attribute name	Data type	Cardinality	Description
vnfInstanceToBeAdded	VnfInstanceData	0..N	An existing VNF instance to be added to the NS instance as part of the scaling operation. If needed, the VNF Profile to be used for this VNF instance may also be provided. See note 1, note 2 and note 3.
vnfInstanceToBeRemoved	Identifier	0..N	The VNF instance to be removed from the NS instance as part of the scaling operation. See note 1 and note 4.
scaleNsByStepsData	ScaleNsByStepsData	0..1	The information used to scale an NS instance by one or more scaling steps. See note 1.
scaleNsToLevelData	ScaleNsToLevelData	0..1	The information used to scale an NS instance to a target size. See note 1.
additionalParamsForNs	KeyValuePairs	0..1	Allows the OSS/BSS to provide additional parameter(s) at the NS level necessary for the NS scaling (as opposed to the VNF level, which is covered in additionalParamsForVnf).
additionalParamsForVnf	ParamsForVnf	0..N	Allows the OSS/BSS to provide additional parameter(s) per VNF instance (as opposed to the NS level, which is covered in additionalParamsforNs). This is for VNFs that are to be created by the NFVO as part of the NS scaling and not for existing VNF that are covered by the scaleVnfData.
locationConstraints	VnfLocationConstraint	0..N	The location constraints for the VNF to be instantiated as part of the NS scaling. An example can be a constraint for the VNF to be in a specific geographic location.
nestedNsLocationConstraints	NestedNsLocationConstraints	0..N	Defines the location constraints for the nested NS to be instantiated as part of the NS instantiation. An example can be a constraint for the nested NS to be in a specific geographic location.
<p>NOTE 1: No more than two attributes between vnfInstanceToBeAdded, vnfInstanceToBeRemoved, scaleNsByStepsData and scaleNsToLevelData shall be present. In case of two, the attributes shall be vnfInstanceToBeAdded and vnfInstanceToBeRemoved.</p> <p>NOTE 2: The DF of the VNF instance shall match the VNF DF present in the associated VNF Profile of the new NS flavour.</p> <p>NOTE 3: This functionality is the same as the one provided by the Update NS operation when the AddVnf update type is selected (see clause 7.3.5).</p> <p>NOTE 4: This functionality is the same as the one provided by the Update NS operation when the RemoveVnf update type is selected (see clause 7.3.5).</p>			

### 6.5.3.46 Type: ScaleNsByStepsData

This type represents the information used to scale an NS instance by one or more scaling steps, with respect to a particular NS scaling aspect. Performing a scaling step means increasing/decreasing the capacity of an NS instance in a discrete manner, i.e. moving from one NS scale level to another. The NS scaling aspects and their corresponding NS scale levels applicable to the NS instance are declared in the NSD. It shall comply with the provisions defined in table 6.5.3.46-1.

**Table 6.5.3.46-1: Definition of the ScaleNsByStepsData data type**

Attribute name	Data type	Cardinality	Description
scalingDirection	Enum (inlined)	1	The scaling direction. Possible values are: - SCALE_IN - SCALE_OUT
aspectId	IdentifierInNsd	1	The aspect of the NS that is requested to be scaled, as declared in the NSD.
numberOfSteps	Integer	0..1	The number of scaling steps to be performed. Defaults to 1.

#### 6.5.3.47 Type: ScaleNsToLevelData

This type represents the information used to scale an NS instance to a target size. The target size is either expressed as an NS instantiation level or as a list of NS scale levels, one per NS scaling aspect, of the current DF. The NS instantiation levels, the NS scaling aspects and their corresponding NS scale levels applicable to the NS instance are declared in the NSD. It shall comply with the provisions defined in table 6.5.3.47-1.

**Table 6.5.3.47-1: Definition of the ScaleNsToLevelData data type**

Attribute name	Data type	Cardinality	Description
nsInstantiationLevel	IdentifierInNsd	0..1	Identifier of the target NS instantiation level of the current DF to which the NS instance is requested to be scaled. See note.
nsScaleInfo	NsScaleInfo	0..N	For each NS scaling aspect of the current DF, defines the target NS scale level to which the NS instance is to be scaled. See note.

NOTE: Either nsInstantiationLevel or nsScaleInfo, but not both, shall be present.

#### 6.5.3.48 Type: NsScaleInfo

This type represents the target NS Scale level for each NS scaling aspect of the current deployment flavour. It shall comply with the provisions defined in table 6.5.3.48-1.

**Table 6.5.3.48-1: Definition of the NsScaleInfo data type**

Attribute name	Data type	Cardinality	Description
nsScalingAspectId	IdentifierInNsd	1	Identifier of the NS scaling aspect.
nsScaleLevelId	IdentifierInNsd	1	Identifier of the NS scale level.

#### 6.5.3.49 Type: ScaleVnfData

This type represents defines the information to scale a VNF instance to a given level, or to scale a VNF instance by steps. It shall comply with the provisions defined in table 6.5.3.49-1.



**Table 6.5.3.49-1: Definition of the ScaleVnfData data type**

Attribute name	Data type	Cardinality	Description
vnfInstanceId	Identifier	1	Identifier of the VNF instance being scaled.
scaleVnfType	Enum (inlined)	1	Type of the scale VNF operation requested. Allowed values are: <ul style="list-style-type: none"> <li>- SCALE_OUT</li> <li>- SCALE_IN</li> <li>- SCALE_TO_INSTANTIATION_LEVEL</li> <li>- SCALE_TO_SCALE_LEVEL(S)</li> </ul> The set of types actually supported depends on the capabilities of the VNF being managed. See note 1.
scaleToLevelData	ScaleToLevelData	0..1	The information used for scaling to a given level. See note 2.
scaleByStepData	ScaleByStepData	0..1	The information used for scaling by steps. See note 2.
NOTE 1: ETSI GS NFV-IFA 010 [2] specifies that the lifecycle management operations that expand or contract a VNF instance include scale in, scale out, scale up and scale down. Vertical scaling (scale up, scale down) is not supported in the present document.			
NOTE 2: Either scaleToLevelData or scaleByStepData but not both shall be present. The scaleByStepData is used for scale out/in type of scaling, and the scaleToLevelData is used for scale to instantiation/scale level type of scaling.			

### 6.5.3.50 Type: ScaleToLevelData

This type describes the information used to scale a VNF instance to a target size. The target size is either expressed as an instantiation level of that DF as defined in the VNFD, or given as a list of scale levels, one per scaling aspect of that DF. Instantiation levels and scaling aspects are declared in the VNFD. The NFVO shall then invoke the ScaleVnfToLevel operation towards the appropriate VNFM. It shall comply with the provisions defined in table 6.5.3.50-1.

**Table 6.5.3.50-1: Definition of the ScaleToLevelData data type**

Attribute name	Data type	Cardinality	Description
vnfInstantiationLevelId	IdentifierInVnfd	0..1	Identifier of the target instantiation level of the current deployment flavour to which the VNF is requested to be scaled. See note.
vnfScaleInfo	VnfScaleInfo	0..N	For each scaling aspect of the current deployment flavour, indicates the target scale level to which the VNF is to be scaled. See note.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the NFVO as input to the scaling process, specific to the VNF being scaled.
NOTE: Either the vnfInstantiationLevelId attribute or the vnfScaleInfo attribute shall be included.			

### 6.5.3.51 Type: VnfScaleInfo

This type describes the provides information about the scale level of a VNF instance with respect to one scaling aspect. It shall comply with the provisions defined in table 6.5.3.51-1.

**Table 6.5.3.51-1: Definition of the VnfScaleInfo data type**

Attribute name	Data type	Cardinality	Description
aspectId	IdentifierInVnfd	1	The scaling aspect.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure).
scaleLevel	Integer	1	The scale level for that aspect. Minimum value 0, maximum value maxScaleLevel as declared in the VNFD.

### 6.5.3.52 Type: ScaleByStepData

This type describes the information to scale a VNF instance by steps. The NFVO shall then invoke the Scale VNF operation towards the appropriate VNFM. It shall comply with the provisions defined in table 6.5.3.52-1.

**Table 6.5.3.52-1: Definition of the ScaleByStepData data type**

Attribute name	Data type	Cardinality	Description
aspectId	IdentifierInVnfd	1	Identifier of (reference to) the aspect of the VNF that is requested to be scaled, as declared in the VNFD.
numberOfSteps	Integer	0..1	Number of scaling steps. It shall be a positive number. Defaults to 1.  The VNF provider defines in the VNFD whether or not a particular VNF supports performing more than one step at a time. Such a property in the VNFD applies for all instances of a particular VNF. See note.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the NFVO as input to the scaling process, specific to the VNF instance being scaled.

NOTE: A scaling step is the smallest unit by which a VNF instance can be scaled w.r.t a particular scaling aspect.

### 6.5.3.53 Type: NsVirtualLinkInfo

This type specifies the information about an NS VL instance. It shall comply with the provisions defined in table 6.5.3.53-1.

**Table 6.5.3.53-1: Definition of the NsVirtualLinkInfo data type**

Attribute name	Data type	Cardinality	Description
id	IdentifierInNs	1	Identifier of the VL instance.
nsVirtualLinkDesclId	IdentifierInNsd	1	Identifier of the VLD in the NSD.
nsVirtualLinkProfileId	IdentifierInNsd	1	Identifier of the VL profile in the NSD.
resourceHandle	ResourceHandle	1..N	Identifier(s) of the virtualised network resource(s) and/or multi-site connectivity service(s) realizing the VL instance. See note.
linkPort	NsLinkPortInfo	0..N	Link ports of the VL instance. Cardinality of zero indicates that no port has yet been created for the VL instance.

NOTE: As an NS can include NFs deployed in NFVI PoPs under the control of several different VIMs, therefore deploying an NS VL can involve several VIMs, each allocating different virtualised network resources, as well as WIMs handling the connectivity in between the NFVI-PoPs in the form of multi-site connectivity services. When this NsVirtualLink is provided as an ExtVirtualLink as input of a VNF LCM operation, the id of the ExtVirtualLink shall be the same as the corresponding NsVirtualLink.

### 6.5.3.54 Type: ChangeVnfPackageData

This type specifies the information needed to change the current VNF package for a VNF instance. Clause B.3 of the ETSI GS NFV-IFA 007 [19] illustrates the variants of changes to the current VNF Package and information flow procedures.

This operation encompasses the following scenarios:

- Changes of the VNF virtualised resources, such as requirements, composition and structure between the VNF versions, without changing the VNF software version.
- Changes of both the VNF software version and the VNF virtualised resources. This case includes replacing the VNF software version by means of virtualised resources management, such as terminating the current virtualised resource instances running the current software version and instantiating new virtualised resource instances with the destination VNF software version. The new virtualised resource instances may have the same characteristics as the current virtualised resource instances.

- Changes related to the VNFD, such as correction of bugs in the VNFD, changes in the naming scheme of VNFD components (e.g. name of the VDU, vduId), and adding/removing descriptors of VNF Package changes (VnfPackageChangeInfo).

**NOTE:** For software updates that are executed by functional entities outside NFV-MANO and that require synchronization of the information held by the NFV-MANO entities with a new VNF package that reflects the same changes, an alternative procedure using the PATCH method on the "Individual VNF instance" resource has been defined, as illustrated in clause B.2 of ETSI GS NFV-IFA 007 [19]. This procedure assumes certain restrictions on the characteristics of the new VNF package, as defined in note 1 in table 5.5.2.2-1 of ETSI GS NFV-SOL 003 [4].

This type shall comply with the provisions defined in table 6.5.3.54-1.

**Table 6.5.3.54-1: Definition of the ChangeVnfPackageData data type**

Attribute name	Data type	Cardinality	Description
vnfInstanceId	Identifier	1	Identifier of the VNF instance.
vnfdId	Identifier	1	Identifier of the VNFD which defines the destination VNF Package for the change.
extVirtualLinks	ExtVirtualLinkData	0..N	Information about external VLs to connect the VNF to. Entries in the list that are unchanged need not be supplied as part of this request.
extManagedVirtualLinks	ExtManagedVirtualLinkData	0..N	Information about internal VLs that are managed by the NFVO.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the OSS/BSS as input to the change current VNF package process, specific to the VNF of which the underlying VNF package is changed, as declared in the VNFD as part of "ChangeCurrentVnfPkgOpConfig" defined in ETSI GS NFV-IFA 011 [29].
extensions	KeyValuePairs	0..1	If present, this attribute provides modifications to the values of the "extensions" attribute in "VnfInstance", as defined in clause 6.5.3.57. Provisions for handling extensions during the operation, and needed passed parameter values in case of conflicts, are defined in clause 5.4.11a.3.1 of ETSI GS NFV-SOL 003 [4].
vnfConfigurableProperties	KeyValuePairs	0..1	If present, this attribute provides modifications to the values of the VNF configurable properties attribute in the "VnfInstance", as defined in clause 6.5.3.57. Provisions for handling VNF configurable properties during the operation, and needed passed parameter values in case of conflicts, are defined in clause 5.4.11a.3.1 of ETSI GS NFV-SOL 003 [4].
overridingVersionDependency	OverridingVersionDependency	0..N	<p>If present, information in each overridingVersionDependency replaces the versionDependency in the VnfProfile, NsProfile or PnfProfile indicated in the OverridingVersionDependency.</p> <p>Only the versionDependency in the VnfProfile, NsProfile or PnfProfile with the same dependentConstituentId as in the overridingVersionDependency is replaced. See note.</p> <p>If no versionDependency with the dependentConstituentId indicated in the overridingVersionDependency exists in the VnfProfile, NsProfile or PnfProfile, the new versionDependency is added to the runtime information that the NFVO keeps about the profile.</p>
<b>NOTE:</b>	A VnfProfile, NsProfile or PnfProfile may contain multiple VersionDependencies as it may describe the version dependencies of the descriptor referenced in the profile or of other descriptors with the same external invariant identifier.		

### 6.5.3.55 Type: NsLinkPortInfo

This type represents information about a link port of a VL instance. It shall comply with the provisions defined in table 6.5.3.55-1.

**Table 6.5.3.55-1: Definition of the NsLinkPortInfo data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this link port as provided by the entity that has created the link port.
resourceHandle	ResourceHandle	1	Identifier of the virtualised resource realizing this link port.
nsCpHandle	NsCpHandle	0..1	Identifier of the CP/SAP instance to be connected to this link port. The value refers to a vnfExtCpInfo item in the VnfInstance, or a pnfExtCpInfo item in the PnfInfo, or a sapInfo item in the NS instance. There shall be at most one link port associated with any connection point instance.
NOTE: When the NsVirtualLink, from which the present NsLinkPort is part of, is provided as an ExtVirtualLink as input of a VNF LCM operation, the id of the ExtLinkPort shall be the same as the corresponding NsLinkPort.			

### 6.5.3.56 Type: NsCpHandle

This type represents an identifier of the CP or SAP instance. It shall comply with the provisions defined in table 6.5.3.56-1.

**Table 6.5.3.56-1: Definition of the NsCpHandle data type**

Attribute name	Data type	Cardinality	Description
vnfInstancelid	Identifier	0..1	Identifier of the VNF instance associated to the CP instance. This attribute shall be present if the CP instance is VNF external CP. See note 1 and note 4.
vnfExtCpInstancelid	IdentifierInVnf	0..1	Identifier of the VNF external CP instance in the scope of the VNF instance. This attribute shall be present if the CP instance is VNF external CP. See note 1 and note 4.
pnfInfoId	Identifier	0..1	Identifier of the PNF instance associated to the CP instance. This attribute shall be present if the CP instance is PNF external CP. See note 2 and note 4.
pnfExtCpInstancelid	IdentifierInPnf	0..1	Identifier of the PNF external CP instance in the scope of the PNF. This attribute shall be present if the CP instance is PNF external CP. See note 2 and note 4.
nsInstancelid	Identifier	0..1	Identifier of the NS instance associated to the SAP instance. This attribute shall be present if the CP instance is NS SAP. See note 3 and note 4.
nsSapInstancelid	IdentifierInNs	0..1	Identifier of the SAP instance in the scope of the NS instance. This attribute shall be present if the CP instance is NS SAP. See note 3 and note 4.
NOTE 1: For the VNF external CP instance, both vnfInstancelid and vnfExtCpInstancelid shall be present as a pair. NOTE 2: For the PNF external CP instance, both pnfInfoId and PnfExtCpInstancelid shall be present as a pair. NOTE 3: For the SAP instance, both nsInstancelid and nsSapInstancelid shall be present as a pair. NOTE 4: One pair of identifiers (VNF external CP, PNF external CP or SAP) shall be present.			

### 6.5.3.57 Type: VnfInstance

This type represents a VNF instance. It shall comply with the provisions defined in table 6.5.3.57-1.

NOTE: Clause B.3.2 of ETSI GS NFV-SOL 003 [4] provides examples illustrating the relationship among the different run-time information elements (CP, VL and link ports) used to represent the connectivity of a VNF.

Table 6.5.3.57-1: Definition of the VnfInstance data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the VNF instance.
vnfInstanceName	String	0..1	Name of the VNF instance.  Modifications to this attribute can be requested using the "ModifyVnfInfoData" structure.
vnfInstanceDescription	String	0..1	Human-readable description of the VNF instance.  Modifications to this attribute can be requested using the "ModifyVnfInfoData" structure.
vnfdId	Identifier	1	Identifier of the VNFD on which the VNF instance is based.  Modifications to this attribute can be requested using the "ModifyVnfInfoData" structure. See note 1.
vnfProvider	String	1	Provider of the VNF and the VNFD. The value is copied from the VNFD.
vnfProductName	String	1	Name to identify the VNF Product. The value is copied from the VNFD.
vnfSoftwareVersion	Version	1	Software version of the VNF. The value is copied from the VNFD.
vnfdVersion	Version	1	Identifies the version of the VNFD. The value is copied from the VNFD.
versionDependency	VersionDependency	0..N	Describes version dependencies currently valid for the VNF instance. Identifies versions of descriptors of other constituents in the NSD upon which the VNF depends. The dependencies may be described for the VNFD referenced in this VnfInfo with vnfdId and for VNFDs with the same VnfdExtInvariantId.  There shall not be more than one versionDependency present with the same dependentConstituentId.
vnfPkgId	Identifier	1	Identifier of information held by the NFVO about the specific VNF package on which the VNF is based. This identifier has been allocated by the NFVO.
vnfConfigurableProperties	KeyValuePairs	0..1	Additional VNF-specific attributes that provide the current values of the configurable properties of the VNF instance.  These attributes represent values that are stored persistently in the VnfInstance structure and that correspond to configuration parameters of the VNF instance.  Modifying these attributes affects the configuration of the VNF instance either directly (if the VNF instance is in INSTANTIATED state at the time of the modification) or as part of the subsequent VNF instantiation operation (if the VNF instance is in NOT_INSTANTIATED state at the time of the modification).  Configurable properties referred in these attributes are declared in the VNFD (see note 2 and note 3).

Attribute name	Data type	Cardinality	Description
			<p>These configurable properties include the following standard attributes, which are declared in the VNFD if auto-scaling and/or auto-healing are supported by the VNF:</p> <ul style="list-style-type: none"> <li>- isAutoscaleEnabled: If present, the VNF supports auto-scaling. If set to true, auto-scaling is currently enabled. If set to false, auto-scaling is currently disabled.</li> <li>- isAutohealEnabled: If present, the VNF supports auto-healing. If set to true, auto-healing is currently enabled. If set to false, auto-healing is currently disabled.</li> </ul> <p>Modifications to these attributes can be requested using the "ModifyVnfInfoData" structure.</p> <p>In addition, the provisions in clause 6.7 shall apply.</p>
vimId	Identifier	0..N	Identifier of a VIM that manages resources for the VNF instance.
instantiationState	Enum (inlined)	1	<p>The instantiation state of the VNF.</p> <p>Permitted values:</p> <ul style="list-style-type: none"> <li>- NOT_INSTANTIATED: The VNF instance is terminated or not instantiated.</li> <li>- INSTANTIATED: The VNF instance is instantiated.</li> </ul>
instantiatedVnfInfo	Structure (inlined)	0..1	<p>Information specific to an instantiated VNF instance.</p> <p>This attribute shall be present if the instantiationState attribute value is INSTANTIATED.</p>
>flavourId	IdentifierInVnfd	1	Identifier of the VNF deployment flavour applied to this VNF instance.
>vnfState	VnfOperationalStateType	1	State of the VNF instance.
>scaleStatus	VnfScaleInfo	0..N	<p>Scale status of the VNF, one entry per aspect. Represents for every scaling aspect how "big" the VNF has been scaled w.r.t. that aspect. This attribute shall be present if the VNF supports scaling.</p> <p>See clause B.2 of ETSI GS NFV-SOL 003 [4] for an explanation of VNF scaling.</p>
>maxScaleLevels	VnfScaleInfo	0..N	<p>Maximum allowed scale levels of the VNF, one entry per aspect.</p> <p>This attribute shall be present if the VNF supports scaling.</p>
>extCplInfo	VnfExtCplInfo	1..N	<p>Information about the external CPs exposed by the VNF instance.</p> <p>When trunking is enabled, the list of entries includes both, external CPs corresponding to parent ports of a trunk, and external CPs associated to sub-ports of a trunk.</p>
>vipCplInfo	VipCplInfo	0..N	<p>VIP CPs that are part of the VNF instance. Shall be present when that particular VIP CP of the VNFC instance is associated to an external CP of the VNF instance.</p> <p>May be present otherwise.</p>

Attribute name	Data type	Cardinality	Description
>virtualCpInfo	VirtualCpInfo	0..N	virtual CPs that are part of the VNF instance. Shall be present when a particular virtual CP is associated to an external CP of the VNF instance.  May be present otherwise.
>extVirtualLinkInfo	ExtVirtualLinkInfo	0..N	Information about the external VLs the VNF instance is connected to.
>extManagedVirtualLinkInfo	ExtManagedVirtualLinkInfo	0..N	Information about the externally-managed internal VLs of the VNF instance. See note 4 and note 5.
>monitoringParameters	VnfMonitoringParameter	0..N	Performance metrics tracked by the VNFM (e.g. for auto-scaling purposes) as identified by the VNF provider in the VNFD.
>localizationLanguage	String	0..1	Information about localization language of the VNF (includes e.g. strings in the VNFD).  The localization languages supported by a VNF can be declared in the VNFD, and localization language selection can take place at instantiation time.  The value shall comply with the format defined in IETF RFC 5646 [9].
>vnfcResourceInfo	VnfcResourceInfo	0..N	Information about the virtualised compute and storage resources used by the VNFCs of the VNF instance.
>vnfVirtualLinkResourceInfo	VnfVirtualLinkResourceInfo	0..N	Information about the virtualised network resources used by the VLs of the VNF instance. See note 5.
>virtualStorageResourceInfo	VirtualStorageResourceInfo	0..N	Information about the virtualised storage resources used as storage for the VNF instance.
>mciInfo	MciInfo	0..N	Information on the MCIO(s) representing VNFC instance(s) realized by one or a set of OS containers and created from the same VDU for the VNF instance.
metadata	KeyValuePairs	0..1	Additional VNF-specific attributes that provide metadata describing the VNF instance.  These attributes represent values that are stored persistently in the VnfInstance structure for consumption by functional blocks that invoke the VNF lifecycle management interface. They are not consumed by the VNFM, or the lifecycle management scripts.  Modifying the values of these attributes has no effect on the VNF instance, it only affects the information represented in the VnfInstance structure.  Metadata that the VNF provider foresees are expected to be declared in the VNFD (see note 2).  Modifications to these attributes can be requested using the "ModifyVnfInfoData" structure.
extensions	KeyValuePairs	0..1	Additional VNF-specific attributes that affect the lifecycle management of this VNF instance. These attributes represent values that are stored persistently in the VnfInstance structure for consumption by the VNFM or by the lifecycle management scripts during the execution of VNF lifecycle management operations.

Attribute name	Data type	Cardinality	Description
			<p>Modifying the values of these attributes has no direct effect on the VNF instance; however, the modified attribute values can be considered during subsequent VNF lifecycle management operations, which means that the modified values can indirectly affect the configuration of the VNF instance.</p> <p>All extensions that are allowed for the VNF are declared in the VNFD.</p> <p>Modifications to these attributes can be requested using the "ModifyVnfInfoData" structure.</p> <p>In addition, the provisions in clause 6.7 shall apply.</p>
<p>NOTE 1: Modifying the value of this attribute shall not be performed when conflicts exist between the previous and the newly referred VNF package, i.e. when the new VNFD is changed with respect to the previous VNFD in other aspects than merely referencing to other VNF software images. In order to avoid misalignment of the VnfInstance with the current VNF's on-boarded VNF Package, the values of attributes in the VnfInstance that have corresponding attributes in the VNFD shall be kept in sync with the values in the VNFD.</p> <p>NOTE 2: ETSI GS NFV-SOL 001 [i.1] specifies the structure and format of the VNFD based on TOSCA specifications.</p> <p>NOTE 3: VNF configurable properties are sometimes also referred to as configuration parameters applicable to a VNF. Some of these are set prior to instantiation and cannot be modified if the VNF is instantiated, some are set prior to instantiation (are part of initial configuration) and can be modified later, and others can be set only after instantiation. The applicability of certain configuration may depend on the VNF and the required operation of the VNF at a certain point in time.</p> <p>NOTE 4: It is possible to have several ExtManagedVirtualLinkInfo for the same VNF internal VL in case of a multi-site VNF spanning several VIMs. The set of ExtManagedVirtualLinkInfo corresponding to the same VNF internal VL shall indicate so by referencing to the same VnfVirtualLinkDesc and externally-managed multi-site VL instance (refer to clause 6.5.3.59).</p> <p>NOTE 5: Even though externally-managed internal VLs are also used for VNF-internal connectivity, they shall not be listed in the "vnfVirtualLinkResourceInfo" attribute as this would be redundant.</p>			

### 6.5.3.58 Type: CpProtocollInfo

This type describes the protocol layer(s) that a CP or SAP uses together with protocol-related information, like addresses. It shall comply with the provisions defined in table 6.5.3.58-1.

**Table 6.5.3.58-1: Definition of the CpProtocollInfo data type**

Attribute name	Data type	Cardinality	Description
layerProtocol	Enum (inlined)	1	<p>The identifier of layer(s) and protocol(s) associated to the network address information.</p> <p>Permitted values:</p> <ul style="list-style-type: none"> <li>- IP_OVER_ETHERNET</li> <li>- IP_FOR_VIRTUAL_CP</li> </ul> <p>See note.</p>
ipOverEthernet	IpOverEthernetAddressInfo	0..1	IP addresses over Ethernet to assign to the CP or SAP instance. Shall be present if layerProtocol is equal to "IP_OVER_ETHERNET", and shall be absent otherwise.
virtualCpAddress	VirtualCpAddressInfo	0..1	IP address data assigned to an external CP instance exposing a virtual CP. It shall be present if layerProtocol is equal to "IP_FOR_VIRTUAL_CP" and the external CP instance exposes a virtual CP and shall not be present otherwise.
<p>NOTE: This attribute allows to signal the addition of further types of layer and protocol in future versions of the present document in a backwards-compatible way. In the current version of the present document, only IP over Ethernet is supported.</p>			



### 6.5.3.59 Type: ExtManagedVirtualLinkInfo

This type provides information about an externally-managed internal virtual link for VNFs. It shall comply with the provisions defined in table 6.5.3.59-1.

**Table 6.5.3.59-1: Definition of the ExtManagedVirtualLinkInfo data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the externally-managed internal VL and the related externally-managed VL information instance.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure).
vnfVirtualLinkDescId	IdentifierInVnfd	1	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.
networkResource	ResourceHandle	1	Reference to the VirtualNetwork resource or multi-site connectivity service providing this VL.
vnfLinkPorts	VnfLinkPortInfo	0..N	Link ports of this VL.
vnfNetAttDefResource	NetAttDefResourceInfo	0..N	Network attachment definition resources that provide the specification of the interface to attach connection points to this VL.
extManagedMultisiteVirtualLinkId	Identifier	0..1	Identifier of the externally-managed multi-site VL instance. The identifier is assigned by the NFV-MANO entity that manages the externally managed multi-site VL instance. It shall be present when the present externally-managed internal VL is part of a multi-site VL, e.g. in support of multi-site VNF spanning several VIMs. All externally-managed internal VL instances corresponding to an internal VL created based on the same virtualLinkDescId shall refer to the same extManagedMultisiteVirtualLinkId.

### 6.5.3.60 Type: VnfcResourceInfo

This type represents the information on virtualised compute and storage resources used by a VNFC in a VNF instance. It shall comply with the provisions defined in table 6.5.3.60-1.

Depending on the form of virtualisation container of the VNFC:

- for a VNFC based on VM, a reference to the corresponding VirtualCompute shall be provided, and
- for a VNFC based on OS container(s), a reference to the Compute MCIO shall be provided. Hence, exposure of information by the NFVO to the OSS/BSS is at the MCIO level.

In addition, the references to the storage resources depend on the form of the VNFC:

- for a VNFC based on VM, storage resource identifiers shall refer to VirtualStorage resources, and
- for a VNFC based on OS container(s), storage resource identifiers shall refer to Storage MCIOs.

**Table 6.5.3.60-1: Definition of the VnfcResourceInfo data type**

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of this VnfcResourceInfo instance.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure).
vduld	IdentifierInVnfd	1	Reference to the applicable VDU in the VNFD. See note 1.

Attribute name	Data type	Cardinality	Description
computeResource	ResourceHandle	1	Reference to the VirtualCompute resource or reference to a Compute MCIO.
storageResourceIds	IdentifierInVnf	0..N	References to the VirtualStorage resources or references to Storage MCIOs. The value refers to a VirtualStorageResourceInfo item in the VnfInstance.
reservationId	Identifier	0..1	The reservation identifier applicable to the resource. It shall be present when an applicable reservation exists.
vnfcCpInfo	Structure (inlined)	0..N	CPs of the VNFC instance. Shall be present when that particular CP of the VNFC instance is exposed as an external CP of the VNF instance or is connected to an external CP of the VNF instance. See note 2. May be present otherwise.
>id	IdentifierInVnf	1	Identifier of this VNFC CP instance and the associated array entry.
>cpId	IdentifierInVnfd	1	Identifier of the VDU CPD, cpId, in the VNFD. See note 1.
>vnfcExtCpId	IdentifierInVnf	0..1	Identifier of the related external CP. Shall be present when the VNFC CP is exposed as an external CP of the VNF instance or connected to an external CP of the VNF instance (see note 2) and shall be absent otherwise.
>cpProtocolInfo	CpProtocolInfo	0..N	Network protocol information for this CP. May be omitted if the VNFC CP is exposed as an external CP. See note 3.
>parentCpId	IdentifierInVnf	0..1	Identifier of another VNFC CP instance that corresponds to the parent port of a trunk that the present VNFC CP instance participates in. Shall be provided if the present CP instance participates in a trunk as subport, and the referred VNFC CP instances are also present in the vnfcCpInfo attribute.
>vnfcLinkPortId	IdentifierInVnf	0..1	Identifier of the "vnfcLinkPortInfo" structure in the "VnfVirtualLinkResourceInfo" or "ExtManagedVirtualLinkInfo" structure. Shall be present if the CP is associated to a link port on an internal VL (including externally-managed internal VL) of the VNF instance and shall be absent otherwise.
>netAttDefResourceId	IdentifierInVnf	0..N	Identifier of the "NetAttDefResourceInfo" structure that provides the specification of the interface to attach the connection point to a secondary container cluster network. See notes 4 and 5.  It shall be present if the internal CP is associated to a VNFC realized by one or a set of OS containers and is connected to a secondary container cluster network. It shall not be present otherwise.
>metadata	KeyValuePairs	0..1	Metadata about this CP.
metadata	KeyValuePairs	0..1	Metadata about this resource.
NOTE 1: ETSI GS NFV-SOL 001 [i.1] specifies the structure and format of the VNFD based on TOSCA specifications.			
NOTE 2: A VNFC CP is "connected to" an external CP if the VNFC CP is connected to an internal VL that exposes an external CP. A VNFC CP is "exposed as" an external CP if it is connected directly to an external VL.			
NOTE 3: The information can be omitted because it is already available as part of the external CP information.			
NOTE 4: Cardinality greater than 1 is only applicable for specific cases where more than one network attachment definition resource is needed to fulfil the connectivity requirements of the internal CP, e.g. to build a link redundant mated pair in SR-IOV cases.			
NOTE 5: When more than one "netAttDefResourceId" is indicated, all shall belong to the same namespace.			

### 6.5.3.61 Type: VnfVirtualLinkResourceInfo

This type represents the information that allows addressing a virtualised resource that is used by an internal VL instance in a VNF instance. It shall comply with the provisions defined in table 6.5.3.61-1.

**Table 6.5.3.61-1: Definition of the VnfVirtualLinkResourceInfo data type**

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of this VnfVirtualLinkResourceInfo instance.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure).
vnfVirtualLinkDescId	IdentifierInVnfd	1	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.
networkResource	ResourceHandle	1	Reference to the VirtualNetwork resource or reference to a Network MCIO.
reservationId	Identifier	0..1	The reservation identifier applicable to the resource. It shall be present when an applicable reservation exists.
vnfLinkPorts	VnfLinkPortInfo	0..N	Links ports of this VL. Shall be present when the linkPort is used for external connectivity by the VNF (refer to VnfLinkPortInfo). May be present otherwise.
metadata	KeyValuePairs	0..1	Metadata about this resource.

### 6.5.3.62 Type: ExtVirtualLinkInfo

This type represents information about an VNF external VL. It shall comply with the provisions defined in table 6.5.3.62-1.

**Table 6.5.3.62-1: Definition of the ExtVirtualLinkInfo data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the external VL and the related external VL information instance.
resourceHandle	ResourceHandle	1	Reference to the resource realizing this VL.
extLinkPorts	ExtLinkPortInfo	0..N	Link ports of this VL.
extNetAttDefResource	NetAttDefResourceInfo	0..N	Network attachment definition resources that provide the specification of the interface to attach connection points to this VL.
currentVnfExtCpData	VnfExtCpData	1..N	Allows the API consumer to read the current CP configuration information for the connection of external CPs to the external virtual link, as specified in clause 5.5.3.2 of ETSI GS NFV-SOL 003 [4].

### 6.5.3.63 Type: ExtLinkPortInfo

This type represents information about a link port of an external VL, i.e. a port providing connectivity for the VNF to an NS VL. It shall comply with the provisions defined in table 6.5.3.63-1.

**Table 6.5.3.63-1: Definition of the ExtLinkPortInfo data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this link port as provided by the entity that has created the link port.
resourceHandle	ResourceHandle	1	Reference to the virtualised resource realizing this link port.
cplInstanceId	IdentifierInVnf	0..1	Identifier of the external CP of the VNF connected to this link port. There shall be at most one link port associated with any external connection point instance. The value refers to an "extCplInfo" item in the VnfInstance.

Attribute name	Data type	Cardinality	Description
secondaryCplInstanceid	IdentifierInVnf	0..1	Additional external CP of the VNF connected to this link port.  If present, this attribute shall refer to a "secondary" ExtCplInfo item in the VNF instance that exposes a virtual IP CP instance which shares this linkport with the external CP instance referenced by the "cplInstanceid" attribute.  See note 1.
trunkResourceid	IdentifierInVim	0..1	Identifier of the trunk resource in the VIM. Shall be present if the present link port corresponds to the parent port that the trunk resource is associated with. See note 2.
NOTE 1: The use cases UC#4 and UC#5 in Annex A.4 of ETSI GS NFV-IFA 007 [19] provide examples for such a configuration. NOTE 2: The value of "trunkResourceid" is scoped by the value of "vimConnectionId" in the "resourceHandle" attribute.			

### 6.5.3.64 Type: VnfLinkPortInfo

This type represents a link port of an internal VL of a VNF. It shall comply with the provisions defined in table 6.5.3.64-1.

**Table 6.5.3.64-1: Definition of the VnfLinkPortInfo data type**

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of this link port as provided by the entity that has created the link port.
resourceHandle	ResourceHandle	1	Reference to the virtualised resource realizing this link port.
cplInstanceid	IdentifierInVnf	0..1	When the link port is used for external connectivity by the VNF, this attribute represents the identifier of the external CP associated with this link port.  When the link port is used for internal connectivity in the VNF, this attribute represents the identifier of the VNFC CP to be connected to this link port.  Shall be present when the link port is used for external connectivity by the VNF.  May be present if used to reference a VNFC CP instance.  There shall be at most one link port associated with any external connection point instance or internal connection point (i.e. VNFC CP) instance.  The value refers to an "extCplInfo" item in the VnfInstance or a "vnfcCplInfo" item of a "vnfcResourceInfo" item in the VnfInstance. See note 1.
cplInstanceType	Enum (inlined)	0..1	Type of the CP instance that is identified by cplInstanceid.  Shall be present if "cplInstanceid" is present, and shall be absent otherwise.  Permitted values: <ul style="list-style-type: none"> <li>- VNFC_CP: The link port is connected to a VNFC CP.</li> <li>- EXT_CP: The link port is associated to an external CP.</li> </ul> See note 1.

Attribute name	Data type	Cardinality	Description
vipCplInstanceld	IdentifierInVnf	0..1	VIP CP instance of the VNF connected to this link port. May be present. See notes 1 and 2.
trunkResourceld	IdentifierInVim	0..1	Identifier of the trunk resource in the VIM. Shall be present if the present link port corresponds to the parent port that the trunk resource is associated with. See note 3.
<p>NOTE 1: Either cplInstanceld with cplInstanceType set to "EXT_CP" or any combination of cplInstanceld with cplInstanceType set to "VNFC_CP" and vipCplInstanceld (i.e. one or both of them) shall be present for a VnfLinkPortInfo. In case both cplInstanceld with cplInstanceType set to "VNFC_CP" and vipCplInstanceld are present, the two different CP instances share the linkport.</p> <p>NOTE 2: Clause A.4 of ETSI GS NFV-IFA 007 [19] provides examples for configurations where both vipCplInstanceld and vnfcCplInstanceld are present (UC#5 and UC#5-b), only vnfcCplInstanceld is present (UC#2), or only vipCplInstanceld is present (UC#6 and UC#6-b).</p> <p>NOTE 3: The value of "trunkResourceld" is scoped by the value of "vimConnectionId" in the "resourceHandle" attribute.</p>			

### 6.5.3.65 Type: VnffgInfo

This type specifies the information about a VNFFG instance. It shall comply with the provisions defined in table 6.5.3.65-1.

**Table 6.5.3.65-1: Definition of the VnffgInfo data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this VNFFG instance.
vnffgdId	IdentifierInNsd	1	Identifier of the VNFFGD in the NSD.
vnfInstanceld	Identifier	1..N	Identifier(s) of the constituent VNF instance(s) of this VNFFG instance.
pnfInfold	Identifier	0..N	Identifier(s) of the constituent PNF instance(s) of this VNFFG instance.
nsVirtualLinkInfold	IdentifierInNs	1..N	Identifier(s) of the constituent VL instance(s) of this VNFFG instance.
nsCpHandle	NsCpHandle	1..N	Identifiers of the CP instances attached to the constituent VNFs and PNFs or the SAP instances of the VNFFG. See note.
nfpInfo	NfpInfo	1..N	Information on the NFP instances.
NOTE: It indicates an exhaustive list of all the CP instances and SAP instances of the VNFFG.			

### 6.5.3.66 Type: NfpInfo

This type represents an NFP instance. It shall comply with the provisions defined in table 6.5.3.66-1.

**Table 6.5.3.66-1: Definition of the NfpInfo data type**

Attribute name	Data type	Cardinality	Description
id	IdentifierInNs	1	Identifier of this NFP instance.
nfpdId	IdentifierInNsd	0..1	Identifier of the NFPD used to instantiate this NFP instance. It shall be present if the NFP instance is instantiated from the NFPD.
nfpName	String	0..1	Human readable name for the NFP instance.
description	String	0..1	Human readable description for the NFP instance.
cpGroup	CpGroupInfo	1..N	Group(s) of CPs and/or SAPs which the NFP passes through. See note.
totalCp	Integer	0..1	Total number of CP and SAP instances in this NFP instance.
nfpRule	NfpRule	1	NFP classification and selection rule.

Attribute name	Data type	Cardinality	Description
nfpState	Enum (inlined)	1	The state of the NFP instance.  Permitted values: - ENABLED: The NFP instance is enabled. - DISABLED: The NFP instance is disabled.
NOTE: When multiple identifiers are included, the position of the identifier in the CpGroup data type specifies the position of the group in the path.			

### 6.5.3.67 Type: SapInfo

This type represents an SAP instance. It shall comply with the provisions defined in table 6.5.3.67-1.

**Table 6.5.3.67-1: Definition of the SapInfo data type**

Attribute name	Data type	Cardinality	Description
id	IdentifierInNs	1	Identifier of the SAP instance.
sapId	IdentifierInNsd	1	Identifier of the SAPD in the NSD.
sapName	String	1	Human readable name for the SAP instance.
description	String	1	Human readable description for the SAP instance.
sapProtocolInfo	CpProtocolInfo	1..N	Network protocol information for this SAP.

### 6.5.3.68 Type: NsMonitoringParameter

This type represents a monitoring parameter that is tracked by the NFVO, for example, for auto-scaling purposes. It shall comply with the provisions defined in table 6.5.3.68-1.

**Table 6.5.3.68-1: Definition of the NsMonitoringParameter data type**

Attribute name	Data type	Cardinality	Description
id	IdentifierInNsd	1	Identifier of the monitoring parameter defined in the NSD.
name	String	0..1	Human readable name of the monitoring parameter, as defined in the NSD.
performanceMetric	String	1	Performance metric that is monitored. This attribute shall contain the related "Measurement Name" value as defined in clause 7.2 of ETSI GS NFV-IFA 027 [15].

### 6.5.3.69 Type: VnfMonitoringParameter

This type represents a monitoring parameter that is tracked by the VNFM, for example, for auto-scaling purposes. It shall comply with the provisions defined in table 6.5.3.69-1.

**Table 6.5.3.69-1: Definition of the VnfMonitoringParameter data type**

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnfd	1	Identifier of the monitoring parameter defined in the VNFD.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure).
name	String	0..1	Human readable name of the monitoring parameter, as defined in the VNFD.
performanceMetric	String	1	Performance metric that is monitored. This attribute shall contain the related "Measurement Name" value as defined in clause 7.2 of ETSI GS NFV-IFA 027 [15].

## 6.5.3.70 Type: VnfExtCplInfo

This type represents information about an external CP of a VNF. It shall comply with the provisions defined in table 6.5.3.70-1.

Table 6.5.3.70-1: Definition of the VnfExtCplInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of the external CP instance and the related information instance.
cpId	IdentifierInVnfd	1	Identifier of the external CPD, VnfExtCpd, in the VNFD.
cpConfigId	IdentifierInVnf	1	Identifier that references the applied "VnfExtCpConfig" entry in the "cpConfig" map of the "currentVnfExtCpData" in the "ExtVirtualLinkInfo" structure.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure).
cpProtocolInfo	CpProtocolInfo	1..N	Network protocol information for this CP.
extLinkPortId	Identifier	0..1	Identifier of the "extLinkPortInfo" structure inside the "extVirtualLinkInfo" structure. Shall be present if the CP is associated to a link port. See note 2.
netAttDefResourceId	IdentifierInVnf	0..N	Identifier of the "NetAttDefResourceInfo" structure that provides the specification of the interface to attach the connection point to a secondary container cluster network. See notes 3 and 4.  It shall be present if the external CP is associated to a VNFC realized by one or a set of OS containers and is connected to a secondary container cluster network. It shall not be present otherwise.
metadata	KeyValuePairs	0..1	Metadata about this external CP.
associatedVnfcCpld	IdentifierInVnf	0..1	Identifier of the "vnfcCplInfo" structure in "VnfcResourceInfo" structure that represents the VNFC CP which is exposed by this external CP instance, either directly or via a floating IP address. Shall be present in case this CP instance maps to a VNFC CP. See note 1.
associatedVipCpld	IdentifierInVnf	0..1	Identifier of the "VipCplInfo" structure that represents the VIP CP that is exposed by this external CP instance, either directly or via a floating IP address, and the related "VipCplInfo" structure in "VnfInstance". Shall be present in case this CP instance maps to a VIP CP. See note 1.
associatedVirtualCpld	IdentifierInVnf	0..1	Identifier of the "VirtualCplInfo" structure that represents the virtual CP that is exposed by this external CP instance. Shall be present in case this CP instance maps to a virtual CP. See note 1.
associatedVnfVirtualLinkId	IdentifierInVnf	0..1	Identifier of the "VnfVirtualLinkResourceInfo" structure that represents the internal VL or of the "ExtManagedVirtualLinkInfo" structure that represents the externally-managed internal VL, which is exposed by this external CP instance. Shall be present in case this CP instance maps to an internal VL (including externally-managed internal VL). See note 1.
NOTE 1: The attributes "associatedVnfcCpld", "associatedVipCpld", "associatedVirtualCpld" and "associatedVnfVirtualLinkId" are mutually exclusive. Exactly one shall be present.			
NOTE 2: An external CP instance is not associated to a link port in the cases indicated for the "extLinkPorts" attribute in clause 6.5.3.26.			
NOTE 3: Cardinality greater than 1 is only applicable for specific cases where more than one network attachment definition resource is needed to fulfil the connectivity requirements of the external CP, e.g. to build a link redundant mated pair in SR-IOV cases.			
NOTE 4: When more than one "netAttDefResourceId" is indicated, all shall belong to the same namespace.			

### 6.5.3.71 Type: CpGroupInfo

This type represents describes a group of CPs and/or SAPs pairs associated to the same position in an NFP. It shall comply with the provisions defined in table 6.5.3.71-1.

**Table 6.5.3.71-1: Definition of the CpGroupInfo data type**

Attribute name	Data type	Cardinality	Description
cpPairInfo	CpPairInfo	1..N	One or more pair(s) of ingress and egress CPs or SAPs which the NFP passes by.  See note.
forwardingBehaviour	Enum (inlined)	0..1	Identifies a rule to apply to forward traffic to the ingress CPs or SAPs of the group.  Permitted values: ALL = Traffic flows shall be forwarded simultaneously to all CPs or SAPs of the group. LB = Traffic flows shall be forwarded to one CP or SAP of the group selected based on a load-balancing algorithm.
forwardingBehaviourInputParameters	ForwardingBehaviourInputParameters	0..1	Provides input parameters to configure the forwarding behaviour (e.g. identifies a load balancing algorithm and criteria).
NOTE: All CP or SAP pairs in a group shall be instantiated from connection point descriptors or service access point descriptors referenced in the corresponding NfpPositionDesc.			

### 6.5.3.72 Type: CpPairInfo

This type represents describes a pair of ingress and egress CPs or SAPs which the NFP passes by. It shall comply with the provisions defined in table 6.5.3.72-1.

**Table 6.5.3.72-1: Definition of the CpPairInfo data type**

Attribute name	Data type	Cardinality	Description
vnfExtCplds	IdentifierInVnf	0..2	Identifier(s) of the VNF CP(s) which form the pair.  See note 1 and note 2.
pnfExtCplds	IdentifierInPnf	0..2	Identifier(s) of the PNF CP(s) which form the pair.  See note 1 and note 2.
saplds	IdentifierInNs	0..2	Identifier(s) of the SAP(s) which form the pair.  See note 1 and note 2.
NOTE 1: The presence of a single vnfExpCpld, pnfExtCpld, or sapld occurrence indicates that the CP or SAP is used both as an ingress and egress port at a particular NFP position.			
NOTE 2: Only one of these three attributes shall be present.			

### 6.5.3.73 Type: ForwardingBehaviourInputParameters

This type represents provides input parameters to configure the forwarding behaviour. It shall comply with the provisions defined in table 6.5.3.73-1.



**Table 6.5.3.73-1: Definition of the ForwardingBehaviourInputParameters data type**

Attribute name	Data type	Cardinality	Description
algorithmName	Enum (Inlined)	0..1	May be included if forwarding behaviour is equal to LB. Shall not be included otherwise.  Permitted values: - ROUND_ROBIN - LEAST_CONNECTION - LEAST_TRAFFIC - LEAST_RESPONSE_TIME - CHAINED_FAILOVER - SOURCE_IP_HASH - SOURCE_MAC_HASH
algorithmWeights	Integer	0..N	Percentage of messages sent to a CP instance. May be included if applicable to the algorithm. See note 1 and note 2.
NOTE 1: If applicable to the algorithm but not provided, default values determined by the VIM or NFVI are expected to be used.			
NOTE 2: Weight applies to the CP instances in the order they have been created.			

#### 6.5.3.74 Type: DeleteVnfSnapshotData

This type specifies the identifier of information of an available VNF Snapshot to be deleted and the identifier of the related VNF instance of the NS instance. It shall comply with the provisions defined in table 6.5.3.74-1.

**Table 6.5.3.74-1: Definition of the DeleteVnfSnapshotData data type**

Attribute name	Data type	Cardinality	Description
vnfInstanceld	Identifier	1	Identifier of the VNF instance to identify the VNFM holding the VNF Snapshot information to be deleted.
vnfSnapshotInfold	Identifier	1	Identifier of information held by the VNFM about the VNF Snapshot to be deleted. This identifier was allocated by the VNFM.

#### 6.5.3.75 Type: RevertVnfToSnapshotData

This type specifies the identifier of an existing VNF instance of the NS instance to be reverted and the identifier of an existing VNF Snapshot to be reverted to. It shall comply with the provisions defined in table 6.5.3.75-1.

**Table 6.5.3.75-1: Definition of the RevertVnfToSnapshotData data type**

Attribute name	Data type	Cardinality	Description
vnfInstanceld	Identifier	1	Identifier of the VNF instance to be reverted.
vnfSnapshotInfold	Identifier	1	Identifier of information held by the VNFM about the individual VNF Snapshot resource with the information of the VNF snapshot to be reverted to. This identifier was allocated by the VNFM.
additionalParams	KeyValuePairs	0..1	Additional input parameters for the revert to VNF snapshot process, specific for the VNF being "reverted", as declared in the VNFD as part of "RevertToSnapshotVnfOpConfig" defined in ETSI GS NFV-IFA 011 [29].

#### 6.5.3.76 Type: CreateVnfSnapshotData

This type represents the information that are needed for VNF snapshot creation. When the NFVO invokes the Create VNF snapshot operation, a set of these parameters are then passed by the NFVO to the VNFM. It shall comply with the provisions defined in table 6.5.3.76-1.

**Table 6.5.3.76-1: Definition of the CreateVnfSnapshotData data type**

Attribute name	Data type	Cardinality	Description
vnfInstanceId	Identifier	1	Identifier of the VNF instance to perform the snapshot from.
additionalParams	KeyValuePairs	0..1	Additional input parameters for the snapshot creation process, specific for the VNF being "snapshotted", as declared in the VNFD as part of "CreateSnapshotVnfOpConfig" defined in ETSI GS NFV-IFA 011 [29].
userDefinedData	KeyValuePairs	0..1	User defined data for the VNF snapshot.

### 6.5.3.77 Type: VnfcSnapshotInfo

This type represents a VNFC Snapshot. It shall comply with the provisions defined in table 6.5.3.77-1.

**Table 6.5.3.77-1: Definition of the VnfcSnapshotInfo data type**

Attribute name	Data type	Cardinality	Description
id	IdentifierLocal	1	Identifier of the information held by the VNFM about a specific VNFC Snapshot. This identifier is allocated by the VNFM and is unique within the scope of a VNF snapshot. The attribute also identifies the compute snapshot image associated to this VNFC snapshot within the context of a referred VNF snapshot.
vnfcInstanceId	IdentifierInVnf	1	Identifier of the snapshotted VNFC instance.
creationStartedAt	DateTime	1	Timestamp indicating when the VNFC Snapshot creation has been started by the VNFM.
creationFinishedAt	DateTime	1	Timestamp indicating when the VNFC Snapshot has been completed by the VNFM.
vnfcResourceInfo	IdentifierInVnf	1	Reference to the "VnfcResourceInfo" structure that represents the resources of the snapshotted VNFC instance. A snapshot of that structure is available in the "vnfInstance" attribute of the "VnfSnapshot" structure.
computeSnapshotResource	ResourceHandle	0..1	Reference to a compute snapshot resource. See note 1.
storageSnapshotResources	Structure (inlined)	0..N	Mapping of the storage resources associated to the VNFC with the storage snapshot resources.
>storageResourceInfo	IdentifierInVnf	1	Reference to the "VirtualStorageResourceInfo" structure in the "VnfInstance" structure that represents the virtual storage resource. The attribute also identifies the storage snapshot image associated to this VNFC snapshot within the context of a referred VNF snapshot.
>storageSnapshotResources	ResourceHandle	0..1	Reference to a storage snapshot resource. See note 2.
userDefinedData	KeyValuePairs	0..1	User defined data for the VNFC Snapshot.
NOTE 1: The identifier of the compute snapshot resource is assigned during creation of a VNFC Snapshot being returned from the VIM as output data in the response message of the individual resource operations. This attribute shall only be present for a VNFC snapshot that has been newly created by the VNFM as a result of the "Create VNF snapshot task".			
NOTE 2: The identifier of the storage snapshot resource is assigned during creation of a VNFC snapshot being returned from the VIM as output data in the response message of the individual resource operations. This attribute shall only be present for a VNFC snapshot with an associated storage resource and that has been newly created by the VNFM as a result of the "Create VNF snapshot task".			

### 6.5.3.78 Type: VnfLinkPortData

This type represents an externally provided link port to be used to connect a VNFC connection point to an externally-managed VL. It shall comply with the provisions defined in table 6.5.3.78-1.

**Table 6.5.3.78-1: Definition of the VnfLinkPortData data type**

Attribute name	Data type	Cardinality	Description
vnfLinkId	Identifier	1	Identifier of this link port as provided by the entity that has created the link port.
resourceHandle	ResourceHandle	1	Resource handle of the virtualised resource that realizes the link port.

### 6.5.3.79 Type: ModificationsTriggeredByVnfPkgChange

This type represents attribute modifications that were performed by the VNFM on an "Individual VNF instance" resource when changing the current VNF package.

The "ModificationsTriggeredByVnfPkgChange" data type shall comply with the provisions defined in table 6.5.3.79-1.

**Table 6.5.3.79-1: Definition of the ModificationsTriggeredByVnfPkgChange data type**

Attribute name	Data type	Cardinality	Description
vnfConfigurableProperties	KeyValuePairs	0..1	This attribute signals the modifications of the "vnfConfigurableProperties" attribute in "VnfInstance" performed by the operation and shall be present if that attribute was modified during the operation. See note 1. In addition, the provisions in clause 6.7 shall apply.
metadata	KeyValuePairs	0..1	This attribute signals the modifications of the "metadata" attribute in "VnfInstance" performed by the operation and shall be present if that attribute was modified during the operation. See note 1.
extensions	KeyValuePairs	0..1	This attribute signals the modifications of the "extensions" attribute in "VnfInstance" performed by the operation and shall be present if that attribute was modified during the operation. See note 1. In addition, the provisions in clause 6.7 shall apply.
vnfdId	Identifier	0..1	If present, this attribute signals the new value of the "vnfdId" attribute in "VnfInstance".
vnfProvider	String	0..1	If present, this attribute signals the new value of the "vnfProvider" attribute in "VnfInstance". See note 2.
vnfProductName	String	0..1	If present, this attribute signals the new value of the "vnfProductName" attribute in "VnfInstance". See note 2.
vnfSoftwareVersion	Version	0..1	If present, this attribute signals the new value of the "vnfSoftwareVersion" attribute in "VnfInstance". See note 2.
vnfdVersion	Version	0..1	If present, this attribute signals the new value of the "vnfdVersion" attribute in "VnfInstance". See note 2.
NOTE 1: This attribute represents the delta (semantics as per IETF RFC 7396 [11], JSON Merge Patch) between the value of the attribute at the start of the "Change current VNF package" operation and the value of the attribute at its completion.			
NOTE 2: If present, this attribute (which depends on the value of the "vnfdId" attribute) was modified implicitly during the related operation and contains a copy of the value of the related attribute from the VNFD in the VNF Package identified by the "vnfdId" attribute.			

### 6.5.3.80 Type: WanConnectionData

This type provides information used to connect the comprising network resources realizing a VL, e.g. when the VL is deployed on several sites and across a WAN. It shall comply with the provisions defined in table 6.5.3.80-1.

Table 6.5.3.80-1: Definition of the WanConnectionData data type

Attribute name	Data type	Cardinality	Description
nsVirtualLink	Structure (inlined)	0..1	Information used to identify the NS VL for which the WAN connectivity data is applicable. See note.
>nsVirtualLinkDescId	IdentifierInNsd	1	Identifier of the VLD in the NSD from which the VL is created in the case of a multi-site NS VL.
>nsVirtualLinkProfileId	IdentifierInNsd	1	Identifier of the VL profile in the NSD.
vnfVirtualLink	Structure (inlined)	0..1	Information used to identify the VNF VL for which the WAN connectivity data is applicable. See note.
>vnfProfileId	IdentifierInNsd	1	Identifier of the VNF profile.
>vnfVirtualLinkDescId	IdentifierInVnfd	1	Identifier of the VLD in the VNFD from which the VL is created in the case of a multi-site internal VNF VL. See note.
>vnfVirtualLinkProfileId	IdentifierInVnfd	1	Identifier of the VL profile in the VNFD.
protocolData	WanConnectionProtocolData	1	Protocol specific information for connecting to the WAN.
NOTE: Either a "nsVirtualLink" or a "vnfVirtualLink" shall be provided, but not both.			

### 6.5.3.81 Type: WanConnectionProtocolData

This type provides protocol specific information used to connect the comprising network resources realizing a VL, e.g. when the VL is deployed on several sites and across a WAN. This type supports signalling input information about both pre-provisioned WAN connectivity realized by external entities to NFV-MANO, as well as for the creation of MSCS under NFV-MANO responsibility (i.e. when connectivity is realized when NFVO communicates with the WIM). It shall comply with the provisions defined in table 6.5.3.81-1.

Table 6.5.3.81-1: Definition of the WanConnectionProtocolData data type

Attribute name	Data type	Cardinality	Description
mscsInfo	MscsInfo	0..1	Information about the pre-provisioned multi-site connectivity service (MSCS), if already available. See note.
connectivityServiceEndpointConfigData	ConnectivityServiceEndpointInfo	0..N	Configuration data for the network resources in the NFVI-PoP. See note.
mscsConfigData	MscsConfigData	0..1	Configuration data for the provisioning of the MSCS, if such MSCS is to be created by NFV-MANO. See note.
NOTE: At least one of these attributes shall be present. Annex E documents the applicability of certain attributes depending on the WAN and NFVI-PoP network management and the responsibilities of NFV-MANO in its provisioning.			

### 6.5.3.82 Type: MscsInfo

This type provides information about an already provisioned multi-site connectivity service (MSCS) deployed across a WAN. It shall comply with the provisions defined in table 6.5.3.82-1.

Table 6.5.3.82-1: Definition of the MscsInfo data type

Attribute name	Data type	Cardinality	Description
mscsId	Identifier	1	Identifier of the MSCS established over the WAN.
mscsName	String	0..1	Human readable name of the MSCS.
mscsDescription	String	0..1	Human readable description of the MSCS.
mscsType	Enum (inlined)	1	The type of connectivity that is provided to the virtualised networks in the NFVI-PoP and characterizes the connectivity service across the WAN.  Permitted values: - L1 - L2 - L3
mscsLayerProtocol	Enum (inlined)	0..1	Type of underlying connectivity service and protocol associated to the MSCS.

Attribute name	Data type	Cardinality	Description
			Permitted values are as listed below and restricted by the type of MSCS: <ul style="list-style-type: none"> <li>- EVPN_BGP/MPLS: as specified in IETF RFC 7432 [21]. Only applicable for mscsType="L2".</li> <li>- EVPN_VPWS: as specified in IETF RFC 8214 [30]. Only applicable for mscsType="L2".</li> <li>- VPLS_BGP: as specified in IETF RFC 4761 [22] and IETF RFC 6624 [23]. Only applicable for mscsType="L2".</li> <li>- VPLS_LDP_L2TP: as specified in IETF RFC 4762 [24] and IETF RFC 6074 [25]. Only applicable for mscsType="L2".</li> <li>- VPWS_LDP_L2TP: as specified in IETF RFC 6074 [25]. Only applicable for mscsType="L2".</li> <li>- BGP_IP_VPN: BGP/MPLS based IP VPN as specified in IETF RFC 4364 [26]. Only applicable for mscsType="L3".</li> </ul>
siteAccessProtectionSchemes	Enum (inlined)	0..N	Information to determine the proper MSCS endpoints configuration to fulfil certain resiliency/protection requirements, e.g. by considering certain availability and redundancy of connectivity service endpoints in between NFVI-PoP and WAN.
>locationConstraints	LocationConstraints	1	Defines the location constraints according to an NFVI-PoP (site).
>protectionScheme	Enum (inlined)	1	Defines the protection scheme.  Permitted values: <ul style="list-style-type: none"> <li>- UNPROTECTED: to indicate no protection.</li> <li>- ONE_TO_ONE: to indicate an active-passive access protection.</li> <li>- ONE_PLUS_ONE: to indicate an active-active access protection.</li> <li>- ONE_TO_N: to indicate an N active to 1 passive access protection.</li> </ul>
mtuMscs	Number	0..1	Maximum Transmission Unit (MTU) that can be forwarded over the MSCS (in bytes). Default value is "1500" (bytes).
mscsEndpoints	MscsEndpointInfo	0..N	Information about the MSCS endpoints of the MSCS.

### 6.5.3.83 Type: MscsEndpointInfo

This type provides information about an MSCS endpoint of the MSCS. It shall comply with the provisions defined in table 6.5.3.83-1.

**Table 6.5.3.83-1: Definition of the MscsEndpointInfo data type**

Attribute name	Data type	Cardinality	Description
mscsEndpointId	Identifier	1	Identifier of the MSCS endpoint.
directionality	Enum (inlined)	1	Directionality of the data traffic in the context of the terminating MSCS endpoint from WAN's perspective.  Permitted values: <ul style="list-style-type: none"> <li>- INBOUND: to indicate into the WAN.</li> <li>- OUTBOUND: to indicate from the WAN.</li> <li>- BOTH: to indicate bidirectional data traffic to/from the WAN.</li> </ul>
connectivityServiceEndpointId	Identifier	1..N	References the connectivity service endpoint configuration information applicable to support the MSCS endpoint. More than one connectivity service endpoint can be referred when endpoints are in LAG mode.

### 6.5.3.84 Type: ConnectivityServiceEndpointInfo

This type provides configuration data for the NFVI-PoP network gateway and its connectivity service endpoints. The connectivity service endpoints are used as endpoints by an MSCS. It shall comply with the provisions defined in table 6.5.3.84-1.

**Table 6.5.3.84-1: Definition of the ConnectivityServiceEndpointInfo data type**

Attribute name	Data type	Cardinality	Description
connectivityServiceEndpointId	Identifier	1	Identifier of the connectivity service endpoint of the NFVI-PoP (site) providing the UNI in between the NFVI-PoP and the WAN.
vimId	Identifier	1	Identifier of the VIM responsible for the management of the virtualised networks in the NFVI-PoP of the applicable VL, and for which the connection data is applicable.
siteToWanLayer2ProtocolData	SiteToWanLayer2ProtocolData	0..1	Layer 2 protocol specific information for the configuration of the NFVI-PoP network gateway to enable the stitching of the intra-site VN to the MSCS over the WAN.
siteToWanLayer3ProtocolData	SiteToWanLayer3ProtocolData	0..1	Layer 3 protocol specific information for the configuration of NFVI-PoP network gateway to enable the stitching of the intra-site VN to the MSCS over the WAN.

### 6.5.3.85 Type: SiteToWanLayer2ProtocolData

This type provides information about Layer 2 protocol specific information for the configuration of the NFVI-PoP network gateway to enable the stitching of the intra-site VN to the MSCS over the WAN. It shall comply with the provisions defined in table 6.5.3.85-1.

**Table 6.5.3.85-1: Definition of the SiteToWanLayer2ProtocolData data type**

Attribute name	Data type	Cardinality	Description
layer2ConnectionInfo	Structure (inlined)	1	Layer 2 protocol parameters of the Connectivity Service Endpoint (CSE).
>connectionType	Enum (inlined)	1	The type of connection to be established on the connectivity service point.  Permitted values: <ul style="list-style-type: none"> <li>- CSE: defined by the characteristics of the existing referred connectivity service point.</li> <li>- AGGREGATE_CSE: create an aggregation of the connectivity service endpoints.</li> </ul>
>interfaceType	Enum (inlined)	1	To indicate whether to create logical interfaces on the referred connectivity service endpoint or new aggregated connectivity service endpoint.  Permitted values: <ul style="list-style-type: none"> <li>- PARENT: use the mapped interface to the connectivity service endpoint as is, i.e. do not create logical interfaces.</li> <li>- LOGICAL: create logical interfaces.</li> </ul>
>interfaceTagging	Enum (inlined)	1	The type of frames to forward on the connectivity service point.  Permitted values: <ul style="list-style-type: none"> <li>- UNTAGGED: an interface where frames are not tagged.</li> <li>- TAGGED: an interface configured to forward tagged frames (i.e. enabled for VLAN tagging).</li> </ul>

Attribute name	Data type	Cardinality	Description
>encapsulationType	Enum (inlined)	1	The type of encapsulation. If the interfaceTagging="TAGGED", either "VLAN" or "VXLAN" shall be set.  Permitted values: - ETH: generic Ethernet encapsulation. - VLAN: encapsulation based on VLAN. - VXLAN: encapsulation based on VXLAN.
>vlanTaggingType	Enum (inlined)	0..1	Type of encapsulation method for VLAN tagging. Shall be present if interfaceTagging="TAGGED" and encapsulationType="VLAN".  Permitted values: - DOT1Q: used when packets on the CSE are encapsulated with one or a set of customer VLAN identifiers. - QINQ: used when packets on the CSE are encapsulated with multiple customer VLAN identifiers and a single service VLAN identifier. - QINANY: used when packets on the CSE have no specific customer VLAN and a service VLAN identifier is used.
>wanSegmentIds	Structure (inlined)	0..N	Segment identifiers to pass on the tagged interface. Shall be present if encapsulationType="VLAN" or "VXLAN".
>>wanSegmentIdValue	String	1	Identifier of the network segment (e.g. VLAN id or VNI).
>>wanSegmentIdUpperRange	String	0..1	Identifier of the upper range network segment, in case the "wanSegmentIds" is used to define a range.
>vxlanConfig	Structure (inlined)	0..1	Additional configuration needed when using VXLAN encapsulation. Shall be present if interfaceTagging="TAGGED" and encapsulationType="VXLAN".
>>peerMode	Enum (inlined)	1	Type of VXLAN access mode. Default value is "STATIC".  Permitted values: - STATIC - BGP_EVPN
>>peers	IpAddress	0..N	List of IP addresses of VTEP peers when using static mode.
>lagInterfaceData	Structure (inlined)	0..1	Information for setting up a LAG interface aggregating multiple connectivity service endpoints.
>>aggregatedEndpoints	Identifier	1..N	List of the connectivity service endpoints that are to be aggregated. Shall be present if connectionType="AGGREGATE_CSE". In case of aggregating connectivity service endpoints, only one SiteToWanLayer2ProtocolData shall be provided for the whole set of aggregated endpoints.
>>lacpActivation	Boolean	1	Indicates whether to activate LACP on the interface. If "TRUE", the LACP is to be activated, or "FALSE" otherwise. Default value is "FALSE".
>>lacpConfig	KeyValuePairs	0..1	Specific configuration for the LACP. Examples include configuration to indicate the minimum number of active connectivity service endpoints, the time internal and priority of LACP packet data units (PDU).
>layer2ControlProtocol	KeyValuePairs	0..1	Configuration enabling the L2 control protocol in between the NFVI-PoP network gateway and the WAN.
mtuL2	UnsignedInt	0..1	Maximum Transmission Unit (MTU) that can be forwarded at layer 2 (in bytes). Default value is "1500" (bytes).
virtualRoutingAndForwarding	Structure (inlined)	0..1	Configuration related to the L2 virtual routing and forwarding (MAC-VRF).
>macVrfName	String	1	Name (or identifier) of the MAC-VRF instance.

Attribute name	Data type	Cardinality	Description
forwardingConfig	Structure (inlined)	0..1	Information related to the forwarding of the VN in the NFVI-PoP to the connectivity service endpoint, if information about the VN to "stitch" is already known by the OSS/BSS. Shall not be provided otherwise, in which case the NFVO will infer the forwarding configuration based on the NS VL, or external VL, or externally-managed VL provisioning.
>networkResources	ResourceHandle	0..N	Reference to the VN resource to be forwarded into/from the MSCS. See note.
>vnSegmentIds	Structure (inlined)	0..N	Identifiers of the network segments of the VN resources to be forwarded into/from the MSCS. See note.
>>vnSegmentIdValue	String	1	Identifier of the network segment.
>>vnSegmentIdUpperRange	String	0..1	Identifier of the upper range network segment, in case the "vnSegmentIds" is used to define a range.
NOTE: Either "networkResources" or "vnSegmentIds" shall be provided, but not both.			

### 6.5.3.86 Type: SiteToWanLayer3ProtocolData

This type provides information about Layer 3 protocol specific information for the stitching of the intra-site VN to the multi-site connectivity service over the WAN. It shall comply with the provisions defined in table 6.5.3.86-1.

**Table 6.5.3.86-1: Definition of the SiteToWanLayer3ProtocolData data type**

Attribute name	Data type	Cardinality	Description
logicalInterfaceIpAddress	Structure (inlined)	0..1	IP addressing information associated to a logical interface. Shall be present if the "interfaceType" of the SiteToWanLayer2ProtocolData is equal to "LOGICAL".
>ipAddress	IpAddress	1	The IP address to be assigned to the interface.
>associatedSegmentId	String	1	The associated segment identifier that has triggered the creation of the logical interface. The value shall be one of the values listed in the "wanSegmentIds" of the "siteToWanLayer2ProtocolData".
routingInfo	Structure (inlined)	1..N	The routing information that is activated on the connectivity service endpoint. More than one "routingInfo" is allowed to enable stacking different routing protocols (e.g. one routing protocol for IPv4 and another one for IPv6).
>routingProtocol	Enum (inlined)	1	The routing protocol that is activated on the connectivity service endpoint.  Permitted values: <ul style="list-style-type: none"> <li>- BGP: used for dynamic routing BGPv4.</li> <li>- RIP: used for dynamic routing RIPv2.</li> <li>- OSPF: used for dynamic routing (OSPF version 2 for IPv4; and OSPF version 3 for IPv6).</li> <li>- STATIC: used for static routing.</li> <li>- DIRECT: used when the NFVI-PoP network is directly connected to the WAN provider network.</li> <li>- VRRP: used when the NFVI-PoP network is directly connected to the WAN provider network with Virtual Router Redundancy Protocol support (VRRP).</li> </ul>
>routingAddressFamily	Enum (inlined)	1	The IP version applicable to the dynamic routing protocol type.  Permitted values: <ul style="list-style-type: none"> <li>- IPV4</li> <li>- IPV6</li> </ul>
>staticRouting	Structure (inlined)	0..N	Defines a static route. It shall only be present if the routingProtocol="STATIC".



Attribute name	Data type	Cardinality	Description
>>ipVersion	Enum (inlined)	1	The IP version applicable to the routing entry.  Permitted values: - IPV4 - IPV6
>>ipPrefix	IpAddress	1	The IP network prefix.
>>prefixSize	UnsignedInt	1	The IP prefix size.
>>nextHop	IpAddress	1	The next hop's IP address.
>ospfRouting	Structure (inlined)	0..1	Defines parameters for OSPF routing. It shall only be present if the routingProtocol="OSPF".
>>areald	String	1	The routing area identifier, e.g. a number or an IP address.
>bgpRouting	Structure (inlined)	0..1	Defines parameters for BGP routing. It shall only be present if the routingProtocol="BGP".
>>bgpAs	UnsignedInt	1	The Autonomous System (AS) identification applicable to the BGP routing info entry.
>>bgpNeighbour	IpAddress	0..1	The IP address of the BGP neighbour.
>>bgpAdditionalParam	KeyValuePairs	0..1	Additional parameters for setting up the BGP neighbour, such as information about the remote AS.
>routeMapDistributions	Structure (inlined)	0..N	Maps of routes that are permitted or denied for redistribution.
>>policy	Enum (inlined)	1	The policy to apply to the route distribution.  Permitted values: - PERMIT - DENY
>>>matchAndSetRule	KeyValuePairs	1	Set of "match" and "set" rules for the distribution of routing information. "Match" key entries define the routes that are selected. "Set" key entries define what information to modify when distributing the routing information.
>>>sequence	UnsignedInt	1	Sequence or index number assigned to the route-map.
mtuL3	UnsignedInt	0..1	Maximum Transmission Unit (MTU) that can be forwarded at layer 3 (in bytes). Default value is "1500" (bytes).
virtualRoutingAndForwarding	Structure (inlined)	0..1	Configuration related to the Virtual Routing and Forwarding (VRF).
>vrfName	String	1	Name (or identifier) of the VRF instance.
bfdConfig	KeyValuePairs	0..1	Configuration related to Bidirectional Forwarding Detection (BFD).

### 6.5.3.87 Type: MscsConfigData

This type provides configuration data for the creation of an MSCS. It shall comply with the provisions defined in table 6.5.3.87-1.

**Table 6.5.3.87-1: Definition of the MscsConfigData data type**

Attribute name	Data type	Cardinality	Description
mscsType	Enum (inlined)	1	The type of connectivity that is requested to be provided to the virtualised networks in the NFVI-PoP and characterizes the connectivity service across the WAN.  Permitted values: - L1 - L2 - L3
siteAccessProtectionSchemes	Enum (inlined)	0..N	Information to determine the proper MSCS endpoints configuration to fulfil certain resiliency/protection requirements, e.g. by considering certain availability and redundancy of connectivity service endpoints in between NFVI-PoP and WAN.

Attribute name	Data type	Cardinality	Description
>locationConstraints	LocationConstraints	1	Defines the location constraints according to an NFVI-PoP (site).
>protectionScheme	Enum (inlined)	1	Defines the protection scheme.  Permitted values: - UNPROTECTED - ONE_TO_ONE - ONE_PLUS_ONE - ONE_TO_N
mtuMscs	Number	0..1	Maximum Transmission Unit (MTU) that can be forwarded over the MSCS (in bytes). Default value is "1500" (bytes).
wanLayer2ProtocolData	WanLayer2ProtocolData	0..1	Layer 2 protocol specific information for the configuration of the multi-site connectivity service over the WAN.
wanLayer3ProtocolData	WanLayer3ProtocolData	0..1	Layer 3 protocol specific information for the configuration of the multi-site connectivity service over the WAN.

### 6.5.3.88 Type: WanLayer2ProtocolData

This type provides information about Layer 2 protocol specific information for the configuration of the MSCS over the WAN. It shall comply with the provisions defined in table 6.5.3.88-1.

**Table 6.5.3.88-1: Definition of the WanLayer2ProtocolData data type**

Attribute name	Data type	Cardinality	Description
mcsLayer2Protocol	Enum (inlined)	0..1	Type of underlying connectivity service and protocol associated to the type of MSCS.  Permitted values are as listed below and restricted by the type of MSCS: - EVPN_BGP/MPLS: as specified in IETF RFC 7432 [21]. - EVPN_VPWS: as specified in IETF RFC 8214 [30]. - VPLS_BGP: as specified in IETF RFC 4761 [22] and IETF RFC 6624 [23]. - VPLS_LDP_L2TP: as specified in IETF RFC 4762 [24] and IETF RFC 6074 [25]. - VPWS_LDP_L2TP: as specified in IETF RFC 6074 [25].
isSegmentPreservation	Boolean	1	Indicates the requirement of whether to ensure network segment (e.g. VLAN id) preservation across the MSCS endpoints (i.e. from/to the NFVI-PoPs). If "TRUE", segment identifiers shall be preserved, "FALSE" otherwise. Default value is "FALSE".
isSegmentCosPreservation	Boolean	1	Indicates the requirement of whether to ensure network segment class of service preservation across the MSCS endpoints (i.e. from/to the NFVI-PoPs). If "TRUE", segment class of service shall be preserved, "FALSE" otherwise. Default value is "FALSE".

### 6.5.3.89 Type: WanLayer3ProtocolData

This type provides information about Layer 3 protocol specific information for the configuration of the MSCS over the WAN. It shall comply with the provisions defined in table 6.5.3.89-1.

**Table 6.5.3.89-1: Definition of the WanLayer3ProtocolData data type**

Attribute name	Data type	Cardinality	Description
mcsLayer3Protocol	Enum (inlined)	0..1	Type of underlying connectivity service and protocol associated to the type of MSCS.  Permitted values are as listed below and restricted by the type of MSCS: - BGP_IP_VPN: BGP/MPLS based IP VPN as specified in IETF RFC 4364 [26].

### 6.5.3.90 Type: WanConnectionInfo

This type provides information about the connectivity to the WAN of network resources realizing a VL, e.g. when the VL is deployed on several sites across a WAN. It shall comply with the provisions defined in table 6.5.3.90-1.

**Table 6.5.3.90-1: Definition of the WanConnectionInfo data type**

Attribute name	Data type	Cardinality	Description
wanConnectionInfo	IdentifierInNs	1	Identifies this WAN connection information.
nsVirtualLinkInfo	IdentifierInNs	0..1	References the NS VL instance to which the connection information is associated. Shall be present if a corresponding NS VL instance has been created. See note.
vnfVirtualLinkResourceInfo	IdentifierInVnf	0..1	References the VNF VL instance to which the connection information is associated. Shall be present if a corresponding VNF VL instance has been created. See note.
protocolInfo	WanConnectionProtocolInfo	0..1	Protocol specific information for connecting to the WAN.
NOTE: Either a "nsVirtualLinkInfo" or a "vnfVirtualLinkResourceInfo" shall be provided, but not both.			

### 6.5.3.91 Type: WanConnectionProtocolInfo

This type provides protocol specific information about the connectivity to the WAN of network resources realizing a VL, e.g. when the VL is deployed on several sites and across a WAN, and the related multi-site connectivity service (MSCS) enabling the connectivity through the WAN. This type supports providing information about both pre-provisioned WAN connectivity realized by external entities to NFV-MANO but parts of such connectivity is known to the NFVO, as well as information about MSCS created under NFV-MANO responsibility (i.e. connectivity is realized when NFVO communicates with the WIM). It shall comply with the provisions defined in table 6.5.3.91-1.

**Table 6.5.3.91-1: Definition of the WanConnectionProtocolInfo data type**

Attribute name	Data type	Cardinality	Description
mcsInfo	McsInfo	0..1	Information about the provisioned MSCS, if already available.
connectivityServiceEndpoints	ConnectivityServiceEndpointInfo	0..N	Configuration information about the network resources in the NFVI-PoP and their connectivity to the WAN.

### 6.5.3.92 Type: ModifyWanConnectionInfoData

This type represents attribute modifications for WAN connection information. The attributes of the "WanConnectionInfo" that can be modified according to the provisions of the "UpdateNsRequest" in clause 6.5.2.12 related to WAN connection information are included in the "ModifyWanConnectionInfoData" data type. It shall comply with the provisions defined in table 6.5.3.92-1.

**Table 6.5.3.92-1: Definition of the ModifyWanConnectionInfoData data type**

Attribute name	Data type	Cardinality	Description
wanConnectionInfoId	Identifier	1	Identifier of the WAN connection information to modify.
mcsName	String	0..1	New value of the "mcsName" attribute in "MscsInfo", or "null" to remove the attribute.
mcsDescription	String	0..1	New value of the "mcsDescription" attribute in "MscsInfo", or "null" to remove the attribute.
mcsEndpoints	MscsEndpointInfo	0..N	New content of certain entries of MSCS endpoints in the "mcsEndpoints" attribute in "MscsInfo", as defined below this table.
removeMscsEndpointIds	Identifier	0..N	List of identifier entries to be deleted from the "mcsEndpoints" attribute array in "MscsInfo", to be used as "deleteIdList" as defined below this table.
connectivityServiceEndpoints	ConnectivityServiceEndpointInfo	0..N	New content of certain entries of connectivity service endpoints in the "connectivityServiceEndpointInfos" attribute in "WanConnectionProtocolInfo", as defined below this table.
removeConnectivityServiceEndpoints	Identifier	0..N	List of identifier entries to be deleted from the "connectivityServiceEndpointInfos" attribute array in "WanConnectionProtocolInfo", to be used as "deleteIdList" as defined below this table.

The following provisions shall apply when modifying an attribute that is an array of objects of type "MscsEndpointInfo" and "ConnectivityServiceEndpointInfo".

Assumptions:

- A) "oldList" is the array to be modified, "newList" is the array that contains the changes and "deleteIdList" is the array that contains the identifiers of those "oldList" entries to be deleted.
- B) "oldEntry" is an entry in "oldList" and "newEntry" is an entry in "newList".
- C) A "newEntry" has a "corresponding entry" if there exists an "oldEntry" that has the same content of "id" attribute as the "newEntry"; a "newEntry" has no corresponding entry if no such "oldEntry" exists.
- D) In any array of objects, the content of "id" is unique (i.e. there are no two entries with the same content of "id").

Provisions:

- 1) For each "newEntry" in "newList" that has no corresponding entry in "oldList", the "oldList" array shall be modified by adding that "newEntry".
- 2) For each "newEntry" in "newList" that has a corresponding "oldEntry" in "oldList", the value of "oldEntry" shall be replaced by the value of "newEntry".
- 3) For each entry in "deleteIdList", delete the entry in "oldList" that has the same content of the "id" attribute as the entry in "deleteIdList".

### 6.5.3.93 Type: WanConnectionInfoModification

This type represents attribute modifications that were performed on WAN connection information. The attributes that can be included consist of those requested to be modified explicitly with the "UpdateNsRequest" data structure. It shall comply with the provisions defined in table 6.5.3.93-1.

**Table 6.5.3.93-1: Definition of the WanConnectionInfoModification data type**

Attribute name	Data type	Cardinality	Description
wanConnectionInfoId	Identifier	1	Identifier of the WAN connection information that has been modified.
mscsName	String	0..1	If present, this attribute signals modifications of the "mcsName" attribute in "MscsInfo" as defined in clause 6.5.3.82.
mcsDescription	String	0..1	If present, this attribute signals modifications of the "mcsDescription" attribute in "MscsInfo" as defined in clause 6.5.3.82.
mcsEndpoints	MscsEndpointInfo	0..N	If present, this attribute signals modifications of certain entries in the "mcsEndpoints" attribute array in "MscsInfo", as defined in clause 6.5.3.82.
removeMcsEndpointIds	Identifier	0..N	If present, this attribute signals the deletion of certain entries in the "mcsEndpoints" attribute array in "MscsInfo", as defined in clause 6.5.3.82.
connectivityServiceEndpoints	ConnectivityServiceEndpointInfo	0..N	If present, this attribute signals modifications of certain entries in the "connectivityServiceEndpoints" attribute array in "WanConnectionProtocolInfo", as defined in clause 6.5.3.91.
removeConnectivityServiceEndpoints	Identifier	0..N	If present, this attribute signals the deletion of certain entries in the "connectivityServiceEndpoints" attribute array in "WanConnectionProtocolInfo", as defined in clause 6.5.3.91.

#### 6.5.3.94 Type: VirtualStorageResourceInfo

This type represents the information that allows addressing a virtualised resource that is used by a VNF instance. It shall comply with the provisions defined in table 6.5.3.94-1.

**Table 6.5.3.94-1: Definition of the VirtualStorageResourceInfo data type**

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of this VirtualStorageResourceInfo instance.
virtualStorageDescId	IdentifierInVnfd	1	Identifier of the VirtualStorageDesc in the VNFD.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure).
storageResource	ResourceHandle	1	Reference to the VirtualStorage resource or reference to a Storage MCIO.
reservationId	Identifier	0..1	The reservation identifier applicable to the resource. It shall be present when an applicable reservation exists.
metadata	KeyValuePairs	0..1	Metadata about this resource.

#### 6.5.3.95 Type: AddNsVirtualLinkData

This type specifies the parameters used for the creation of a new NsVirtualLink instance. It shall comply with the provisions defined in table 6.5.3.95-1.

**Table 6.5.3.95-1: Definition of the AddNsVirtualLinkData data type**

Attribute name	Data type	Cardinality	Description
nsVirtualLinkProfileId	Identifier	1	Identifier of the virtual link profile to be used to create a new NsVirtualLink instance. See note.
NOTE: All NsVirtualLink instances of a particular NS DF based on a specific "NsVirtualLinkDesc" have the same characteristics as they use the same "VirtualLinkProfile".			

### 6.5.3.96 Type: NestedNsLocationConstraint

This type represents the association of location constraints to a nested NS instance to be created according to a specific NS profile. It shall comply with the provisions defined in table 6.5.3.96-1.

**Table 6.5.3.96-1: Definition of the NestedNsLocationConstraint data type**

Attribute name	Data type	Cardinality	Description
nsProfileId	IdentifierInNsd	1	Identifier of an NsProfile defined in the NSD which the existing nested NS instance shall be matched with.
locationConstraints	LocationConstraints	1	Defines the location constraints for the nested NS instance to be created based on the NS profile. See note.
NOTE: These constraints are typically determined by the OSS/BSS from service requirements (e.g. latency requirements, regulatory requirements). The NFVO can map such location constraints to eligible NFVI-PoPs/resource zones where the VNF instances as part of the nested NS are to be created.			

### 6.5.3.97 Type: VipCpInfo

This type provides information related to virtual IP (VIP) CP. It shall comply with the provisions defined in table 6.5.3.97-1.

**Table 6.5.3.97-1: Definition of the VipCpInfo data type**

Attribute name	Data type	Cardinality	Description
cpInstanceId	IdentifierInVnf	1	Identifier of this VIP CP instance and of this VipCpInfo information element.
cpId	IdentifierInVnfd	1	Identifier of the VIP Connection Point Descriptor, VipCpd, in the VNFD.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure).
vnfExtCpId	IdentifierInVnf	0..1	When the VIP CP is exposed as external CP of the VNF, the identifier of this external VNF CP instance.
cpProtocolInfo	CpProtocolInfo	0..N	Protocol information for this CP. There shall be one cpProtocolInfo for layer 3. There may be one cpProtocolInfo for layer 2.
associatedVnfcCpIds	IdentifierInVnf	0..N	Identifiers of the VnfcCps that share the virtual IP address allocated to the VIP CP instance. See note.
vnfLinkPortId	IdentifierInVnf	0..1	Identifier of the "VnfLinkPortInfo" structure in the "VnfVirtualLinkResourceInfo" or "ExtManagedVirtualLinkInfo" structure. Shall be present if the CP is associated to a link port on an internal VL (including externally-managed internal VL).
metadata	KeyValuePair	0..N	Metadata about this VIP CP.
NOTE: It is possible that there is no associated VnfcCp because the VIP CP is available but not associated yet.			

### 6.5.3.98 Type: TerminateNsData

This type represents the information used to terminate NS. It shall comply with the provisions defined in table 6.5.3.98-1.

**Table 6.5.3.98-1: Definition of the TerminateNsData data type**

Attribute name	Data type	Cardinality	Description
additionalParamsforNs	KeyValuePairs	0..1	Allows the OSS/BSS to provide additional parameter(s) to the termination process at the NS level.

### 6.5.3.99 Type: TerminateVnfData

This type represents the information to terminate a VNF that is part of an NS. It shall comply with the provisions defined in table 6.5.3.99-1.

**Table 6.5.3.99-1: Definition of the TerminateVnfData data type**

Attribute name	Data type	Cardinality	Description
vnfInstanceId	Identifier	1	Identifies the VNF instance, part of the NS, to be terminated.
terminationType	Enum (inlined)	0..1	Indicates whether forceful or graceful termination is requested. See note.  Permitted values: - FORCEFUL - GRACEFUL
gracefulTerminationTimeout	Integer	0..1	The attribute is only applicable in case of graceful termination. It defines the time to wait for the VNF to be taken out of service before shutting down the VNF and releasing the resources. The unit is seconds.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the OSS/BSS as input to the termination process, specific to the VNF being terminated.  EXAMPLE:       Input parameters to VNF-specific termination procedures.
NOTE:    If the VNF is still in service, requesting forceful termination can adversely impact network service.			

### 6.5.3.100 Type: DataFlowMirroringData

This type represents the information for the data flow mirroring to be associated to this NS instance. It shall comply with the provisions defined in table 6.5.3.100-1.

**Table 6.5.3.100-1: Definition of the DataFlowMirroringData data type**

Attribute name	Data type	Cardinality	Description
mirroringId	Identifier	1	Identifier of this DataFlowMirroringData.
mirroringName	String	1	Name of Data Flow Mirroring.
description	String	1	Information description of Data Flow Mirroring.
collectorDetails	CollectorDetails	1	Information about where the mirrored flow is to be delivered.
dataFlowData	DataFlowData	1..N	Information about the data flows to be mirrored.

### 6.5.3.101 Type: DataFlowData

This type represents the information about the data flows to be mirrored. It shall comply with the provisions defined in table 6.5.3.101-1.

**Table 6.5.3.101-1: Definition of the DataFlowData data type**

Attribute name	Data type	Cardinality	Description
dataFlowInfold	IdentifierInNsd	0..1	Identifier of the Data flow information in NSD. It may be provided if a corresponding data flow has been declared in the NSD. See note.
dataFlowDetails	DataFlowDetails	1	Detailed information about the data flows that are requested to be mirrored.
NOTE:    A value need not be provided at runtime if the API consumer does not intend to refer to a data flow defined in the NSD.			

### 6.5.3.102 Type: DataFlowDetails

This type represents the detailed information about the data flows that are requested to be mirrored. It shall comply with the provisions defined in table 6.5.3.102-1.

**Table 6.5.3.102-1: Definition of the DataFlowDetails data type**

Attribute name	Data type	Cardinality	Description
description	String	0..1	Information description of the data flow.
cpdId	IdentifierInNsd	0..1	Identifies the descriptor of an external connection point on a VNF/PNF or a SAP which acts as the mirroring point of the data flow to be mirrored. See note 1.
direction	DirectionType	0..1	The direction of the data flow that are requested to be mirrored. See note 1.
targetIpAddress	IpAddress	0..1	If present, only the data flow to or/and from the target IP address shall be mirrored. See note 2.
NOTE 1: It shall be present when dataFlowInfold in DataFlowData is not provided.			
NOTE 2: Depends on the value of the direction of the data flow to be mirrored, this attribute defines whether the data flow from the connection point to the target IP address or the data flow from the target IP address to the connection point or both shall be mirrored.			

### 6.5.3.103 Type: CollectorDetails

This type represents the information about where the mirrored flow is to be delivered. It shall comply with the provisions defined in table 6.5.3.103-1.

**Table 6.5.3.103-1: Definition of the CollectorDetails data type**

Attribute name	Data type	Cardinality	Description
collectorId	Identifier	1	Identifier of the collector where the mirrored flow is to be delivered.
collectorName	String	1	Name of the collector where the mirrored flow is to be delivered.
portId	IdentifierInVim	0..1	Identifier of the collector port managed by VIM where the mirrored flow is to be delivered. See note.
NOTE: The collector which is attached to this port can be a virtual machine that uses the mirrored data for analysis purpose. The collector is not managed by VIM.			

### 6.5.3.104 Type: CreateDataFlowMirroring

This type represents the information for the data flow mirroring to be associated to this NS instance. It shall comply with the provisions defined in table 6.5.3.104-1.

**Table 6.5.3.104-1: Definition of the CreateDataFlowMirroring data type**

Attribute name	Data type	Cardinality	Description
dataFlowMirroringData	DataFlowMirroringData	1	Information for the data flow mirroring to be associated to this NS instance.
vnfInstanceId	Identifier	0..1	Identifier of the VNF instance from where the data flows are requested to be mirrored.
cpInstanceId	Identifier	0..1	Identifier of the CP instance from where the data flows are requested to be mirrored.

### 6.5.3.105 Type: UpdateDataFlowMirroring

This type represents the information on the data flow mirroring associated to the NS instance that is being modified. It shall comply with the provisions defined in table 6.5.3.105-1.



**Table 6.5.3.105-1: Definition of the UpdateDataFlowMirroring data type**

Attribute name	Data type	Cardinality	Description
mirroringId	Identifier	1	Identifier of the Data Flow Mirroring to be updated.
mirroringName	String	0..1	Name of Data Flow Mirroring. See note.
description	String	0..1	Information description of Data Flow Mirroring. See note.
collectorDetails	CollectorDetails	0..1	New information about the destination where the mirrored flow is to be delivered. See note.
NOTE: Updating Data Flow Mirroring can involve one or more following cases: 1) updating the name of data flow mirroring, related to "mirroringName"; 2) updating information description of data flow mirroring, related to "description"; 3) updating the definition about where the mirrored data flow is requested to be delivered, related to "collectorDetails".			

### 6.5.3.106 Type: MirroringInfo

This type represents the runtime information that the NFVO holds about a Data Flow Mirroring. It shall comply with the provisions defined in table 6.5.3.106-1.

**Table 6.5.3.106-1: Definition of the MirroringInfo type**

Attribute name	Data type	Cardinality	Description
mirroringId	Identifier	1	Unique identifier of the Data Flow Mirroring. The identifier is assigned by the NFVO. See note.
mirroringName	String	1	Name of Data Flow Mirroring.
description	String	1	Information description of Data Flow Mirroring.
collectorDetails	CollectorDetails	1	Information about where the mirrored flow is to be delivered.
vnfInstanceId	Identifier	1..N	Identifier of the VNF instance from where the data flows are requested to be mirrored.
cpInstanceId	Identifier	1..N	Identifier of the CP instance from where the data flows are mirrored.
dataFlowData	DataFlowData	1..N	Information about the data flows to be mirrored.
NOTE: It is responsibility of the NFVO to map the mirroringIds exposed towards the API consumer on the Os-Ma-nfvo reference point with the corresponding identifiers of the data flow mirroring that are created towards the VIM.			

### 6.5.3.107 Type: McioInfo

This type provides information about an MCIO representing the set of VNFC instances realized by one or a set of OS containers which have been created based on the same VDU.

Within the CISM, an MCIO controller monitors the actual state of an MCIO representing the set of VNFC instances realized by one or a set of OS containers and compare it to the desired state as specified in the respective declarative descriptor. It triggers actions toward the CIS to align the actual to the desired state. Monitoring the actual state includes monitoring the number of MCIO instances available at any specific point in time. In addition, an MCIO controller maintains properties and runtime information on the MCIO instances which have been created based on the same VDU. The McioInfo data structure provides the runtime information on the MCIOs obtained from the MCIO controller.

NOTE: There are different types of MCIOs. The set of VNFC instances based on the same VDU is represented by one MCIO, e.g. of type Deployment. Each individual VNFC instance is represented by another type of MCIO, e.g. a POD.

Runtime information of the set of OS containers realizing an individual VNFC instance is not part of the McioInfo data structure; such runtime information is provided in the ResourceHandle data structure referenced from the VnfcResourceInfo. The McioInfo does not provide runtime information of a constituent VNFC instance created based on a specific VDU.

It shall comply with the provisions defined in table 6.5.3.107-1.

**Table 6.5.3.107-1: Definition of the McioInfo data type**

Attribute name	Data type	Cardinality	Description
mcioId	Identifier	1	Identifier of this MCIO, created by the CISM. See note 3.
mcioName	String	1	Human readable name of this MCIO. See note 4.
mcioNamespace	String	1	Namespace of this MCIO.
vduld	IdentifierInVnfd	1	Reference to the related VDU in the VNFD applicable to this resource.
cismId	Identifier	1	Identifier of the CISM managing this MCIO.
mcioType	String	1	The type of MCIO. Specific values, their semantics and associated MCIO types are defined in clause 6.5.4.13. Additional values are also permitted. See note 1.
desiredInstances	Integer	1	Number of desired MCIO instances.
availableInstances	Integer	1	Number of available MCIO instances.
additionalInfo	KeyValuePairs	0..1	Additional information which is specific to the MCIO, its type, and which is available from the CISM. See note 2.
<p>NOTE 1: The type of MCIO as specified in the declarative descriptor of the MCIO, and that can be read from the CISM.</p> <p>EXAMPLE 1: In case of MCIOs managed by Kubernetes® (see in [i.12]), the type of MCIO corresponds to the "kind" property of the declarative descriptor.</p> <p>NOTE 2: If the attribute additionalInfo is present, it may contain runtime information on the actual and desired state of the MCIO(s).</p> <p>NOTE 3: When the container infrastructure service is a Kubernetes® instance, the mcioId is the combined values from the kind and name fields of the Kubernetes® resource object, separated by a slash.</p> <p>EXAMPLE 2: "Deployment/abcd".</p> <p>NOTE 4: When the container infrastructure service is a Kubernetes® instance, the mcioName is the name field of the resource object.</p>			

### 6.5.3.108 Type: VirtualCplInfo

This type represents the information related to a virtual CP instance of a VNF. It shall comply with the provisions defined in table 6.5.3.108-1.

**Table 6.5.3.108-1: Definition of the VirtualCplInfo data type**

Attribute name	Data type	Cardinality	Description
cplInstanceId	IdentifierInVnf	1	Identifier of this virtual CP instance.
cpdId	IdentifierInVnfd	1	Identifier of the VirtualCpd in the VNFD.
resourceHandle	ResourceHandle	1	Reference to the virtualised resource realizing this virtual CP.
vnfExtCplId	IdentifierInVnf	0..1	When the virtual CP is exposed as external CP of the VNF, the identifier of this external VNF CP instance.
cpProtocolInfo	CpProtocolInfo	0..N	Protocol information for this CP. There shall be one cpProtocolInfo for each layer protocol supported.
vduld	IdentifierInVnfd	1..N	Reference to the VDU(s) which implement the service accessible via the virtual CP instance. See note.
additionalServiceInfo	AdditionalServiceInfo	0..N	Additional service identification information of the virtual CP instance.
metadata	KeyValuePair	0..1	Metadata about this virtual CP instance.
<p>NOTE: A consumer of the VNF LCM interface can learn the actual VNFC instances implementing the service accessible via the virtual CP instance by querying the "vnfcResourceInfo" from the "InstantiatedVnfInfo" and filtering by corresponding "vduld" values.</p>			

### 6.5.3.109 Type: AdditionalServiceInfo

This type represents the additional service information of the virtual CP instance used to expose properties of the virtual CP to NFV-MANO. It shall comply with the provisions defined in table 6.5.3.109-1.

**Table 6.5.3.109-1: Definition of the AdditionalServiceInfo data type**

Attribute name	Data type	Cardinality	Description
portInfo	ServicePortInfo	1..N	Service port numbers exposed by the virtual CP instance.
serviceInfo	KeyValuePairs	0..1	Service matching information exposed by the virtual CP instance. See note.
NOTE: This attribute shall only be present if additional information is needed to identify the service termination within the VNF, such as for example a URL path information in an HTTP request required to allow a single virtual CP IP address to be used for several HTTP based services that use the same port number.			

### 6.5.3.110 Type: ServicePortInfo

This type represents the information related to the service identifying port properties exposed by the virtual CP instance. It shall comply with the provisions defined in table 6.5.3.110-1.

**Table 6.5.3.110-1: Definition of the ServicePortInfo data type**

Attribute name	Data type	Cardinality	Description
name	String	1	The name of the port exposed by the virtual CP instance.
protocol	Enum (inlined)	1	The L4 protocol for this port exposed by the virtual CP instance.  Permitted values: - TCP - UDP - SCTP
port	Integer	1	The L4 port number exposed by the virtual CP instance.
isPortConfigurable	Boolean	1	Specifies whether the port attribute value is allowed to be configurable.

### 6.5.3.111 Type: OverridingVersionDependency

This type represents the information to override or add a version dependency in the runtime information that the NFVO keeps about a profile of a NSD constituent. It shall comply with the provisions defined in table 6.5.3.111-1.

**Table 6.5.3.111-1: Definition of the OverridingVersionDependency data type**

Attribute name	Data type	Cardinality	Description
profileId	IdentifierInNsd	1	Reference to a vnfProfile, nsProfile or pnfProfile where the versionDependency overrides an existing versionDependency or is added to the existing version dependencies. See note.
versionDependency	VersionDependency	1	Describes version dependencies for a specific NSD constituent.
NOTE: If the referenced profile contains a versionDependency with a dependentConstituent equal to the one indicated in the versionDependency attribute of this information element, the versionDependency in this information element overrides the one in the profile, otherwise it is added to the version dependencies in the profile.			

### 6.5.3.112 Type: VersionDependency

This type represents the information related to all dependencies that an NSD constituent has on the versions of other NSD constituents. It shall comply with the provisions defined in table 6.5.3.112-1.

**Table 6.5.3.112-1: Definition of the VersionDependency data type**

Attribute name	Data type	Cardinality	Description
dependentConstituentId	IdentifierInNsd	1	Identifier of the NSD constituent (vnfdId, nsdId or pnfdId) which has version dependencies on other NSD constituents.
versionDependencyStatement	VersionDependencyStatement	0..N	Identifies one or multiple versions of an NSD constituents upon which the dependent constituent identified by dependentConstituentId has a dependency. Cardinality 0 is used to remove an existing version dependency in a profile.

### 6.5.3.113 Type: VersionDependencyStatement

This type represents the information related to one or more VNF, NS or PNF descriptor identifiers which describe one single dependency. When more than one descriptor is indicated, they correspond to different versions of the same VNF, NS or PNF and they represent alternatives to fulfil the dependency. It shall comply with the provisions defined in table 6.5.3.113-1.

**Table 6.5.3.113-1: Definition of the VersionDependencyStatement data type**

Attribute name	Data type	Cardinality	Description
descriptorId	IdentifierInNsd	1..N	Identifies a VNFD, NSD or PNFD upon which the entity using this information element depends.  When more than one descriptor is indicated, they shall correspond to versions of the same VNF, NS or PNF and they represent alternatives, i.e. the presence of one of them fulfills the dependency.

### 6.5.3.114 Type: NetAttDefResourceInfo

This type represents the information related to a network attachment definition resource that provides the specification of the interface used to connect one or multiple connection points to a secondary container cluster network. It shall comply with the provisions defined in table 6.5.3.114-1.

**Table 6.5.3.114-1: Definition of the NetAttDefResourceInfo data type**

Attribute name	Data type	Cardinality	Description
netAttDefResourceInfoId	Identifier	1	Identifier of this network attachment definition resource as provided by the entity that has created it.
netAttDefResource	ResourceHandle	1	Resource handle of the resource in the scope of the CISM.
associatedExtCpId	IdentifierInVnf	0..N	Identifier of the external CP of the VNF associated to this network attachment definition resource. Shall be present when the network attachment definition resource is used for external connectivity by the VNF.
associatedVnfcCpId	IdentifierInVnf	0..N	Identifier of the VNFC CP of the VNF associated to this network attachment definition resource. May be present when the network attachment definition resource is used for internal connectivity by the VNF.

### 6.5.3.115 Type: NetAttDefResourceData

This type represents the information related to a network attachment definition resource that provides the specification of the interface to be used to connect one or multiple connection points to a secondary container cluster network realizing a VL. It shall comply with the provisions defined in table 6.5.3.115-1.

**Table 6.5.3.115-1: Definition of the NetAttDefResourceData data type**

Attribute name	Data type	Cardinality	Description
netAttDefResourceId	Identifier	1	Identifier of this network attachment definition resource as provided by the entity that has created it.
resourceHandle	ResourceHandle	1	Resource handle of the resource identifying the network attachment definition resource that provides the specification of the interface to attach the connection points to a secondary container cluster network.

### 6.5.3.116 Type: IntVnfCpData

This type represents the input information related to one or more VNF internal CP instances created based on the same CPD. It shall comply with the provisions defined in table 6.5.3.116-1.

**Table 6.5.3.116-1: Definition of the IntVnfCpData data type**

Attribute name	Data type	Cardinality	Description
cpdId	IdentifierInVnfd	1	Identifier of the CPD in the VNFD.
netAttDefResourceId	IdentifierInVnf	1..N	Identifier of the "NetAttDefResourceData" structure that provides the specification of the interface to attach the VNF internal CP created from the CPD identified by cpdId to a secondary container cluster network. See note.
NOTE:	Cardinality greater than 1 is only applicable for specific cases where more than one network attachment definition resource is needed to fulfil the connectivity requirements of the VNF internal CP, e.g. to build a link redundant mated pair in SR-IOV cases. When more than one "netAttDefResourceId" is indicated, all shall belong to the same namespace as defined by the corresponding "netAttDefResourceNamespace" attribute in the "NetAttDefResourceData".		

## 6.5.4 Referenced simple data types and enumerations

### 6.5.4.1 Introduction

This clause defines simple data types that can be referenced from data structures defined in the previous clauses.

### 6.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.2.

### 6.5.4.3 Enumeration: NsLcmOpType

The enumeration NsLcmOpType represents those lifecycle operations that trigger an NS lifecycle management operation occurrence notification. It shall comply with the provisions defined in table 6.5.4.3-1.

**Table 6.5.4.3-1: Enumeration NsLcmOpType**

Enumeration value	Description
INSTANTIATE	Represents the "Instantiate NS" LCM operation.
SCALE	Represents the "Scale NS" LCM operation.
UPDATE	Represents the "Update NS" LCM operation.
TERMINATE	Represents the "Terminate NS" LCM operation.
HEAL	Represents the "Heal NS" LCM operation.

#### 6.5.4.4 Enumeration: NsLcmOperationStateType

The enumeration NsLcmOperationStateType shall comply with the provisions defined in table 6.5.4.4-1. More information of the meaning of the states can be found in clause 6.6.2.2.

**Table 6.5.4.4-1: Enumeration NsLcmOperationStateType**

Enumeration value	Description
PROCESSING	The NS LCM operation is currently in execution.
COMPLETED	The NS LCM operation has been completed successfully.
PARTIALLY_COMPLETED	The NS LCM operation has been partially completed with acceptable errors.
FAILED_TEMP	The NS LCM operation has failed and execution has stopped, but the execution of the operation is not considered to be closed.
FAILED	The NS LCM operation has failed and it cannot be retried or rolled back, as it is determined that such action will not succeed.
ROLLING_BACK	The NS LCM operation is currently being rolled back.
ROLLED_BACK	The NS LCM operation has been successfully rolled back, i.e. The state of the NS prior to the original operation invocation has been restored as closely as possible.

#### 6.5.4.5 Enumeration: NsComponentType

The enumeration NsComponentType represents the NS component type. It shall comply with the provisions defined in table 6.5.4.5-1.

**Table 6.5.4.5-1: Enumeration NsComponentType**

Enumeration value	Description
VNF	Represents the impacted NS component is a VNF.
PNF	Represents the impacted NS component is a PNF.
NS	Represents the impacted NS component is a nested NS.

#### 6.5.4.6 Enumeration: LcmOpNameForChangeNotificationType

The enumeration LcmOpNameForChangeNotificationType represents the name of the lifecycle operation that impacts the NS component and trigger an NS change notification. It shall comply with the provisions defined in table 6.5.4.6-1.

**Table 6.5.4.6-1: Enumeration LcmOpNameForChangeNotificationType**

Enumeration value	Description
VNF_INSTANTIATE	Represents the "Instantiate VNF" LCM operation.
VNF_SCALE	Represents the "Scale VNF" LCM operation.
VNF_SCALE_TO_LEVEL	Represents the "Scale VNF to Level" LCM operation.
VNF_CHANGE_FLAVOUR	Represents the "Change VNF Flavour" LCM operation.
VNF_TERMINATE	Represents the "Terminate VNF" LCM operation.
VNF_HEAL	Represents the "Heal VNF" LCM operation.
VNF_OPERATE	Represents the "Operate VNF" LCM operation.
VNF_CHANGE_EXT_CONN	Represents the "Change external VNF connectivity" LCM operation.
VNF_MODIFY_INFO	Represents the "Modify VNF Information" LCM operation.
VNF_CREATE_SNAPSHOT	Represents the "Create VNF Snapshot" LCM operation.
VNF_REVERT_TO_SNAPSHOT	Represents the "Revert To VNF Snapshot" LCM operation.
VNF_CHANGE_VNFPKG	Represents the "Change current VNF package" LCM operation.
NS_INSTANTIATE	Represents the "Instantiate NS" LCM operation.
NS_SCALE	Represents the "Scale NS" LCM operation.
NS_UPDATE	Represents the "Update NS" LCM operation.
NS_TERMINATE	Represents the "Terminate NS" LCM operation.
NS_HEAL	Represents the "Heal NS" LCM operation.

#### 6.5.4.7 Enumeration: LcmOpOccStatusForChangeNotificationType

The enumeration LcmOpOccStatusForChangeNotificationType represents the status of the lifecycle management operation occurrence that impacts the NS component and triggers an NS change notification. It shall comply with the provisions defined in table 6.5.4.7-1.

**Table 6.5.4.7-1: Enumeration LcmOpOccStatusForChangeNotificationType**

Enumeration value	Description
START	The impact on the NS component is identified.
COMPLETED	The impact on the NS component stops and related lifecycle operation completes successfully.
PARTIALLY_COMPLETED	The impact on the NS component stops and related lifecycle operation partially completes. Inconsistency state may exist on the NS component.
FAILED	The impact on the NS component stops and related lifecycle operation fails. Inconsistency state may exist for the NS component.
ROLLED_BACK	The impact on the NS component stops and related lifecycle operation is rolled back.

#### 6.5.4.8 Enumeration: OperationalStates

The enumeration OperationalStates shall comply with the provisions defined in table 6.5.4.8-1.

**Table 6.5.4.8-1: Enumeration OperationalStates**

Enumeration value	Description
STARTED	The VNF instance is up and running.
STOPPED	The VNF instance has been shut down.

#### 6.5.4.9 Enumeration: StopType

The enumeration StopType shall comply with the provisions defined in table 6.5.4.9-1.

**Table 6.5.4.9-1: Enumeration StopType**

Enumeration value	Description
FORCEFUL	The VNFM will stop the VNF immediately after accepting the request.
GRACEFUL	The VNFM will first arrange to take the VNF out of service after accepting the request. Once that operation is successful or once the timer value specified in the "gracefulStopTimeout" attribute expires, the VNFM will stop the VNF.

#### 6.5.4.10 Enumeration: CancelModeType

The enumeration CancelModeType defines the valid modes of cancelling an NS LCM operation occurrence. It shall comply with the provisions defined in table 6.5.4.10-1.

**Table 6.5.4.10-1: Enumeration CancelModeType**

Enumeration value	Description
GRACEFUL	The NFVO shall not start any new VNF lifecycle management and resource management operation, and shall wait for the ongoing VNF lifecycle management and resource management operations in the underlying system, typically the VNFM and VIM, to finish execution or to time out. After that, the NFVO shall put the operation occurrence into the FAILED_TEMP state.
FORCEFUL	The NFVO shall not start any new VNF lifecycle management and resource management operation, shall cancel the ongoing VNF lifecycle management and resource management operations in the underlying system, typically the VNFM and VIM, and shall wait for the cancellation to finish or to time out. After that, the NFVO shall put the operation occurrence into the FAILED_TEMP state.

### 6.5.4.11 Enumeration: LcmOpOccNotificationVerbosityType

The enumeration LcmOpOccNotificationVerbosityType provides values to control the verbosity of LCM operation occurrence notifications. It shall comply with the provisions defined in table 6.5.4.11-1.

**Table 6.5.4.11-1: Enumeration LcmOpOccNotificationVerbosityType**

Enumeration value	Description
FULL	This signals a full notification which contains all change details.
SHORT	This signals a short notification which omits large-volume change details to reduce the size of data to be sent via the notification mechanism.

### 6.5.4.12 Enumeration: DirectionType

The enumeration DirectionType represents the direction of the data flow that are requested to be mirrored. It shall comply with the provisions defined in table 6.5.4.12-1.

**Table 6.5.4.12-1: Enumeration DirectionType**

Enumeration value	Description
IN	Represents the inbound data flow to the target connection point that needs to be mirrored.
OUT	Represents the outbound data flow from the target connection point that needs to be mirrored.
BOTH	Represents the inbound and outbound data flows to/from the target connection point and both need to be mirrored.

### 6.5.4.13 Type: McioTypeName

The present type definition provides valid String values for the "mcioType" attribute of the "McioInfo" data type, and the valid values are defined in table 6.5.4.13-1.

**Table 6.5.4.13-1: String values for MCIO types**

String value	Description
"Deployment"	Represents the Deployment compute MCIO as defined in ETSI GS NFV-SOL 018 [i.13].
"Statefulset"	Represents the StatefulSet compute MCIO as defined in ETSI GS NFV-SOL 018 [i.13].

### 6.5.4.14 Enumeration: FeasibilityCheckReserveType

The enumeration FeasibilityCheckReserveType defines the permitted values to represent the feasibility check and reserve options applicable for the NS LCM operations. It shall comply with the provisions defined in table 6.5.4.14-1.

**Table 6.5.4.14-1: Enumeration FeasibilityCheckReserveType**

Enumeration value	Description
NO_FEASIBILITY_CHECK	The NFVO shall not perform any feasibility check for the NS LCM operation.  This is the default value.
FEASIBILITY_CHECK_ONLY	The NFVO shall only perform feasibility check for the NS LCM operation, no resources are reserved.
FEASIBILITY_CHECK_WITH_OPERATION	The NFVO shall perform feasibility check and based on the feasibility check result, the NS LCM operation i.e. the actual Instantiate NS or the actual Update NS operation, is performed at the scheduled time i.e. "startTime" or "updateTime" attributes, indicated in the NS LCM operation.



Enumeration value	Description
FEASIBILITY_CHECK_WITH_RESERVATION_AND_OPERATION	The NFVO shall perform feasibility check and based on the feasibility check result, resources are reserved as part of the feasibility check, and the NS LCM operation i.e. the actual Instantiate NS or the actual Update NS operation, is performed at the scheduled time i.e. "startTime" or "updateTime" attributes, indicated in the NS LCM operation using resources reserved during feasibility check.

#### 6.5.4.15 Enumeration: FeasibilityCheckResultType

The enumeration FeasibilityCheckResultType defines the permitted values to represent the result of executing a feasibility check according to the options defined in clause 6.5.4.14. It shall comply with the provisions defined in table 6.5.4.15-1.

**Table 6.5.4.15-1: Enumeration FeasibilityCheckResultType**

Enumeration value	Description
FEASIBILITY_CHECK_DONE	Indicates that the feasibility check process is done for the NS LCM operation. Applicable when the "feasibilityCheckReserve" attribute in the NS LCM operation is set to "FEASIBILITY_CHECK_ONLY" or "FEASIBILITY_CHECK_WITH_OPERATION".
FEASIBILITY_CHECK_DONE_WITH_RESERVATION	Indicates that the feasibility check and reservation process is done for the NS LCM operation. Applicable when the "feasibilityCheckReserve" attribute in the NS LCM operation is set to "FEASIBILITY_CHECK_WITH_RESERVATION_AND_OPERATION".

## 6.6 Handling of errors during NS lifecycle management operations

### 6.6.1 Basic concepts (informative)

#### 6.6.1.1 Motivation

NS lifecycle management operation occurrences can fail. Failure can be caused by multiple reasons, which generally fall into the following categories:

- Transient errors which do not require intervention from a human operator or a higher-layer management entity for resolution, e.g. momentary network outage.
- "Permanent" errors which require such intervention.

It is unreasonable to expect that all errors can be resolved automatically, therefore the possibility of intervention will usually be incorporated in the system design as acknowledged means of error resolution.

#### 6.6.1.2 Failure resolution strategies: Retry, Rollback and Continue

Most transient errors are handled best with a retry mechanism. Retry might happen automatically at the point of failure within the same NS LCM workflow (where it makes sense to limit the number of automatic retries). It is important to strive for designing retry operations that have no unintended side effects from the original invocation of the operation. This is called *idempotent retry*. Idempotent retry can also be used as an on-demand error resolution mechanism (see below) if the original operation failed because of a condition that has been resolved manually by the human operator or by a higher-level management entity, so idempotent retry is suitable for general error resolution in most cases.

However, even if a system is designed with idempotent retry capabilities, eventual success of the operation cannot be guaranteed. In this case, the system needs to decide the error handling strategy, either by a backward action or a forward action. By a backward action, it means the concerned error is not acceptable and permanent. Therefore, the system attempts to resolve the inconsistent state by requesting to roll back the changes made by the operation. By a forward action, it means the concerned error is acceptable and can be fixed later (typically after current operation). Therefore, the system decides to skip the concerned error and continues the operation, e.g. based on policy configuration. Given that, rollback and continue as error handling strategies are also desired to be allowed in the system design.

In many cases, idempotent retry can resolve transient errors and lead to success eventually. Depending on the situation, rollback followed by a repetition of the operation could take longer than a successful retry, as rollback first removes allocated resources and then the repetition of the operation allocates them again, which costs time. Therefore, it often makes sense to perform first idempotent retry, which is followed by either rollback or continue if the retry has failed.

Idempotent retry is meaningful and useful for all operation types. For some operations, rollback is better suited and has a better chance of success. In general, rollback is well-suited for additive operations such as InstantiateNs or scale out, while ill-suited for subtractive ones such as scale in or TerminateNs, or for HealNs. For some operations, continue is better suited if the concerned error is acceptable.

Both rollback and idempotent retry can fail. In that case, the system can be left in an inconsistent state after a failed operation, which requires resolution by a higher-level entity such as the OSS/BSS or human operator.

### 6.6.1.3 Error handling at NFVO and OSS/BSS

If the NFVO executes an NS LCM workflow and encounters a problem, the following options are possible:

- Stop on first error:
  - Once the NFVO encounters an error, the normal execution of the NS LCM workflow is interrupted, and an error handling procedure is triggered (i.e. automatic retry, automatic rollback, automatic fail, escalate). See the clauses below for description of error handling procedures.
  - It is assumed that all NSs and all NFVOs support "stop on first error".

**EXAMPLE 1:** OSS/BSS is attempting to instantiate an NS with 10 VNFs. The first 8 VNFs are instantiated successfully, however, an error occurs when attempting to instantiate VNF #9. The NFVO stops execution and chooses which of the error handling options it invokes (note that it even could try multiple options after each other).

- Best Effort:
  - Each time the NFVO encounters an error, it is decided whether the execution of a part or all of the remaining steps of the NS LCM workflow is performed, or whether the execution is interrupted and an error handling procedure is triggered (i.e. automatic retry, automatic rollback, automatic fail, escalate). See the clauses below for description of error handling procedures.
  - Support of "best effort" requires a suitable workflow design.
  - It is therefore assumed that not all NSs and not all NFVOs support "best effort".

**EXAMPLE 2:** Same example as above. After the error occurs attempting to instantiate VNF #8, the NFVO continues by creating #9 and #10, and then chooses which error handling options it invokes.

The NFVO has the following error handling procedures to react to errors (see clause 6.6.1.2 for general elaboration regarding retry, rollback, and continue):

- Automatic Retry: The NFVO retries (once or more) to continue the execution of the workflow without involving an external entity. Automatic retry of failed parts of the workflow might even be built into the workflow itself. Retry can eventually succeed or fail. Successful retry leads to the NS LCM operation to be reported as successful. Failed retry is typically escalated.
- Automatic Rollback: The NFVO rolls back the NS to the state prior to starting the NS LCM operation without involving an external entity. Rollback can eventually succeed or can fail, preventing the NS from reaching that previous state. Successful rollback leads to the NS LCM operation to be reported as rolled back. Failed rollback is typically escalated.

- **Automatic Continue:** The NFVO skips the error and continue the NS LCM operation without involving an external entity. Continue can eventually succeed or fail. Successful continue leads to the NS LCM operation to be reported as partially completed. Failed continue is typically escalated or trying other error handling procedures like automatic rollback.
- **Escalate:** After failed automatic retry/retries, automatic rollback or automatic continue is typically not the first option in most situations, but the error is preferably reported to the OSS/BSS for further resolution. The same applies if no automatic error resolution was attempted by the NFVO, or if automatic rollback has failed or if automatic continue is not appropriate (e.g. based on policy configuration). This is done by sending an NS LCM operation occurrence notification.
- **Unresolvable Error:** The NFVO determines that the operation has failed and definitely cannot be recovered (e.g. if no retry, no continue, and no rollback is possible), and that escalating the error to the OSS/BSS will have no chance to lead to a resolution either. In this case, the NFVO would report that the operation has terminally failed. After that, other means of resolution can be attempted, such as the invocation of Heal NS, or manual procedures using the GUI of the NFVO or VIM to release stranded resources.

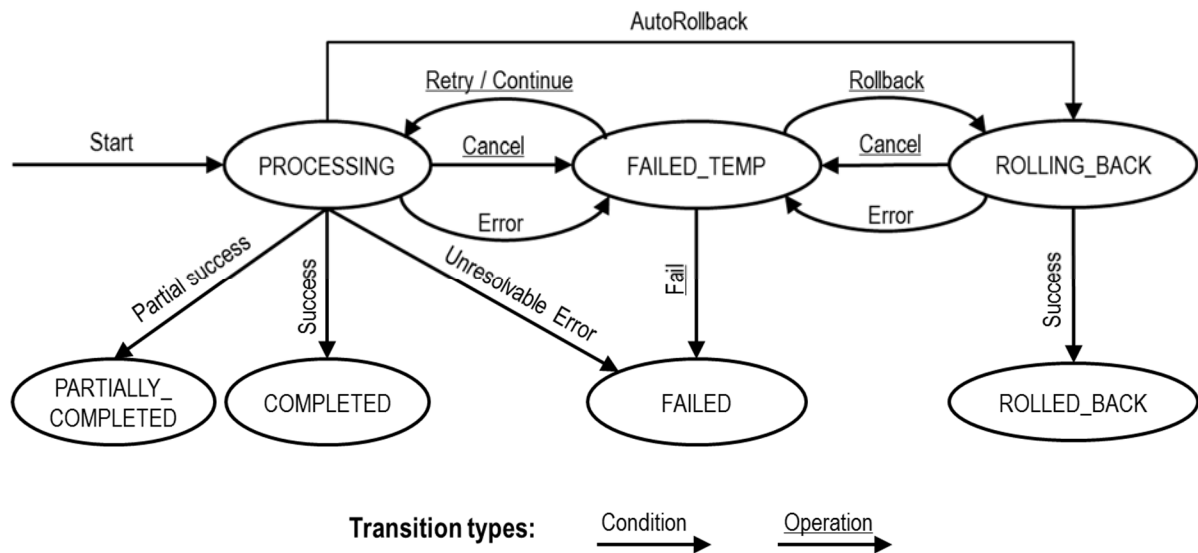
The OSS/BSS has the following error handling procedures to react to error reports from the NFVO:

- **On-demand retry:** After the NFVO has reported the error to the OSS/BSS, the OSS/BSS or the human operator takes steps to resolve the situation that has led to the occurrence of the error. Subsequently, the retry of the operation is triggered towards the NFVO by the OSS/BSS via the NS LCM interface.
- **On-demand rollback:** After the NFVO has reported the error to the OSS/BSS, and after the OSS/BSS or the human operator has decided to roll back the operation, the rollback of the operation is triggered towards the NFVO by the OSS/BSS via the NS LCM interface.
- **On-demand continue:** After the NFVO has reported the error to the OSS/BSS, and after the OSS/BSS or the human operator has decided to continue the operation, the continue of the operation is triggered towards the NFVO by the OSS/BSS via the NS LCM interface.
- **Fail:** After the NFVO has reported the error to the OSS/BSS, and after the OSS/BSS or the human operator has determined that neither on-demand retry nor on-demand rollback will fix the error, or on-demand continue is not appropriate, the NS LCM operation can be declared as terminally failed towards the NFVO by the OSS/BSS via the NS LCM interface. After that, other means of resolution can be attempted, such as the invocation of HealNs, or manual procedures using the GUI of the NFVO or VIM to release stranded resources.

## 6.6.2 States and state transitions of an NS lifecycle management operation occurrence

### 6.6.2.1 General

An NS lifecycle management operation occurrence supports a number of states and error handling operations. The states and state transitions that shall be supported by the NFVO are shown in figure 6.6.2.1-1. Transitions labelled with underlined text represent error handling operations; other transitions represent conditions.



**Figure 6.6.2.1-1: States of an NS lifecycle management operation occurrence**

### 6.6.2.2 States of an NS lifecycle management operation occurrence

At each time, an NS lifecycle management operation occurrence is in one of the following states. There are transitional states (states from which a different state can be reached) and terminal states (states from which no other state can be reached; i.e. the state of an NS lifecycle management operation occurrence in a terminal state cannot change anymore). The state diagram covers the use cases when the operation request has set with or without feasibility checking capability.

**PROCESSING:** The NS LCM operation is currently in execution. This state has the following characteristics:

- This is the initial state for any NS operation.
- This is a transient state.
- During the execution of the "Instantiate NS" and "Update NS" operations, the NFVO executes the feasibility checking if the "feasibilityCheckReserve" attribute is present in the request structures.
- This state may block other NS LCM operations from being executed on the same NS instance (up to NS and NFVO implementation).
- The operations "Retry", "Continue", "Fail", and "Rollback" shall not be permitted to be invoked for an operation that is in this state.
- All failures of procedures executed by the NFVO as part of the NS LCM operation while in "PROCESSING" state should result by default in transiting to FAILED\_TEMP, with the following three alternative options:
  - If a failure occurs in the "PROCESSING" state from which the NFVO knows that the NS instance can be brought into a consistent state by immediately rolling back the operation, the NS lifecycle management operation occurrence may transit directly into the "ROLLING\_BACK" state ("AutoRollback").
  - If a failure occurs in the "PROCESSING" state from which the NFVO knows that it can neither be fixed by retrying nor be rolled back nor be skipped by continuing, the NS lifecycle management operation occurrence may transit directly into the "FAILED" state ("Unresolvable Error").
  - If a failure occurs in the "PROCESSING" state from which the NFVO knows that the failure is acceptable and continues the NS LCM operation till it finishes, the NS lifecycle management operation occurrence may transit directly into the "PARTIALLY\_COMPLETED" state ("Partial success").
- If a "Cancel" request was issued during the operation is in "PROCESSING" state, processing will be cancelled but this might not be immediate. This is represented by a flag in the data model that indicates there is a pending "Cancel" request for this state. Upon successful cancellation, the NS lifecycle management operation occurrence shall transit into the "FAILED\_TEMP" state.

**COMPLETED:** The operation has completed successfully. This is a terminal state.

**PARTIALLY COMPLETED:** The operation has completed partially, i.e. with acceptable errors. This state has the following characteristics:

- This is a terminal state.
- Such an operation state is typically the result of an automatic continue operation inside the NFVO or an on-demand continue operation from a higher layer management entity (i.e. OSS/BSS) for a given error.
- The result of the NS LCM operation (the actual resource changes) can show an inconsistent state of the NS. Nevertheless, these changes shall be synchronized between the NFVO and OSS/BSS (by reporting them in the LCCN, and by allowing the OSS/BSS to obtain them on request) in order for other NS LCM operations (e.g. Heal, Terminate, Update) to be guaranteed to work on resources that are known to the OSS/BSS.

The fact that an LCM operation is in "PARTIALLY\_COMPLETED" state shall not block other operations from execution on the NS instance by the NFVO. However, the NS instance may itself be in a state that disallows certain operations.

**FAILED\_TEMP:** The operation has failed and execution has stopped, but the execution of the operation is not considered to be closed. This state has the following characteristics:

- This is a transient state.
- This state may block other NS LCM operations from being executed on the same NS instance (enforced by the NFVO, and up to NS and NFVO capabilities).
- Retry and/or rollback and/or continue and/or fail may be invoked for the operation.
- If the NS LCM operation is retried or continued, the NS lifecycle management operation occurrence shall transit into the "PROCESSING" state.
- If the NS LCM operation is rolled back, the NS lifecycle management operation occurrence shall transit into the "ROLLING\_BACK" state.
- If the NS LCM operation is marked as "failed", the NS lifecycle management operation occurrence shall transit into the "FAILED" state.
- Operation cancellation and failure to roll back should result in FAILED\_TEMP.

**FAILED:** The operation has failed and it cannot be retried, rolled back, or continued, as it is determined that such action will not succeed. This state has the following characteristics:

- This is a terminal state.
- Such an operation state is typically the result of a decision of a higher layer management entity (i.e. OSS/BSS) or its human operator that an operation in "FAILED\_TEMP" state cannot be retried or rolled back or continued ("Fail").
- Such an operation state can also be reached immediately in case of failure of an operation in "PROCESSING" state that can neither be retried, rolled back, nor continued ("Unresolvable Error").
- The result of the NS LCM operation (the actual resource changes) can show an inconsistent state of the NS. Nevertheless, these changes shall be synchronized between the NFVO and OSS/BSS (by reporting them in the LCCN, and by allowing the OSS/BSS to obtain them on request) in order for other NS LCM operations (e.g. Heal, Terminate) to be guaranteed to work on resources that are known to the OSS/BSS.
- The fact that a LCM operation is in "FAILED" state shall not block other operations from execution on the NS instance by the NFVO. However, the NS instance may itself be in a state that disallows certain operations.

**ROLLED\_BACK:** The state of the NS prior to the original operation invocation has been restored as closely as possible. This state has the following characteristics:

- This is a terminal state.

- This may involve recreating some resources that have been deleted by the operation, the recreated resources should be as similar as possible to the deleted ones. Differences between original resources and re-created ones may include a different resource identity, but also different dynamic attributes such as an IP address.

**ROLLING\_BACK:** The NS LCM operation is currently being rolled back. This state has the following characteristics:

- This is a transient state.
- This state may block other NS LCM operations from being executed on the same NS instance (up to NS and NFVO implementation).
- The operations "Retry", "Continue", and "Rollback" shall not be permitted to be invoked for an operation that is in this state.
- If a "Cancel" request was issued during the operation is in "ROLLING\_BACK" state, rolling back will be cancelled but this might not be immediate. This is represented by a flag in the data model that indicates there is a pending "Cancel" request for this state. Upon successful cancellation, the NS lifecycle management operation occurrence shall transit into the "FAILED\_TEMP" state.
- If a failure occurs during rolling back, the operation should transition to the "FAILED\_TEMP" state.
- Upon successful rollback, the NS lifecycle management operation occurrence shall transit into the "ROLLED\_BACK" state.

In addition, the following provisions apply to NS lifecycle management operation occurrence notifications:

- The "start" notification (i.e. notificationStatus="START") shall be sent when the operation enters one of states "PROCESSING" and "ROLLING\_BACK" from another state, indicating the state entered.
- The "result" notification (i.e. notificationStatus="RESULT") shall be sent when the NS LCM operation occurrence enters one of the error states "FAILED\_TEMP", "FAILED", "ROLLED\_BACK", indicating the state entered, the error cause and the changes to the NS's resources since the operation was initially started.
- The "result" notification (i.e. notificationStatus="RESULT") shall be sent when the operation enters the success state "COMPLETED" or partial success state "PARTIALLY\_COMPLETED", indicating the state entered and the changes to the NS's resources.

The following provisions apply to the sending of notifications related to NS lifecycle changes (NS LCM operation Occurrence Notifications, NS identifier creation and NS identifier deletion notifications):

- The processing of an NS LCM operation occurrence shall not wait for the acknowledgement of the delivery of the triggered notifications.
- Invoking a subsequent LCM operation on the same NS instance shall not be blocked while waiting for the acknowledgement of the delivery of all notifications triggered by a previous LCM operation occurrence on the same NS instance.

Such a notification scheme allows the OSS/BSS to keep in sync with changes to the NS's resources by an ongoing NS LCM operation. If the notification relates to a transient state, further changes can be expected. If the notification relates to a terminal state, no further changes to the NS's resources will be performed by the related NS lifecycle management operation occurrence, and the OSS/BSS can use the information in the notification to synchronize its internal state with the result of the LCM operation. In case of loss of notifications, a query of the resource that represents the NS lifecycle operation occurrence can be used by the OSS/BSS to obtain the same information.

### 6.6.2.3 Error handling operations that change the state of an NS lifecycle operation

**Retry:** This operation retries an NS lifecycle operation. It has the following characteristics:

- Execution of "Retry" for an actual NS LCM operation on a particular NS may be supported, depending on characteristics of the NS and the NS LCM operation.
- The operation may be invoked via an interface, or the NFVO may invoke the operation per its own decision.

**Rollback:** This operation rolls back an NS lifecycle operation. It has the following characteristics:

- Execution of "Rollback" for an actual NS LCM operation on a particular NS may be supported, depending on characteristics of the NS and the NS LCM operation.
- The operation may be invoked via an interface, or the NFVO may invoke the operation per its own decision.

**Continue:** This operation continues an NS lifecycle operation. It has the following characteristics:

- Execution of "Continue" for an actual NS LCM operation on a particular NS may be supported, depending on characteristics of the NS and the NS LCM operation.
- The operation may be invoked via an interface, or the NFVO may invoke the operation per its own decision.

**Fail:** This operation transits the NS lifecycle operation occurrence into the terminal "FAILED" state. It has the following characteristics:

- Execution of "Fail" shall be supported for a LCM operation on a particular NS if at least one of following - Retry, Rollback, Continue, or Cancel - is supported for this operation.
- The operation may be invoked via an interface, or the NFVO may invoke the operation per its own decision.

**Cancel:** This operation cancels an ongoing NS lifecycle management operation, its Retry, Rollback, or Continue. It has the following characteristics:

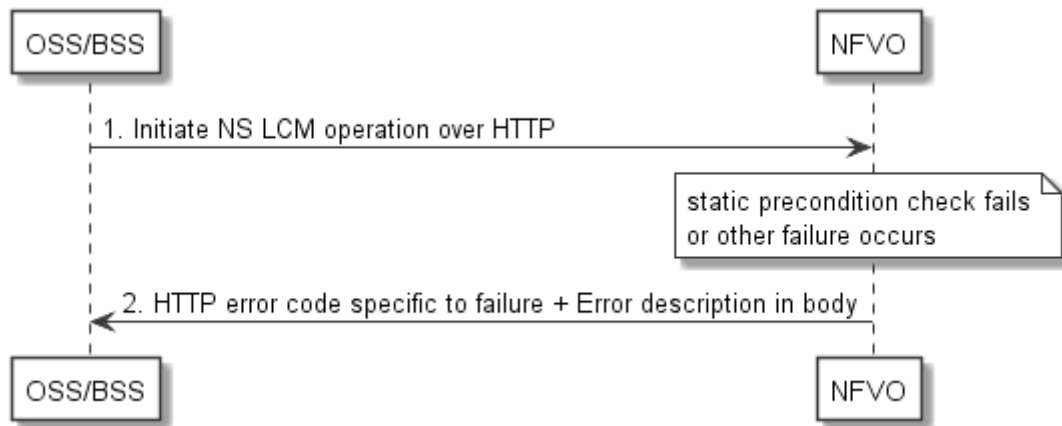
- Execution of "Cancel" for an actual NS LCM operation on a particular NS may be supported, depending on characteristics of the NS and the NS LCM operation.
- The "Cancel" operation need not have immediate effect, depending on the capabilities of the underlying systems, and the currently executed resource management operation.
- Two modes of cancellation are supported: graceful and forceful:
  - When executing the *graceful* "Cancel" operation, the NFVO will not initiate any new operation towards the underlying systems, will wait until the currently executed operations finish or time out, and will then put the NS lifecycle management operation occurrence into the "FAILED\_TEMP" state.
  - When executing the *forceful* "Cancel" operation, the NFVO will cancel all ongoing operations in the underlying systems for which cancellation is supported, will not initiate any new operation towards the underlying systems, will wait for the requested cancellations to finish or time out, and will the put the NS lifecycle management operation occurrence into the "FAILED\_TEMP" state.
- Executing "Cancel" can lead to inconsistencies between the information that the NFVO has about the state of the resources of the NS, and their actual state. The probability of such inconsistencies is bigger when using the *forceful* cancellation mode.

## 6.6.3 Detailed flows

### 6.6.3.1 Immediate failure

If the NS LCM operation fails immediately, i.e. it returns an HTTP error, then the operation has not started, and no "NS LCM operation occurrence resource" has been created. Also, a "start" lifecycle management operation occurrence notification has not been sent. The operation cannot be retried, but the same operation may be invoked again from the API. The NS instance is not changed by a synchronous failure, so no special error handling is required.

Figure 6.6.3.1-1 illustrates the flow.



**Figure 6.6.3.1-1: Immediate failure of an NS LCM operation**

### 6.6.3.2 Failure during actual NS LCM operation execution

After a failed resource management operation, automatic retry can be invoked by the NFVO itself. These invocations are not visible outside of the NFVO, as the NS LCM operation occurrence stays in "PROCESSING" state during these automatic retries. If these do not resolve the issue, intervention (typically by a human operator) is necessary. For that purpose, the NS LCM operation is set into a temporary failure state, and the OSS/BSS is notified. The human operator performs a root cause analysis and eventually resolves the obstacle. Subsequently, and if supported, the operation can be retried, rolled-back or determined as permanently failed. Figure 6.6.3.2-1 illustrates the possible options.

NOTE 1: Excluding automated rollback which is seen as a rare option.

NOTE 2: Excluding "start" notifications (i.e. notificationStatus="START") for simplification purposes.



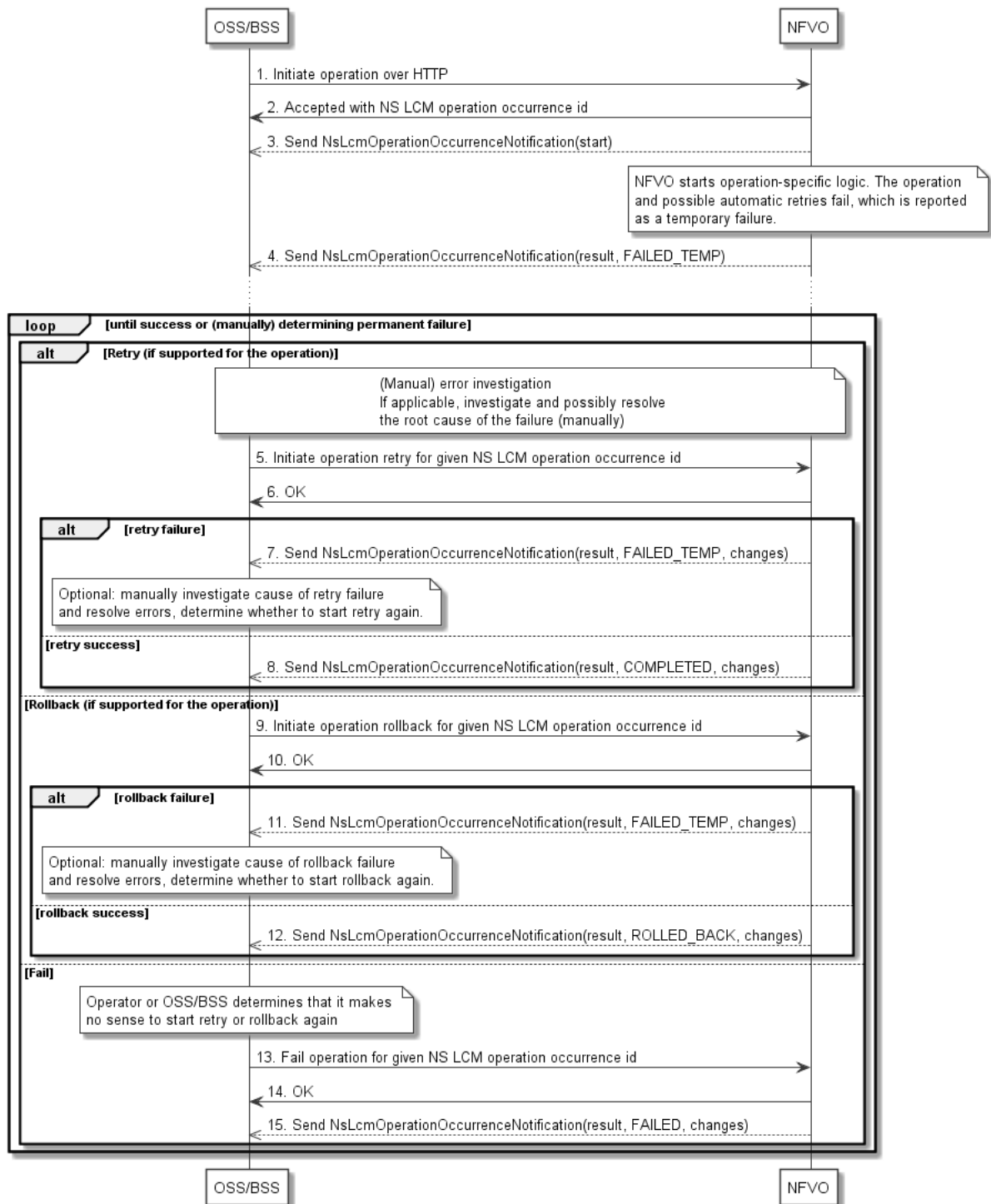


Figure 6.6.3.2-1: Handling failures during the actual execution of an NS LCM operation

### 6.6.3.3 LCM operation cancellation

The cancellation of an NS LCM operation that is in PROCESSING or ROLLING\_BACK state is handled like any other error that leads to stopping the execution of the NS LCM workflow before it can be successfully completed. The NS LCM operation transits into the FAILED\_TEMP state which allows root cause analysis, possible fixing of the root cause, followed by retrying, rolling back, or finally failing of the operation.

## 6.7 Handling of security-sensitive attributes

The NSD allows the NS provider to declare certain additional parameters of NS LCM operations, which are NS-specific, as "sensitive". Likewise, the VNFD allows the VNF provider to declare certain VNF-specific attributes, such as additional parameters of VNF LCM operations or VNF configurable properties, as "sensitive".

In both cases it means that their exposure can be a security risk. Attributes marked as "sensitive" shall be omitted in HTTP response bodies and in notifications in order to prevent their exposure. In case a change to a sensitive attribute is the only modification reported in a notification that notification shall still be sent, omitting the sensitive attribute.

---

# 7 NS Performance Management interface

## 7.1 Description

This interface allows providing performance management (measurement results collection and notifications) related to NSs. Performance information on a given NS instance is sent by the NFVO to the OSS/BSS. Collection and reporting of performance information is controlled by a PM job that groups details of performance collection and reporting information. Further, this interface allows API version information retrieval.

When new performance information is available, the API consumer is notified using the notification `PerformanceInformationAvailableNotification`.

The operations provided through this interface are:

- Create PM Job
- Query PM Job
- Delete PM Job
- Create Threshold
- Query Threshold
- Delete Threshold
- Notify

### 7.1a API version

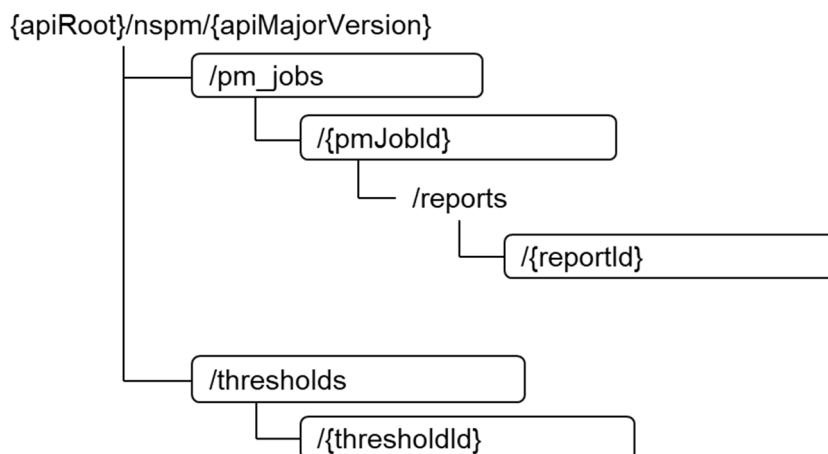
For the NS performance management interface version as specified in the present document, the MAJOR version field shall be 2, the MINOR version field shall be 10, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [16] for a definition of the version fields). Consequently, the `{apiMajorVersion}` URI variable shall be set to "v2".

## 7.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [16].

The string "nspm" shall be used to represent `{apiName}`. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 7.2-1 shows the overall resource URI structure defined for the performance management API.



**Figure 7.2-1: Resource URI structure of the NS Performance Management interface**

Table 7.2-1 lists the individual resources defined, and the applicable HTTP methods.

The NFVO shall support responding to requests for all HTTP methods on the resources in table 7.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [16].

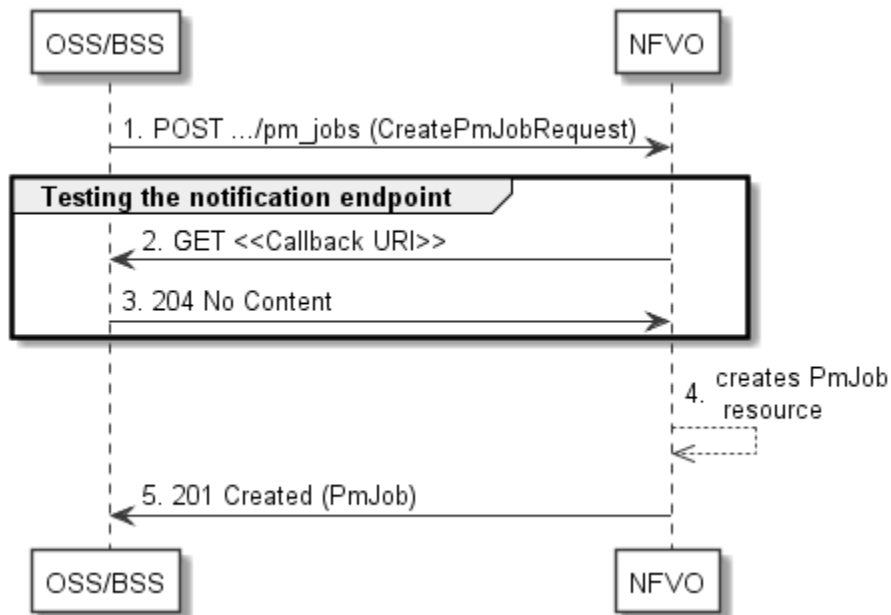
**Table 7.2-1: Resources and methods overview of the NS Performance Management interface**

Resource name	Resource URI	HTTP Method	Cat	Meaning
PM jobs	/pm_jobs	POST	M	Create a PM job.
		GET	M	Query PM jobs.
Individual PM job	/pm_jobs/{pmJobId}	GET	M	Read a single PM job.
		PATCH	M	Update PM job callback.
		DELETE	M	Delete a PM job.
Individual performance report	/pm_jobs/{pmJobId}/reports/{reportId}	GET	M	Read an individual performance report.
Thresholds	/thresholds	POST	M	Create a threshold.
		GET	M	Query thresholds.
Individual threshold	/thresholds/{thresholdId}	GET	M	Read a single threshold.
		PATCH	M	Update threshold callback.
		DELETE	M	Delete a threshold.
Notification endpoint	(provided by API consumer)	POST	See note	Notify about PM related events. See note.
		GET	See note	Test the notification endpoint. See note.
NOTE:	The NFVO shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the OSS/BSS. If the OSS/BSS supports invoking the POST method on the "PM jobs" or "Thresholds" resource towards the NFVO, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.			

## 7.3 Sequence diagrams (informative)

### 7.3.1 Flow of creating a PM job

This clause describes a sequence for creating a performance management jobs.



**Figure 7.3.1-1: Flow of PM job creation**

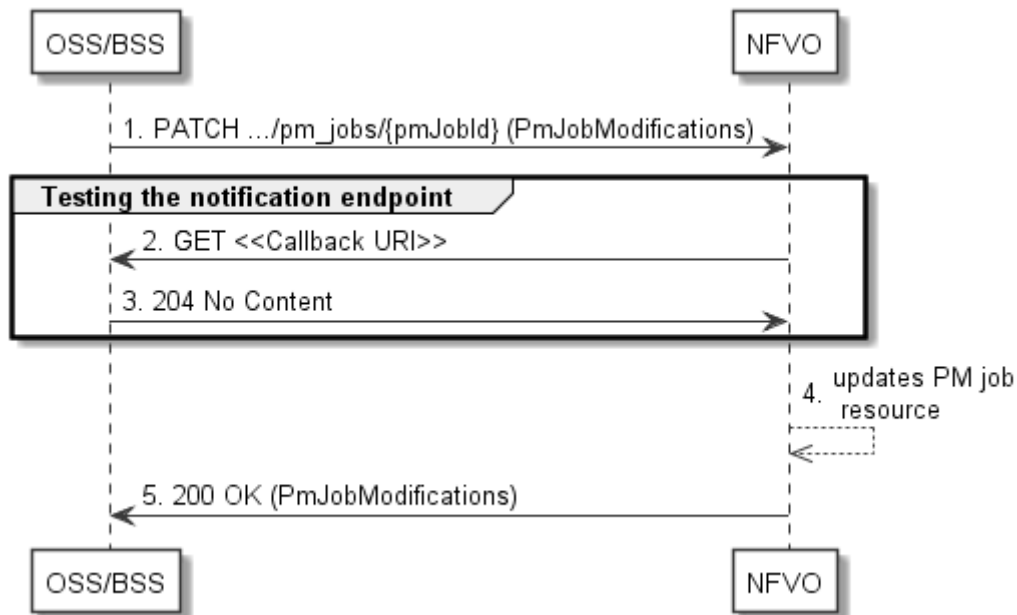
PM job creation, as illustrated in figure 7.3.1-1, consists of the following steps:

- 1) If the OSS/BSS intends to create a PM job, it sends a POST request to the "PM jobs" resource, including one data structure of type "CreatePmJobRequest" in the payload body.
- 2) To test the notification endpoint that was registered by the OSS/BSS during PM job creation, the NFVO sends a GET request to the notification endpoint URI.
- 3) The OSS/BSS returns a "204 No Content" response to indicate success.
- 4) The NFVO creates a PM job instance.
- 5) The NFVO returns a "201 Created" response to the OSS/BSS and includes in the payload body a representation of the PM job just created.

**Error handling:** In case of failure, including an invalid notification endpoint, appropriate error information is provided in the response.

### 7.3.1a Flow of updating the callback URI of a PM job

This clause describes a sequence for updating the callback URI in a PM job.



**Figure 7.3.1a-1: Flow of PM job callback URI update**

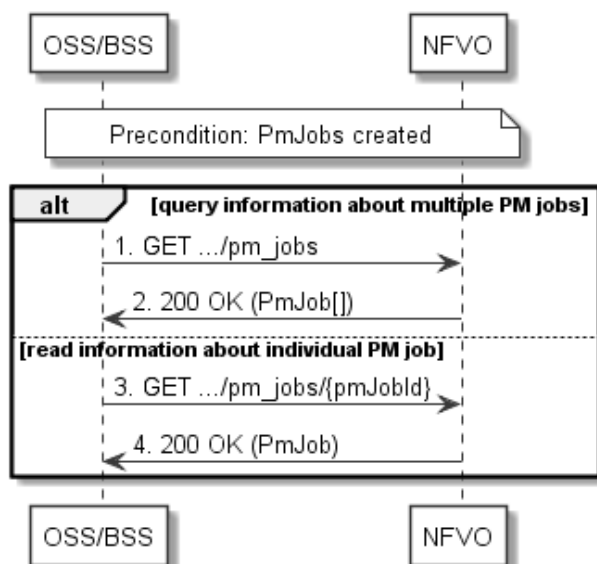
PM job callback URI update, as illustrated in figure 7.3.1a-1, consists of the following steps:

- 1) If the OSS/BSS intends to update the callback URI in a PM job, it sends a PATCH request to the "Individual PM job" resource, including a data structure of type "PmJobModifications" in the payload body.
- 2) To test the notification endpoint that is addressed by the new callback URI, the NFVO sends a GET request to the notification endpoint URI.
- 3) The OSS/BSS returns a "204 No Content" response to indicate success.
- 4) The NFVO updates the callback URI of the "Individual PM job" resource.
- 5) The NFVO returns a "200 OK" response to the OSS/BSS and includes in the payload body a data structure of type "PmJobModifications" to indicate the performed modifications.

**Error handling:** In case of failure, including an invalid notification endpoint, appropriate error information is provided in the response.

## 7.3.2 Flow of querying/reading PM jobs

This clause describes a sequence for querying/reading performance management jobs.



**Figure 7.3.2-1: Flow of PM jobs query/read**

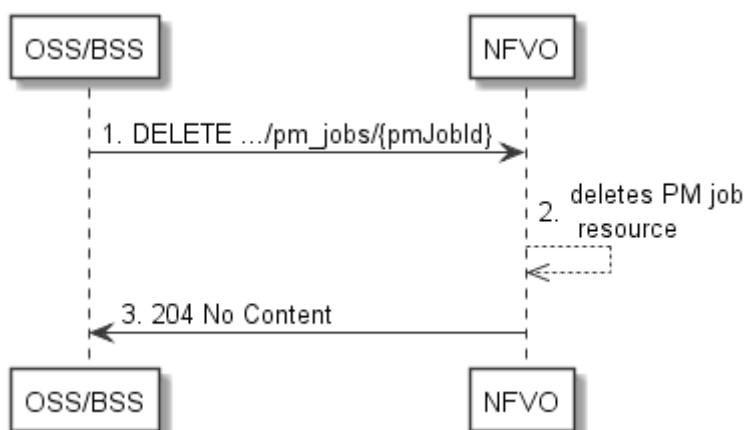
PM jobs query/read, as illustrated in figure 7.3.2-1, consists of the following steps:

- 1) If the OSS/BSS intends to query all PM jobs, it sends a GET request to the "PM jobs" resource.
- 2) The NFVO returns a "200 OK" response to the OSS/BSS, and includes zero or more data structures of type "PmJob" in the payload body.
- 3) If the OSS/BSS intends to read information about a particular PM job, it sends a GET request to the "Individual PM job" resource, addressed by the appropriate PM job identifier in its resource URI.
- 4) The NFVO returns a "200 OK" response to the OSS/BSS, and includes one data structure of type "PmJob" in the payload body.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 7.3.3 Flow of deleting a PM job

This clause describes a sequence for deleting a performance management jobs.



**Figure 7.3.3-1: Flow of PM job deletion**

PM job deletion, as illustrated in figure 7.3.3-1, consists of the following steps:

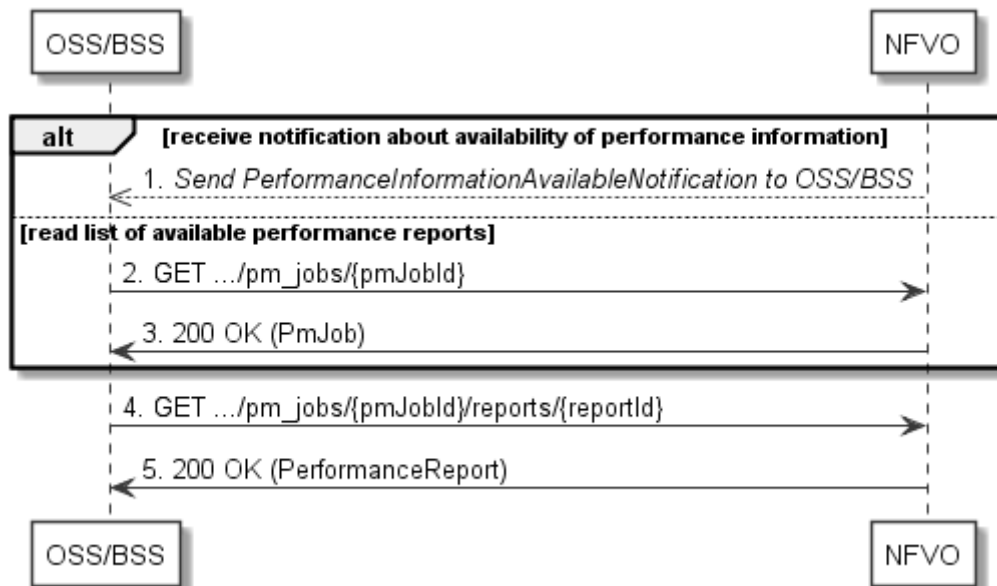
- 1) If the OSS/BSS intends to delete a PM job, it sends a DELETE request to the "Individual PM job" resource, addressed by the appropriate PM job identifier in its resource URI.

- 2) The NFVO deletes the "Individual PM job" resource.
- 3) The NFVO returns a response with a "204 No Content" response code and an empty payload body to the OSS/BSS.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 7.3.4 Flow of obtaining performance reports

This clause describes a sequence for obtaining performance reports.



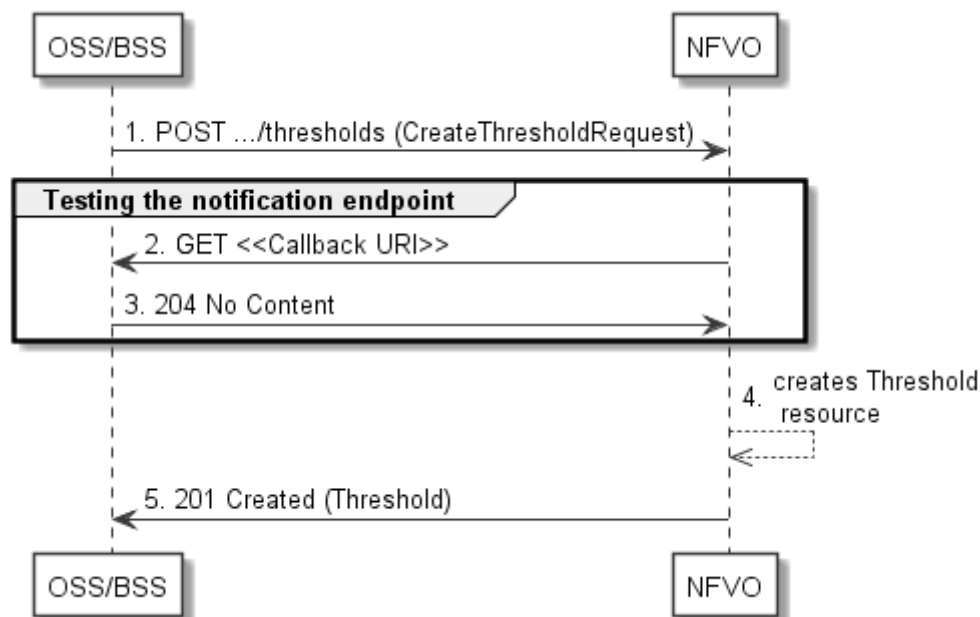
**Figure 7.3.4-1: Flow of obtaining performance reports**

Obtaining a performance report, as illustrated in figure 7.3.4-1, consists of the following steps:

- 1) The NFVO sends to the OSS/BSS a PerformanceInformationAvailableNotification (see clause 7.3.9) that indicates the availability of a new performance report, including a link from which the report can be obtained.
- 2) Alternatively, the OSS/BSS sends a GET request to the "Individual PM job" resource, to obtain a representation of the PM job resource including information about performance reports that are available for this PM job, including their URIs.
- 3) In that case, the NFVO returns a "200 OK" response to the OSS/BSS, and includes a data structure of type "PmJob" in the payload body.
- 4) The OSS/BSS sends to the NFVO a GET request to the URI obtained either in step 1) or step 3), in order to read an "Individual performance report" resource.
- 5) The NFVO returns a "200 OK" response to the OSS/BSS, and includes a data structure of type "PerformanceReport" in the payload body.

### 7.3.5 Flow of creating a threshold

This clause describes a sequence for creating a performance management threshold.



**Figure 7.3.5-1: Flow of threshold creation**

Threshold creation, as illustrated in figure 7.3.5-1, consists of the following steps:

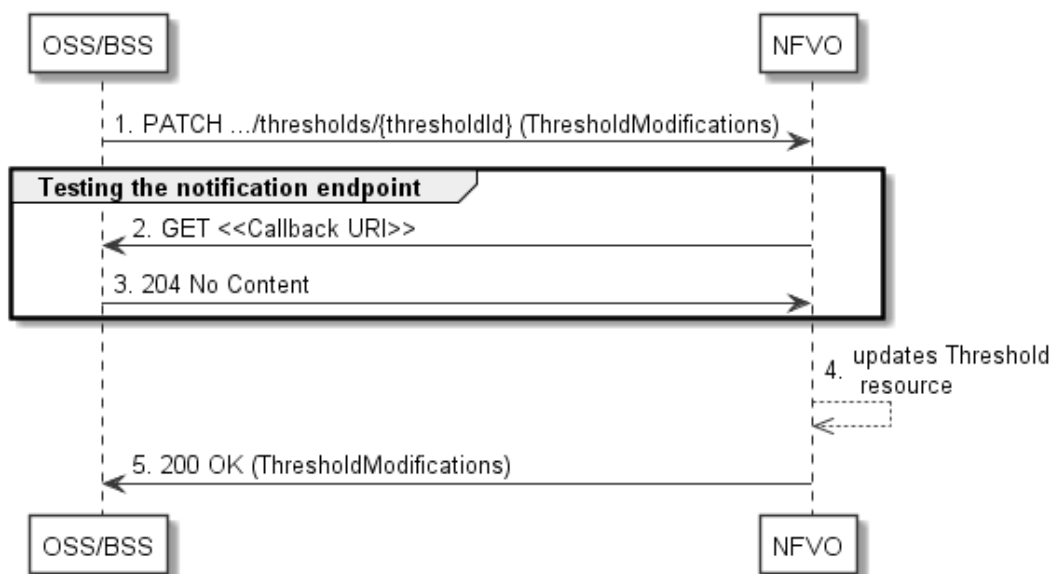
- 1) If the OSS/BSS intends to create a threshold, it sends a POST request to the "Thresholds" resource, including a data structure of type "CreateThresholdRequest" in the payload body.
- 2) To test the notification endpoint that was registered by the OSS/BSS during threshold creation, the NFVO sends a GET request to the notification endpoint URI.
- 3) The OSS/BSS returns a "204 No Content" response to indicate success.
- 4) The NFVO creates a threshold instance.
- 5) The NFVO returns a "201 Created" response to the OSS/BSS and includes in the payload body a representation of the threshold just created.

**Error handling:** In case of failure, including an invalid notification endpoint, appropriate error information is provided in the response.

#### 7.3.5a Flow of updating the callback URI of a threshold

This clause describes a sequence for updating the callback URI in a performance management threshold.





**Figure 7.3.5a-1: Flow of threshold callback URI update**

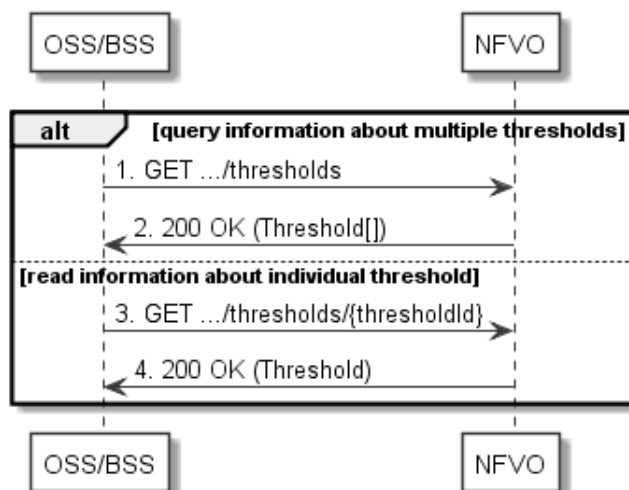
Threshold callback URI update, as illustrated in figure 7.3.5a-1, consists of the following steps:

- 1) If the OSS/BSS intends to update the callback URI in a threshold, it sends a PATCH request to the "Individual threshold" resource, including a data structure of type "ThresholdModifications" in the payload body.
- 2) To test the notification endpoint that is addressed by the new callback URI, the NFVO sends a GET request to the notification endpoint URI.
- 3) The OSS/BSS returns a "204 No Content" response to indicate success.
- 4) The NFVO updates the callback URI of the "Individual Threshold" resource.
- 5) The NFVO returns a "200 OK" response to the OSS/BSS and includes in the payload body a data structure of type "ThresholdModifications" to indicate the performed modifications.

**Error handling:** In case of failure, including an invalid notification endpoint, appropriate error information is provided in the response.

### 7.3.6 Flow of querying/reading thresholds

This clause describes a sequence for querying/reading performance management thresholds.



**Figure 7.3.6-1: Flow of thresholds query/read**

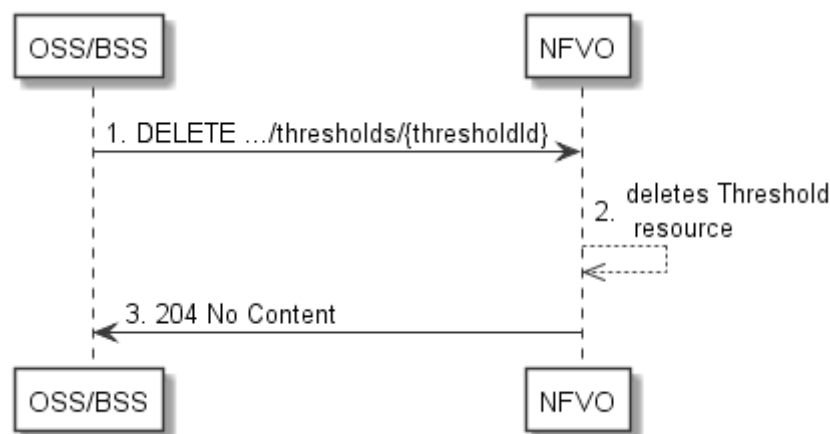
Threshold query/read, as illustrated in figure 7.3.6-1, consists of the following steps:

- 1) If the OSS/BSS intends to query all thresholds, it sends a GET request to the "Thresholds" resource.
- 2) The NFVO returns a "200 OK" response to the OSS/BSS, and includes zero or more data structures of type "Threshold" in the payload body.
- 3) If the OSS/BSS intends to read information about a particular threshold, it sends a GET request to the "Individual threshold" resource with the appropriate threshold identifier in its resource URI.
- 4) The NFVO returns a "200 OK" response to the OSS/BSS, and includes a data structure of type "Threshold" in the payload body.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 7.3.7 Flow of deleting thresholds

This clause describes a sequence for deleting performance management thresholds.



**Figure 7.3.7-1: Flow of threshold deletion**

Threshold deletion, as illustrated in figure 7.3.7-1, consists of the following steps:

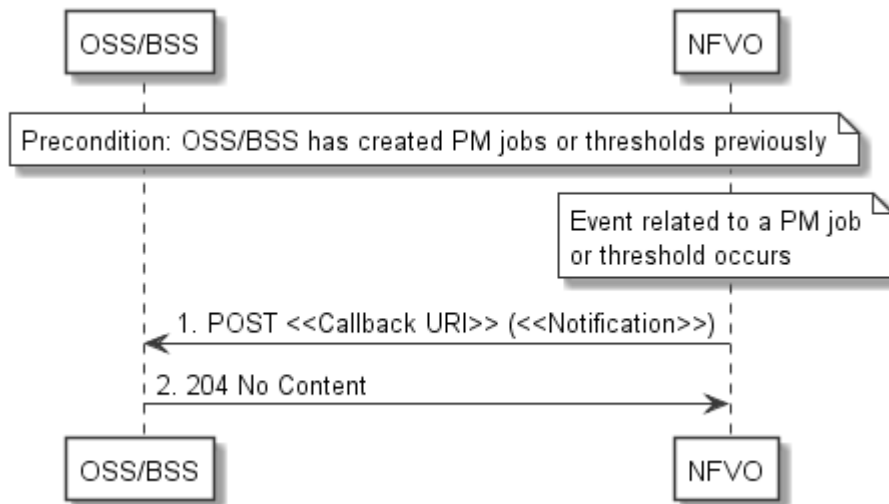
- 1) If the OSS/BSS intends to delete a particular threshold, it sends a DELETE request to the "Individual threshold" resource, addressed by the appropriate threshold identifier in its resource URI.
- 2) The NFVO deletes the "Individual threshold" resource.
- 3) The NFVO returns a "204 No Content" response code to the NFVO. The response body shall be empty.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 7.3.8 Void

### 7.3.9 Flow of sending notifications

This clause describes the procedure for sending notifications related to NS performance management.



**Figure 7.3.9-1: Flow of sending notifications**

**Precondition:** The OSS/BSS has previously created thresholds and or PM jobs which trigger notifications related to NS performance management.

The procedure consists of the following steps as illustrated in figure 7.3.9-1:

- 1) If an event occurs that indicates a threshold crossing or availability of performance information in a PM job, the NFVO generates a notification that includes information about the event, and sends it in the body of a POST request to the URI which the OSS/BSS has registered as part of creating the threshold or PM job. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API (see clauses 7.5.2.4 and 7.5.2.5).
- 2) The OSS/BSS acknowledges the successful delivery of the notification by returning a "204 No Content" response.

**Error handling:** If the NFVO does not receive the "204 No Content" response from the NFVO, it can retry sending the notification.

## 7.4 Resources

### 7.4.1 Introduction

This clause defines all the resources and methods provided by the performance management interface.

#### 7.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [16] are part of the NS performance management interface.

### 7.4.2 Resource: PM jobs

#### 7.4.2.1 Description

This resource represents PM jobs. The API consumer can use this resource to create and query PM jobs.

#### 7.4.2.2 Resource definition

The resource URI is:

`{apiRoot}/nspm/{apiMajorVersion}/pm_jobs`

This resource shall support the resource URI variables defined in table 7.4.2.2-1.

**Table 7.4.2.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 7.1a.

### 7.4.2.3 Resource methods

#### 7.4.2.3.1 POST

The POST method creates a PM job.

This method shall follow the provisions specified in the tables 7.4.2.3.1-1 and 7.4.2.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual PM job" resource as defined in clause 7.4.3 shall have been created.

**Table 7.4.2.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 7.4.2.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	CreatePmJobRequest	1	PM job creation request	
Response body	Data type	Cardinality	Response Codes	Description
	PmJob	1	201 Created	<p>Shall be returned when the PM job has been created successfully.</p> <p>The response body shall contain a representation of the created "Individual PM job" resource, as defined in clause 7.5.2.7.</p> <p>The HTTP response shall include a "Location" HTTP header that points to the created "Individual PM job" resource.</p>
	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 7.4.9.3.2 and the test has failed.</p> <p>In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

## 7.4.2.3.2 GET

The API consumer can use this method to retrieve information about PM jobs.

This method shall follow the provisions specified in the tables 7.4.2.3.2-1 and 7.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 7.4.2.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].  The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.  All attribute names that appear in the PmJob and in data types referenced from it shall be supported by the NFVO in the filter expression.
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.  The following attributes shall be excluded from the PmJob structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: - Reports
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

**Table 7.4.2.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	PmJob	0..N	200 OK	Shall be returned when information about zero or more PM jobs has been queried successfully.  The response body shall contain in an array the representations of zero or more PM jobs, as defined in clause 7.5.2.7.  If the "filter" URI parameter or one of the "all_fields", "fields", "include_fields", "exclude_fields" or "exclude_default" URI parameters was supplied in the request and is supported, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 and 5.3.2 of ETSI GS NFV-SOL 013 [16], respectively.  If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].

	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute selector.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>

#### 7.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 7.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 7.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 7.4.3 Resource: Individual PM job

#### 7.4.3.1 Description

This resource represents an individual PM job. The API consumer can use this resource to delete and read the underlying PM job.

#### 7.4.3.2 Resource definition

The resource URI is:

**{apiRoot}/nspm/{apiMajorVersion}/pm\_jobs/{pmJobId}**

This resource shall support the resource URI variables defined in table 7.4.3.2-1.

**Table 7.4.3.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 7.1a.
pmJobId	Identifier of the PM job. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual PM job" resource. It can also be retrieved from the "id" attribute in the payload body of that response.

### 7.4.3.3 Resource methods

#### 7.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 7.4.3.3.2 GET

The API consumer can use this method for reading an individual PM job.

This method shall follow the provisions specified in the tables 7.4.3.3.2-1 and 7.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 7.4.3.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 7.4.3.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	PmJob	1	200 OK	Shall be returned when information about an individual PM job has been read successfully.  The response body shall contain a representation of the "Individual PM job" resource, as defined in clause 7.5.2.7.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 7.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 7.4.3.3.4 PATCH

This method allows to modify an "individual PM job" resource.

This method shall follow the provisions specified in the tables 7.4.3.3.4-1 and 7.4.3.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.3.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.3.3.4-2: Details of the PATCH request/response on this resource

Request body	Data type	Cardinality	Description	
	PmJobModifications	1	Parameters for the PM job modification.  The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [11].	
Response body	Data type	Cardinality	Response Codes	Description
	PmJobModifications	1	200 OK	Shall be returned when the request has been processed successfully.  The response body shall contain a data structure of type "PmJobModifications".
	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled.  Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity.  The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	422 Unprocessable Entity	Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.  The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.  Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 7.4.9.3.2 and the test has failed.  In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

#### 7.4.3.3.5 DELETE

This method terminates an individual PM job.

This method shall follow the provisions specified in the tables 7.4.3.3.5-1 and 7.4.3.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual PM job" resource shall not exist any longer.



**Table 7.4.3.3.5-1: URI query parameters supported by the DELETE method on this resource**

Name	Cardinality	Description
none supported		

**Table 7.4.3.3.5-2: Details of the DELETE request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the PM job has been deleted successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 7.4.4 Resource: Individual performance report

### 7.4.4.1 Description

This resource represents an individual performance report that has been collected by a PM job. The API consumer can use this resource to read the performance report. The URI of this report can be obtained from a PerformanceInformationAvailableNotification (see clause 7.5.2.5) or from the representation of the "Individual PM job" resource.

It is determined by means outside the scope of the present document, such as configuration or policy, how long an individual performance report is available.

### 7.4.4.2 Resource definition

The resource URI is:

**{apiRoot}/nspm/{apiMajorVersion}/pm\_jobs/{pmJobId}/reports/{reportId}**

This resource shall support the resource URI variables defined in table 7.4.4.2-1.

**Table 7.4.4.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 7.1a.
pmJobId	Identifier of the PM job.
reportId	Identifier of the performance report.

### 7.4.4.3 Resource methods

#### 7.4.4.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 7.4.4.3.2 GET

The API consumer can use this method for reading an individual performance report.

This method shall follow the provisions specified in the tables 7.4.4.3.2-1 and 7.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 7.4.4.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 7.4.4.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	PerformanceReport	1	200 OK	Shall be returned when information of an individual performance report has been read successfully.  The response body shall contain a representation of the "Individual performance report" resource, as defined in clause 7.5.2.10.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 7.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 7.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 7.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 7.4.5 Resource: Thresholds

#### 7.4.5.1 Description

This resource represents thresholds. The API consumer can use this resource to create and query thresholds.

#### 7.4.5.2 Resource definition

The resource URI is:

**{apiRoot}/nspm/{apiMajorVersion}/thresholds**

This resource shall support the resource URI variables defined in table 7.4.5.2-1.

**Table 7.4.5.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 7.1a.

### 7.4.5.3 Resource methods

#### 7.4.5.3.1 POST

The POST method can be used by the API consumer to create a threshold.

This method shall follow the provisions specified in the tables 7.4.5.3.1-1 and 7.4.5.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual threshold" resource as defined in clause 7.4.6 shall have been created.

**Table 7.4.5.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
None supported		

**Table 7.4.5.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	CreateThresholdRequest	1	Request parameters to create a new "Individual threshold" resource.	
Response body	Data type	Cardinality	Response Codes	Description
	Threshold	1	201 Created	<p>Shall be returned when a threshold has been created successfully.</p> <p>The response body shall contain a representation of the created "Individual threshold" resource, as defined in clause 7.5.2.9.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created threshold resource.</p>
	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 7.4.9.3.2 and the test has failed.</p> <p>In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>	

## 7.4.5.3.2 GET

The API consumer can use this method to query information about thresholds.

This method shall follow the provisions specified in the tables 7.4.5.3.2-1 and 7.4.5.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 7.4.5.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].  The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.  All attribute names that appear in the Thresholds data type and in data types referenced from it shall be supported by the NFVO in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

NOTE: There are no attribute selectors defined for this resource as the threshold attributes with cardinality 0..1 or 0..N are not structurally complex in nature.

**Table 7.4.5.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	Threshold	0..N	200 OK	Shall be returned when information about zero or more thresholds has been queried successfully.  If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [16].  The response body shall contain in an array the representations of zero or more thresholds, as defined in clause 7.5.2.9.  If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression.  The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
ProblemDetails	1	400 Bad Request	400 Bad Request	Shall be returned upon the following error: Response too big.  If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].

	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.
--	----------------	------------------------	---------	---

#### 7.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 7.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 7.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 7.4.6 Resource: Individual threshold

#### 7.4.6.1 Description

This resource represents an individual threshold.

#### 7.4.6.2 Resource definition

The resource URI is:

**{apiRoot}/nspm/{apiMajorVersion}/thresholds/{thresholdId}**

This resource shall support the resource URI variables defined in table 7.4.6.2-1.

**Table 7.4.6.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 7.1a.
thresholdId	Identifier of the threshold. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual threshold" resource. It can also be retrieved from the "id" attribute in the payload body of that response.

#### 7.4.6.3 Resource methods

##### 7.4.6.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

##### 7.4.6.3.2 GET

The API consumer can use this method for reading an individual threshold.

This method shall follow the provisions specified in the tables 7.4.6.3.2-1 and 7.4.6.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.6.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.6.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	Threshold	1	200 OK	Shall be returned when information about an individual threshold has been read successfully.  The response body shall contain a representation of the threshold, as defined in clause 7.5.2.9.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 7.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 7.4.6.3.4 PATCH

This method allows to modify an "Individual threshold" resource.

This method shall follow the provisions specified in the tables 7.4.6.3.4-1 and 7.4.6.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.6.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.6.3.4-2: Details of the PATCH request/response on this resource

Request body	Data type	Cardinality	Description	
	ThresholdModifications	1		Parameters for the threshold modification.  The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [11].
Response body	Data type	Cardinality	Response Codes	Description
	ThresholdModifications	1	200 OK	Shall be returned when the request has been processed successfully.  The response body shall contain a data structure of type "ThresholdModifications".
	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled.  Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity.

				The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 7.4.9.3.2 and the test has failed.</p> <p>In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 7.4.6.3.5 DELETE

This method allows to delete a threshold.

This method shall follow the provisions specified in the tables 7.4.6.3.5-1 and 7.4.6.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual threshold" resource shall not exist any longer.

**Table 7.4.6.3.5-1: URI query parameters supported by the DELETE method on this resource**

Name	Cardinality	Description
none supported		

**Table 7.4.6.3.5-2: Details of the DELETE request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	<p>Shall be returned when the threshold has been deleted successfully.</p> <p>The response body shall be empty.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

7.4.7 Void

7.4.8 Void

## 7.4.9 Resource: Notification endpoint

### 7.4.9.1 Description

This resource represents a notification endpoint for NS performance management.

The API producer can use this resource to send notifications related to performance management events to an API consumer which has provided the URI of this resource during the PM job or threshold creation process.

### 7.4.9.2 Resource definition

The resource URI is provided by the API consumer when creating the PM job or threshold.

This resource shall support the resource URI variables defined in table 7.4.9.2-1.

**Table 7.4.9.2-1: Resource URI variables for this resource**

Name	Definition
none supported	

### 7.4.9.3 Resource methods

#### 7.4.9.3.1 POST

The POST method delivers a notification regarding a performance management event from the API producer to an API consumer. The API consumer shall have previously created an "Individual PM job resource" or "Individual threshold resource".

This method shall follow the provisions specified in the tables 7.4.9.3.1-1 and 7.4.9.3.1-2 for URI query parameters, request and response data structures, and response codes.

**Table 7.4.9.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 7.4.9.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	PerformanceInformationAvailableNotification	1	Notification about performance information availability	
	ThresholdCrossedNotification	1	Notification about threshold crossing	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the notification has been delivered successfully.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 7.4.9.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during creation of the PM job or threshold resource.



This method shall follow the provisions specified in the tables 7.4.9.3.2-1 and 7.4.9.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 7.4.9.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 7.4.9.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned to indicate that the notification endpoint has been tested successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 7.4.9.3.3 PUT

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 7.4.9.3.4 PATCH

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 7.4.9.3.5 DELETE

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 7.5 Data Model

### 7.5.1 Introduction

This clause defines the request and response data structures of the NS Performance Management interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error, and may choose to ignore them.

### 7.5.2 Resource and notification data types

#### 7.5.2.1 Introduction

This clause defines the data structures to be used in resource representations and notifications.

#### 7.5.2.2 Void

#### 7.5.2.3 Void

### 7.5.2.4 Type: ThresholdCrossedNotification

This type represents a notification that is sent when a threshold has been crossed. It shall comply with the provisions defined in table 7.5.2.4-1.

NOTE: The timing of sending this notification is determined by the capability of the producing entity to evaluate the threshold crossing condition.

The notification shall be triggered by the NFVO when a threshold has been crossed.

**Table 7.5.2.4-1: Definition of the ThresholdCrossedNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "ThresholdCrossedNotification " for this notification type.
timeStamp	DateTime	1	Date and time of the generation of the notification.
thresholdId	Identifier	1	Identifier of the threshold which has been crossed.
crossingDirection	CrossingDirectionType	1	An indication of whether the threshold was crossed in upward or downward direction.
objectType	String	1	Type of the measured object.  The applicable measured object type for a measurement is defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
objectInstanceId	Identifier	1	Identifier of the measured object instance as per clause 6.2 of ETSI GS NFV-IFA 027 [15].
subObjectInstanceId	IdentifierInNs	0..1	Identifier of the sub-object of the measured object to which the measurement applies. Shall be present if this is required in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type. See note.
performanceMetric	String	1	Performance metric associated with the threshold. This attribute shall contain the related "Measurement Name" value as defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
performanceValue	(any type)	1	Value of the metric that resulted in threshold crossing.  The type of this attribute shall correspond to the related "Measurement Unit" as defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
context	KeyValuePairs	0..1	Measurement context information related to the measured value. The set of applicable keys is defined per measurement in the related "Measurement Context" in clause 7.3 of ETSI GS NFV-IFA 027 [15].
_links	Structure (inlined)	1	Links to resources related to this notification.
>objectInstance	NotificationLink	0..1	Link to the resource representing the measured object instance to which the notified change applies. Shall be present if the measured object instance information is accessible as a resource.
>threshold	NotificationLink	1	Link to the resource that represents the threshold that was crossed.
NOTE: The sub-object allows to structure the measured object but is not to be confused with sub-counters which allow to structure the measurement.			

### 7.5.2.5 Type: PerformanceInformationAvailableNotification

This notification informs the receiver that performance information is available. It shall comply with the provisions defined in table 7.5.2.5-1.

The notification shall be triggered by the NFVO when new performance information collected by a PM job is available.

The periodicity of triggering this notification is influenced by the "reportingPeriod" attribute in the "PmJobCriteria" data structure as defined in clause 7.5.3.3.

**Table 7.5.2.5-1: Definition of the PerformanceInformationAvailableNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "PerformanceInformationAvailableNotification" for this notification type.
timeStamp	DateTime	1	Date and time of the generation of the notification.
pmJobId	Identifier	1	Identifier of the PM job for which performance information is available.
objectType	String	1	Type of the measured object.  The applicable measured object type for a measurement is defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
objectInstanceId	Identifier	1	Identifier of the measured object instance as per clause 6.2 of ETSI GS NFV-IFA 027 [15].
subObjectInstanceIds	IdentifierInNs	0..N	Identifiers of the sub-object instances of the measured object instance for which the measurements have been taken.  Shall be present if the related PM job has been set up to measure only a subset of all sub-object instances of the measured object instance and a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type.  Shall be absent otherwise.
_links	Structure (inlined)	1	Links to resources related to this notification.
>objectInstance	NotificationLink	0..1	Link to the resource representing the measured object instance to which the notification applies. Shall be present if the measured object instance information is accessible as a resource.
>pmJob	NotificationLink	1	Link to the resource that represents the PM job for which performance information is available.
>performanceReport	NotificationLink	1	Link from which the available performance information of data type "PerformanceReport" (see clause 7.5.2.10) can be obtained.  This link should point to an "Individual performance report" resource as defined in clause 7.4.4.

### 7.5.2.6 Type: CreatePmJobRequest

This type represents a request to create a PM job. It shall comply with the provisions defined in table 7.5.2.6-1.

**Table 7.5.2.6-1: Definition of the CreatePmJobRequest data type**

Attribute name	Data type	Cardinality	Description
objectType	String	1	Type of the measured object.  The applicable measured object type for a measurement is defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
objectInstancelds	Identifier	1..N	Identifiers of the measured object instances for which performance information is requested to be collected.
subObjectInstancelds	IdentifierInNs	0..N	Identifiers of the sub-object instances of the measured object instance for which performance information is requested to be collected.  May be present if a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type.  If this attribute is present, the cardinality of the "objectInstancelds" attribute shall be 1.  If this attribute is absent and a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type, measurements will be taken for all sub-object instances of the measured object instance.
criteria	PmJobCriteria	1	Criteria of the collection of performance information.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this PM job, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [16].  This attribute shall only be present if the API consumer requires authorization of notifications.

### 7.5.2.7 Type: PmJob

This type represents a PM job. It shall comply with the provisions defined in table 7.5.2.7-1.

**Table 7.5.2.7-1: Definition of the PmJob data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this PM job.
objectType	String	1	Type of the measured object.  The applicable measured object type for a measurement is defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
objectInstancelds	Identifier	1..N	Identifiers of the measured object instances for which performance information is collected.
subObjectInstancelds	IdentifierInNs	0..N	Identifiers of the sub-object instances of the measured object instance for which performance information is requested to be collected.  May be present if a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type.  If this attribute is present, the cardinality of the "objectInstancelds" attribute shall be 1.  If this attribute is absent and a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type, measurements will be taken for all sub-object instances of the measured object instance.
criteria	PmJobCriteria	1	Criteria of the collection of performance information.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
reports	Structure (inlined)	0..N	Information about available reports collected by this PM job.
>href	Uri	1	The URI where the report can be obtained.
>readyTime	DateTime	1	The time when the report was made available.
>expiryTime	DateTime	0..1	The time when the report will expire.
>fileSize	UnsignedInt	0..1	The size of the report file in bytes, if known.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.
>objects	Link	0..N	Links to resources representing the measured object instances for which performance information is collected. Shall be present if the measured object instance information is accessible as a resource.

### 7.5.2.8 Type: CreateThresholdRequest

This type represents a request to create a threshold. It shall comply with the provisions defined in table 7.5.2.8-1.

Table 7.5.2.8-1: Definition of the CreateThresholdRequest data type

Attribute name	Data type	Cardinality	Description
objectType	String	1	Type of the measured object.  The applicable measured object type for a measurement is defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
objectInstanceId	Identifier	1	Identifier of the measured object instance associated with this threshold.
subObjectInstanceIds	IdentifierInNs	0..N	Identifiers of the sub-object instances of the measured object instance associated with this threshold.  May be present if a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type.  If this attribute is absent and a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the measured object type, measurements will be taken for all sub-object instances of the measured object instance.
criteria	ThresholdCriteria	1	Criteria that define this threshold.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this threshold, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [16].  This attribute shall only be present if the API consumer requires authorization of notifications.

### 7.5.2.9 Type: Threshold

This type represents a threshold. It shall comply with the provisions defined in table 7.5.2.9-1.

Table 7.5.2.9-1: Definition of the Threshold data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this threshold resource.
objectType	String	1	Type of the measured object.  The applicable measured object type for a measurement is defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
objectInstanceId	Identifier	1	Identifier of the measured object instance associated with the threshold.
subObjectInstanceIds	IdentifierInNs	0..N	Identifiers of the sub-object instances of the measured object instance associated with the threshold.  May be present if a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measurement type.  If this attribute is absent and a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type, measurements will be taken for all sub-object instances of the measured object instance.
criteria	ThresholdCriteria	1	Criteria that define this threshold.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.
>object	Link	0..1	Link to a resource representing the measured object instance for which performance information is collected. Shall be present if the measured object instance information is accessible as a resource.

### 7.5.2.10 Type: PerformanceReport

This type defines the format of a performance report provided by the NFVO to the OSS/BSS as a result of collecting performance information as part of a PM job. The type shall comply with the provisions defined in table 7.5.2.10-1.

**Table 7.5.2.10-1: Definition of the PerformanceReport data type**

Attribute name	Data type	Cardinality	Description
entries	Structure (inlined)	1..N	List of performance information entries. Each performance report entry is for a given metric of a given object (i.e. NS instance), but can include multiple collected values.
>objectType	String	1	Type of the measured object.  The applicable measured object type for a measurement is defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
>objectInstanceld	Identifier	1	Identifier of the measured object instance for which the performance metric is reported.
>subObjectInstanceld	IdentifierInNs	0..1	Identifier of the sub-object instance of the measured object instance for which the performance metric is reported. Shall be present if this is required in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type. See note.
>performanceMetric	String	1	Name of the metric collected. This attribute shall contain the related "Measurement Name" value as defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
>performanceValues	Structure (inlined)	1..N	List of performance values with associated timestamp.
>>timeStamp	DateTime	1	Time stamp indicating when the data has been collected.
>>value	(any type)	1	Value of the metric collected. The type of this attribute shall correspond to the related "Measurement Unit" as defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
>>context	KeyValuePairs	0..1	Measurement context information related to the measured value. The set of applicable keys is defined per measurement in the related "Measurement Context" in clause 7.3 of ETSI GS NFV-IFA 027 [15].
NOTE:	The sub-object allows to structure the measured object but is not to be confused with sub-counters which allow to structure the measurement value.		
EXAMPLE:	Measured object: VnfnInstanceXYZ Sub-object: VnfnInstance1 Measurement: vCPU_utilization Sub-counters: vCPU utilization of each of the vCPUs of VnfnInstance1 (vCPU_utilization.vCPU1, vCPU_utilization.vCPU2, etc.).		

### 7.5.2.11 Type: ThresholdModifications

This type represents modifications to a threshold. It shall comply with the provisions defined in table 7.5.2.11-1.

**Table 7.5.2.11-1: Definition of the ThresholdModifications data type**

Attribute name	Data type	Cardinality	Description
callbackUri	Uri	0..1	New value of the "callbackUri" attribute. The value "null" is not permitted. See note.
authentication	SubscriptionAuthentication	0..1	New value of the "authentication" attribute, or "null" to remove the attribute. If present in a request body, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [11]).  This attribute shall not be present in response bodies. See note.
NOTE:	At least one of the attributes defined in this type shall be present in request bodies.		

### 7.5.2.12 Type: PmJobModifications

This type represents modifications to a PM job. It shall comply with the provisions defined in table 7.5.2.12-1.

**Table 7.5.2.12-1: Definition of the PmJobModifications data type**

Attribute name	Data type	Cardinality	Description
callbackUri	Uri	0..1	New value of the "callbackUri" attribute. The value "null" is not permitted. See note.
authentication	SubscriptionAuthentication	0..1	New value of the "authentication" attribute, or "null" to remove the attribute. If present in a request body, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [11]).  This attribute shall not be present in response bodies. See note.
NOTE: At least one of the attributes defined in this type shall be present in request bodies.			

## 7.5.3 Referenced structured data types

### 7.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but can neither be resource representations nor bound to any subscribe/notify mechanism.

### 7.5.3.2 Void

### 7.5.3.3 Type: PmJobCriteria

This type represents collection criteria for PM jobs. It shall comply with the provisions defined in table 7.5.3.3-1.

**Table 7.5.3.3-1: Definition of the PmJobCriteria data type**

Attribute name	Data type	Cardinality	Description
performanceMetric	String	0..N	This defines the types of performance metrics for the specified object instances. Valid values are specified as "Measurement Name" values in clause 7.3 of ETSI GS NFV-IFA 027 [15]. At least one of the two attributes (performance metric or group) shall be present.
performanceMetricGroup	String	0..N	Group of performance metrics. A metric group is a pre-defined list of metrics, known to the API producer that it can decompose to individual metrics. Valid values are specified as "Measurement Group" values in clause 7.3 of ETSI GS NFV-IFA 027 [15]. At least one of the two attributes (performance metric or group) shall be present.
collectionPeriod	UnsignedInt	1	Specifies the periodicity at which the API producer will collect performance information. The unit shall be seconds. See note 1 and note 2.
reportingPeriod	UnsignedInt	1	Specifies the periodicity at which the API producer will report to the API consumer. about performance information. The unit shall be seconds. See note 1 and note 2.
reportingBoundary	DateTime	0..1	Identifies a time boundary after which the reporting will stop. The boundary shall allow a single reporting as well as periodic reporting up to the boundary.



Attribute name	Data type	Cardinality	Description
NOTE 1:	At the end of each reportingPeriod, the API producer will inform the API consumer about availability of the performance data collected for each completed collection period during this reportingPeriod. The reportingPeriod should be equal to or a multiple of the collectionPeriod. In the latter case, the performance data for the collection periods within one reporting period are reported together.		
NOTE 2:	In particular when choosing short collection and reporting periods, the number of PM jobs that can be supported depends on the capability of the producing entity.		

### 7.5.3.4 Type: ThresholdCriteria

This type represents criteria that define a threshold. It shall comply with the provisions defined in table 7.5.3.4-1.

**Table 7.5.3.4-1: Definition of the ThresholdCriteria data type**

Attribute name	Data type	Cardinality	Description
performanceMetric	String	1	Defines the performance metric associated with the threshold. Valid values are specified as "Measurement Name" values in clause 7.2 of ETSI GS NFV-IFA 027 [15].
thresholdType	Enum (inlined)	1	Type of threshold. This attribute determines which other attributes are present in the data structure.  Permitted values: - SIMPLE: Single-valued static threshold  See note 1.
simpleThresholdDetails	Structure (inlined)	0..1	Details of a simple threshold. Shall be present if thresholdType="SIMPLE".
>thresholdValue	Number	1	The threshold value. Shall be represented as a floating point number.
>hysteresis	Number	1	The hysteresis of the threshold.  Shall be represented as a non-negative floating point number.  A notification with crossing direction "UP" will be generated if the measured value reaches or exceeds "thresholdValue" + "hysteresis". A notification with crossing direction "DOWN" will be generated if the measured value reaches or undercuts "thresholdValue" - "hysteresis". See note 2.
NOTE 1:	In the present document, simple thresholds are defined. The definition of additional threshold types is left for future specification.		
NOTE 2:	The hysteresis is defined to prevent storms of threshold crossing notifications. When processing a request to create a threshold, implementations should enforce a suitable minimum value for this attribute (e.g. override the value or reject the request).		

## 7.5.4 Referenced simple data types and enumerations

### 7.5.4.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

### 7.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.2.

### 7.5.4.3 Enumeration: CrossingDirectionType

The enumeration CrossingDirectionType shall comply with the provisions defined in table 7.5.4.3-1.

**Table 7.5.4.3-1: Enumeration CrossingDirectionType**

Enumeration value	Description
UP	The threshold was crossed in upward direction.
DOWN	The threshold was crossed in downward direction.

---

## 8 NS Fault Management interface

### 8.1 Description

This interface allows the OSS/BSS to subscribe to notifications regarding NS alarms provided by the NFVO, and API version information retrieval. An alarm on a given NS results from either a collected virtualised resource fault impacting the connectivity of the NS instance or a VNF alarm, resulting from a virtualised resource alarm, issued by the VNFM for a VNF that is part of this NS instance.

The operations provided through this interface are:

- Get Alarm List
- Acknowledge Alarm
- Subscribe
- Query Subscription Information
- Terminate Subscription
- Notify

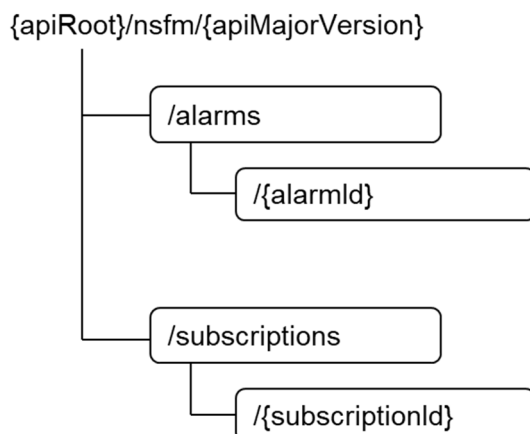
#### 8.1a API version

For the NS fault management interface version as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 10, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [16] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v1".

### 8.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [16]. The string "nsfm" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 8.2-1 shows the overall resource URI structure defined for the NS fault management interface.



**Figure 8.2-1: Resource URI structure of the NS Fault Management interface**

Table 8.2-1 lists the individual resources defined, and the applicable HTTP methods.

The NFVO shall support responding to requests for all HTTP methods on the resources in table 8.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [16].

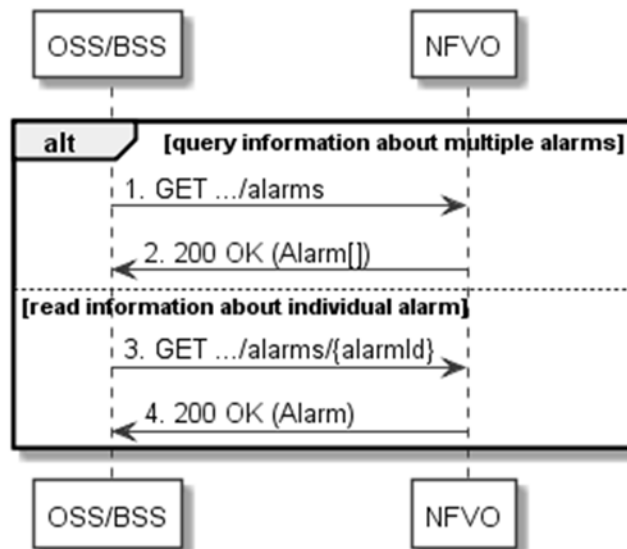
**Table 8.2-1: Resources and methods overview of the NS Fault Management interface**

Resource name	Resource URI	HTTP Method	Cat	Meaning
Alarms	/alarms	GET	M	Query alarms related to NS instances.
Individual alarm	/alarms/{alarmId}	GET	M	Read individual alarm.
		PATCH	M	Acknowledge individual alarm.
Subscriptions	/subscriptions	POST	M	Subscribe to alarms related to NSs.
		GET	M	Query multiple subscriptions.
Individual subscription	/subscriptions/{subscriptionId}	GET	M	Read an individual subscription.
		DELETE	M	Terminate a subscription.
Notification endpoint	(provided by API consumer)	POST	See note	Notify about NS alarms. See note.
		GET	See note	Test the notification endpoint. See note.
NOTE:	The NFVO shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the OSS/BSS. If the OSS/BSS supports invoking the POST method on the "Subscriptions" resource towards the NFVO, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.			

## 8.3 Sequence diagrams (informative)

### 8.3.1 Flow of the Get Alarm List operation

This clause describes a sequence flow for querying one or multiple alarms.



**Figure 8.3.1-1: Flow of alarm query/read**

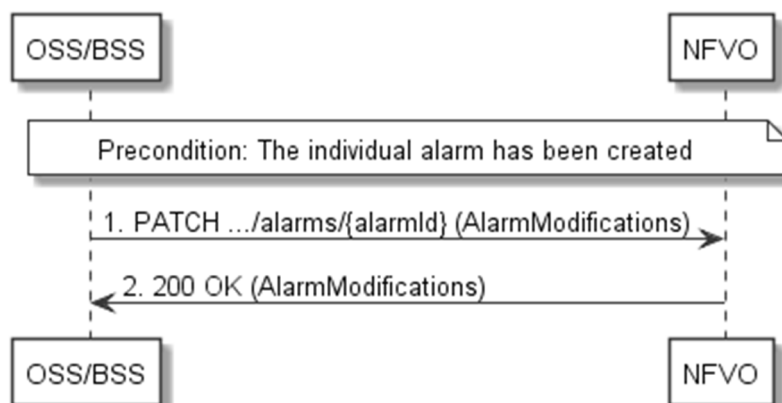
Alarm query, as illustrated in figure 8.3.1-1, consists of the following steps:

- 1) If the OSS/BSS intends to query all alarms, it sends a GET request to the "Alarms " resource.
- 2) The NFVO returns a "200 OK" response to the OSS/BSS, and includes zero or more data structures of type "Alarm" in the payload body.
- 3) If the OSS/BSS intends to read a particular alarm, it sends a GET request to the "Individual alarm" resource, addressed by the appropriate alarm identifier in its resource URI.
- 4) The NFVO returns a "200 OK" response to the OSS/BSS, and includes a data structure of type "Alarm" in the payload body.

**Error handling:** In case of failure, appropriate error information is provided in the response.

## 8.3.2 Flow of acknowledging alarm

This clause describes the procedure to acknowledge an individual alarm.



**Figure 8.3.2-1: Flow of acknowledging alarm**

**Precondition:** The resource representing the individual alarm has been created.

Acknowledge alarm, as illustrated in figure 8.3.2-1, consists of the following steps:

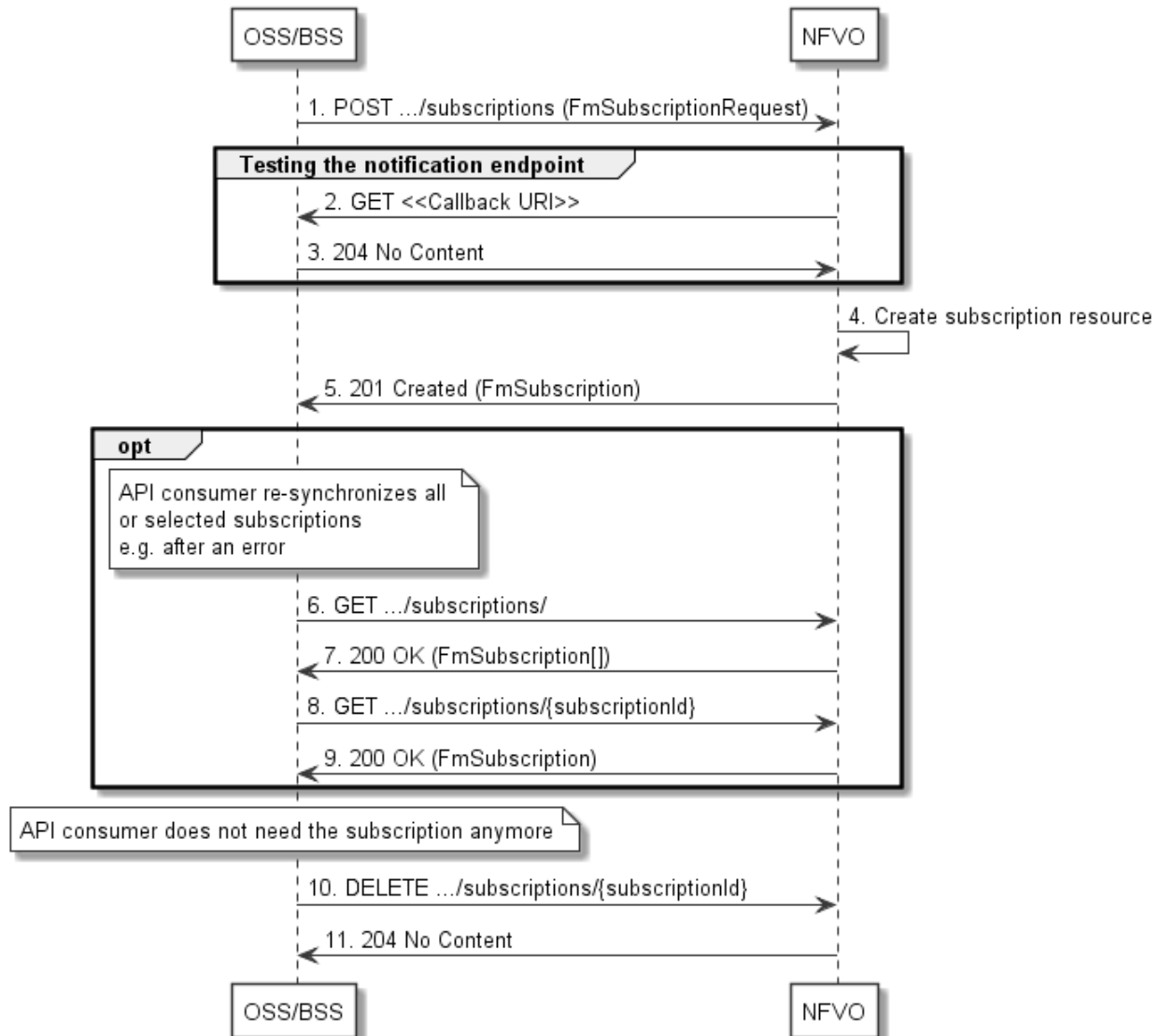
- 1) The OSS/BSS sends a PATCH request to the individual alarm.

- 2) The NFVO returns a "200 OK" response to the OSS/BSS, and includes a data structure of type "AlarmModifications" in the payload body.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 8.3.3 Flow of managing subscriptions

This clause describes the procedure for creating, reading and terminating subscriptions to notifications related to NS fault management.



**Figure 8.3.3-1: Flow of managing subscriptions**

The procedure consists of the following steps as illustrated in figure 8.3.3-1:

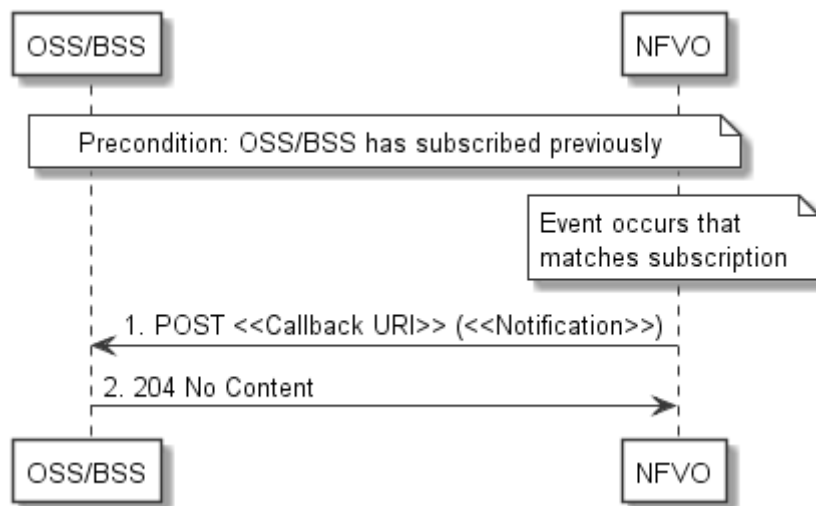
- 1) The OSS/BSS sends a POST request to the "Subscriptions" resource including in the payload body a data structure of type "FmSubscriptionRequest". This data structure contains filtering criteria and a callback URI to which the NFVO will subsequently send notifications about events that match the filter.
- 2) To test the notification endpoint that has been registered by the OSS/BSS as part of the subscription, the NFVO sends a GET request to the notification endpoint URI.
- 3) The OSS/BSS returns a "204 No Content" response to indicate success.

- 4) The NFVO creates a new subscription for notifications related to NS fault management, and a resource that represents this subscription.
- 5) The NFVO returns a "201 Created" response containing a data structure of type "FmSubscription", representing the "Individual subscription" resource just created by the NFVO, and provides the URI of the newly-created resource in the "Location" HTTP header.
- 6) If desired, e.g. to recover from an error situation, the OSS/BSS can query information about its subscriptions by sending a GET request to the "Subscriptions" resource.
- 7) In that case, the NFVO returns a "200 OK" response that contains the list of representations of all existing subscriptions that were created by the NFVO.
- 8) If desired, e.g. to recover from an error situation, the OSS/BSS can read information about a particular subscription by sending a GET request to the resource representing that individual subscription.
- 9) In that case, the NFVO returns a "200 OK" response that contains a representation of that individual subscription.
- 10) When the OSS/BSS does not need the subscription anymore, it terminates the subscription by sending a DELETE request to the resource that represents the individual subscription.
- 11) The NFVO acknowledges the successful termination of the subscription by returning a "204 No Content" response.

**Error handling:** The NFVO rejects a subscription if the subscription information is not valid: endpoint cannot be reached, subscription information is malformed, etc.

### 8.3.4 Flow of sending notifications

This clause describes the procedure for sending notifications related to NS fault management.



**Figure 8.3.4-1: Flow of sending notifications**

**Precondition:** The OSS/BSS has subscribed previously for notifications related to NS fault management.

The procedure consists of the following steps as illustrated in figure 8.3.4-1:

- 1) If an event occurs that matches the filtering criteria which are part of the subscription, the NFVO generates a notification that includes information about the event, and sends it in the body of a POST request to the URI which the NFVO has registered as part of the subscription request. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API (see clauses 8.5.2.5, 8.5.2.6 and 8.5.2.7).

- 2) The OSS/BSS acknowledges the successful delivery of the notification by returning a "204 No Content" response.

**Error handling:** If the NFVO does not receive the "204 No Content" response from the OSS/BSS, it can retry sending the notification.

## 8.4 Resources

### 8.4.1 Introduction

This clause defines all the resources and methods provided by the NS fault management interface.

#### 8.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [16] are part of the NS fault management interface.

### 8.4.2 Resource: Alarms

#### 8.4.2.1 Description

This resource represents a list of alarms related to NS instances.

#### 8.4.2.2 Resource definition

The resource URI is:

**{apiRoot}/nsfm/{apiMajorVersion}/alarms**

This resource shall support the resource URI variables defined in table 8.4.2.2-1.

**Table 8.4.2.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 8.1a.

#### 8.4.2.3 Resource methods

##### 8.4.2.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

##### 8.4.2.3.2 GET

The API consumer can use this method to retrieve information about the alarm list.

This method shall follow the provisions specified in the tables 8.4.2.3.2-1 and 8.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>The following attribute names shall be supported by the NFVO in the filter expression:</p> <ul style="list-style-type: none"> <li>- id</li> <li>- nsInstancelid</li> <li>- rootCauseFaultyComponent.faultyNestedNsInstancelid</li> <li>- rootCauseFaultyComponent.faultyNsVirtualLinkInstancelid</li> <li>- rootCauseFaultyComponent.faultyVnfInstancelid</li> <li>- rootCauseFaultyResource.faultyResourceType</li> <li>- eventType</li> <li>- perceivedSeverity</li> <li>- probableCause</li> </ul>
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

NOTE: There are no attribute selectors defined for this resource as the Alarm attributes with cardinality 0..1 or 0..N are not structurally complex in nature.

Table 8.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	Alarm	0..N	200 OK	<p>Shall be returned when information about zero or more alarms has been queried successfully.</p> <p>The response body shall contain in an array the representations of zero or more alarms as defined in clause 8.5.2.4.</p> <p>If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [16].</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>



	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.
--	----------------	------------------------	---------	---

#### 8.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 8.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 8.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 8.4.3 Resource: Individual alarm

#### 8.4.3.1 Description

This resource represents an individual alarm.

#### 8.4.3.2 Resource definition

The resource URI is:

**{apiRoot}/nsfm/{apiMajorVersion}/alarms/{alarmId}**

This resource shall support the resource URI variables defined in table 8.4.3.2-1.

**Table 8.4.3.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 8.1a.
alarmId	Identifier of the alarm. See note.
NOTE:	This identifier can be retrieved from the "id" attribute of the "alarm" attribute in the AlarmNotification or AlarmClearedNotification. It can also be retrieved from the "id" attribute of the applicable array element in the payload body of the response to a GET request to the "Alarms" resource.

#### 8.4.3.3 Resource methods

##### 8.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

##### 8.4.3.3.2 GET

The API consumer can use this method to read an individual alarm.

This method shall follow the provisions specified in the tables 8.4.3.3.2-1 and 8.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 8.4.3.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 8.4.3.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	Alarm	1	200 OK	Shall be returned when information about an individual alarm has been read successfully.  The response body shall contain a representation of the individual alarm.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 8.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 8.4.3.3.4 PATCH

This method modifies an individual alarm resource.

This method shall follow the provisions specified in the tables 8.4.3.3.4-1 and 8.4.3.3.4-2 for URI query parameters, request and response data structures, and response codes.

**Table 8.4.3.3.4-1: URI query parameters supported by the PATCH method on this resource**

Name	Cardinality	Description
none supported		

Table 8.4.3.3.4-2: Details of the PATCH request/response on this resource

Request body	Data type	Cardinality	Description	
	AlarmModifications	1	The parameter for the alarm modification, as defined in clause 8.5.2.8.  The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [11].	
Response body	Data type	Cardinality	Response Codes	Description
	AlarmModifications	1	200 OK	Shall be returned when the request has been accepted and completed.  The response body shall contain attribute modifications for an 'Individual alarm' resource (see clause 8.5.2.8).
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the "Individual alarm" resource.  Typically, this is due to the fact that the alarm is already in the state that is requested to be set (such as trying to acknowledge an already-acknowledged alarm).  The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled.  Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity.  The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

#### 8.4.3.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 8.4.4 Resource: Subscriptions

#### 8.4.4.1 Description

This resource represents subscriptions. The API consumer can use this resource to subscribe to notifications related to alarms related to an NS and to query its subscriptions.

#### 8.4.4.2 Resource definition

The resource URI is:

**{apiRoot}/nsfm/{apiMajorVersion}/subscriptions**

This resource shall support the resource URI variables defined in table 8.4.4.2-1.

**Table 8.4.4.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 8.1a.

### 8.4.4.3 Resource methods

#### 8.4.4.3.1 POST

The POST method creates a new subscription.

This method shall follow the provisions specified in the tables 8.4.4.3.1-1 and 8.4.4.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual subscription" resource as defined in clause 8.4.5 shall have been created. This method shall not trigger any notification.

Creation of two "Individual subscription" resources with the same callback URI and the same filter can result in performance degradation and will provide duplicates of notifications to the OSS, and might make sense only in very rare use cases. Consequently, the NFVO may either allow creating a new "Individual subscription" resource if another "Individual subscription" resource with the same filter and callback URI already exists (in which case it shall return the "201 Created" response code), or may decide to not create a duplicate "Individual subscription" resource (in which case it shall return a "303 See Other" response code referencing the existing "Individual subscription" resource with the same filter and callback URI).

**Table 8.4.4.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Remarks
none supported		

**Table 8.4.4.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	FmSubscriptionRequest	1	Details of the subscription to be created, as defined in clause 8.5.2.2.	
Response body	Data type	Cardinality	Response Codes	Description
	FmSubscription	1	201 Created	<p>Shall be returned when the subscription has been created successfully.</p> <p>The response body shall contain a representation of the created "Individual subscription" resource.</p> <p>The HTTP response shall include a "Location:" HTTP header that points to the created "Individual subscription" resource.</p>
	n/a		303 See Other	<p>Shall be returned when a subscription with the same callback URI and the same filter already exists and the policy of the NFVO is to not create redundant subscriptions.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource.</p> <p>The response body shall be empty.</p>

	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 8.4.6.3.2 and the test has failed.</p> <p>In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 8.4.4.3.2 GET

The API consumer can use this method to retrieve the list of active subscriptions for alarms related to an NS subscribed by the API consumer. It can be used e.g. for resynchronization after error situations.

This method shall follow the provisions specified in the tables 8.4.4.3.2-1 and 8.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 8.4.4.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Remarks
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>All attribute names that appear in the FmSubscription and in data types referenced from it shall be supported by the NFVO in the filter expression.</p>
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 8.4.4.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	FmSubscription	0..N	200 OK	<p>Shall be returned when the list of subscriptions has been queried successfully.</p> <p>The response body shall contain in an array the representations of all active subscriptions of the functional block that invokes the method, i.e. zero or more representations of FM subscriptions, as defined in clause 8.5.2.3.</p> <p>If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [16].</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>

#### 8.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 8.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 8.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 8.4.5 Resource: Individual subscription

### 8.4.5.1 Description

This resource represents an individual subscription for alarms related to NSs. The API consumer can use this resource to read and to terminate a subscription to notifications related to NS fault management.

### 8.4.5.2 Resource definition

The resource URI is:

**{apiRoot}/nsfm/{apiMajorVersion}/subscriptions/{subscriptionId}**

This resource shall support the resource URI variables defined in table 8.4.5.2-1.

**Table 8.4.5.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 8.1a.
subscriptionId	Identifier of this subscription. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual subscription" resource. It can also be retrieved from the "id" attribute in the payload body of that response.

### 8.4.5.3 Resource methods

#### 8.4.5.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 8.4.5.3.2 GET

The API consumer can use this method for reading an individual subscription for alarms related to NSs subscribed by the API consumer.

This method shall follow the provisions specified in the tables 8.4.5.3.2-1 and 8.4.5.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 8.4.5.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 8.4.5.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	FmSubscription	1	200 OK	Shall be returned when information about an individual subscription has been read successfully.  The response body shall contain a representation of the "Individual subscription" resource.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 8.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 8.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 8.4.5.3.5 DELETE

This method terminates an individual subscription.

This method shall follow the provisions specified in the tables 8.4.5.3.5-1 and 8.4.5.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual subscription" resource shall not exist any longer. This means that no notifications for that subscription shall be sent to the formerly-subscribed API consumer.

NOTE: Due to race conditions, some notifications might still be received by the formerly-subscribed API consumer for a certain time period after the deletion.

**Table 8.4.5.3.5-1: URI query parameters supported by the DELETE method on this resource**

Name	Cardinality	Description
none supported		

**Table 8.4.5.3.5-2: Details of the DELETE request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the "Individual subscription" resource has been deleted successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 8.4.6 Resource: Notification endpoint

### 8.4.6.1 Description

This resource represents a notification endpoint for alarms related to NSs.

The API producer can use this resource to send notifications related to alarms related to NSs or about a rebuilt alarm list to a subscribed API consumer, which has provided the URI of this resource during the subscription process.

### 8.4.6.2 Resource definition

The resource URI is provided by the API consumer when creating the subscription.

This resource shall support the resource URI variables defined in table 8.4.6.2-1.



**Table 8.4.6.2-1: Resource URI variables for this resource**

Name	Definition
none supported	

### 8.4.6.3 Resource methods

#### 8.4.6.3.1 POST

The POST method notifies an alarm related to an NS or that the alarm list has been rebuilt. The API consumer shall have previously created an "individual subscription resource" with a matching filter.

This method shall follow the provisions specified in the tables 8.4.6.3.1-1 and 8.4.6.3.1-2 for URI query parameters, request and response data structures, and response codes.

**Table 8.4.6.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

Each notification request body shall include exactly one of the alternatives defined in table 8.4.6.3.1-2.

**Table 8.4.6.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	AlarmNotification	1	Information of an NS alarm.	
AlarmClearedNotification	1	Information of the clearance of an NS alarm.		
AlarmListRebuiltNotification	1	Information that the alarm list has been rebuilt by the NFVO.		
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the notification has been delivered successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 8.4.6.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during subscription.

This method shall follow the provisions specified in the tables 8.4.6.3.2-1 and 8.4.6.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 8.4.6.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 8.4.6.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned to indicate that the notification endpoint has been tested successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 8.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 8.4.6.3.4 PATCH

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 8.4.6.3.5 DELETE

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 8.5 Data Model

### 8.5.1 Introduction

This clause defines the request and response data structures of the NS fault management interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error, and may choose to ignore them.

### 8.5.2 Resource and notification data types

#### 8.5.2.1 Introduction

This clause defines the data structures to be used in the resource representations and notifications for the NS fault management interface.

#### 8.5.2.2 Type: FmSubscriptionRequest

This type represents a subscription request related to notifications about NS faults. It shall comply with the provisions defined in table 8.5.2.2-1.

**Table 8.5.2.2-1: Definition of the FmSubscriptionRequest data type**

Attribute name	Data type	Cardinality	Description
filter	FmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this subscription, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [16].  This attribute shall only be present if the subscriber requires authorization of notifications.

### 8.5.2.3 Type: FmSubscription

This type represents a subscription related to notifications about NS faults. It shall comply with the provisions defined in table 8.5.2.3-1.

**Table 8.5.2.3-1: Definition of the FmSubscription data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this "Individual subscription" resource.
filter	FmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.

### 8.5.2.4 Type: Alarm

The alarm data type encapsulates information about an alarm. It shall comply with the provisions defined in table 8.5.2.4-1.

**Table 8.5.2.4-1: Definition of the Alarm data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this Alarm information element.
managedObjectId	Identifier	1	Identifier of the affected NS instance.
rootCauseFaultyComponent	FaultyComponentInfo	0..1	The NS components that are causing the NS fault. It shall be present if the alarm affects at least one NS component.
rootCauseFaultyResource	FaultyResourceInfo	0..1	The virtualised resources that are causing the NS fault. It shall be present when the faulty component is "NS Virtual Link" (see clause 8.5.3.4).
alarmRaisedTime	DateTime	1	Time stamp indicating when the alarm is raised by the managed object.
alarmChangedTime	DateTime	0..1	Time stamp indicating when the alarm was last changed. It shall be present if the alarm has been updated.
alarmClearedTime	DateTime	0..1	Time stamp indicating when the alarm was cleared. It shall be present if the alarm has been cleared.
alarmAcknowledgedTime	DateTime	0..1	Time stamp indicating when the alarm was acknowledged. It shall be present if the alarm has been acknowledged.

Attribute name	Data type	Cardinality	Description
ackState	Enum (inlined)	1	Acknowledgement state of the alarm.  Permitted values: - UNACKNOWLEDGED - ACKNOWLEDGED
perceivedSeverity	PerceivedSeverityType	1	Perceived severity of the managed object failure.
eventTime	DateTime	1	Time stamp indicating when the fault was observed.
eventType	EventType	1	Type of event.
faultType	String	0..1	Additional information to clarify the type of the fault.
probableCause	String	1	Information about the probable cause of the fault.
isRootCause	Boolean	1	Attribute indicating if this fault is the root for other correlated alarms. If TRUE, then the alarms listed in the attribute "correlatedAlarmIds" are caused by this fault.
correlatedAlarmIds	Identifier	0..N	List of identifiers of other alarms correlated to this fault.
faultDetails	String	0..N	Provides additional information about the fault.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.
>objectInstance	Link	0..1	Link to the resource representing the NS instance to which the notified alarm is correlated. Shall be present if the NS instance information is accessible as a resource.

### 8.5.2.5 Type: AlarmNotification

This type represents an alarm notification about NS faults. It shall comply with the provisions defined in table 8.5.2.5-1.

This notification shall be triggered by the NFVO when:

- An alarm has been created.
- An alarm has been updated, e.g. the severity of the alarm has changed.

**Table 8.5.2.5-1: Definition of the AlarmNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "AlarmNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
alarm	Alarm	1	Information about an alarm including AlarmId, affected NS identifier, and FaultDetails.
_links	Structure (inlined)	1	Links to resources related to this notification.
>subscription	NotificationLink	1	Link to the related subscription.

### 8.5.2.6 Type: AlarmClearedNotification

This type represents an alarm cleared notification about NS faults. It shall comply with the provisions defined in table 8.5.2.6-1.

The notification shall be triggered by the NFVO when an alarm has been cleared.

**Table 8.5.2.6-1: Definition of the AlarmClearedNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "AlarmClearedNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
alarmId	Identifier	1	Alarm identifier.
alarmClearedTime	DateTime	1	The time stamp indicating when the alarm was cleared.
_links	Structure (inlined)	1	Links to resources related to this notification.
>subscription	NotificationLink	1	Link to the related subscription.
>alarm	NotificationLink	1	Link to the resource that represents the related alarm.

### 8.5.2.7 Type: AlarmListRebuiltNotification

This type represents a notification that the alarm list has been rebuilt, e.g. if the NFVO detects its storage holding the alarm list is corrupted. It shall comply with the provisions defined in table 8.5.2.7-1.

The notification shall be triggered by the NFVO when the alarm list has been rebuilt, e.g. because the NFVO has detected that its storage holding the alarm list was corrupted.

**Table 8.5.2.7-1: Definition of the AlarmListRebuiltNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "AlarmListRebuiltNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
_links	Structure (inlined)	1	Links to resources related to this notification.
>subscription	NotificationLink	1	Link to the related subscription.
>alarms	NotificationLink	1	Link to the alarm list, i.e. the "Alarms" resource.

### 8.5.2.8 Type: AlarmModifications

This type represents attribute modifications for an "Individual alarm" resource, i.e. modifications to a resource representation based on the "Alarm" data type. The attributes of "Alarm" that can be modified according to the provisions in clause 8.5.2.4 are included in the "AlarmModifications" data type.

The "AlarmModifications" data type shall comply with the provisions defined in table 8.5.2.8-1.

**Table 8.5.2.8-1: Definition of the AlarmModifications data type**

Attribute name	Data type	Cardinality	Description
ackState	Enum (inlined)	1	New value of the "ackState" attribute in "Alarm". Permitted values: - ACKNOWLEDGED - UNACKNOWLEDGED

## 8.5.3 Referenced structured data types

### 8.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but can neither be resource representations nor bound to any subscribe/notify mechanism.

### 8.5.3.2 Type: FmNotificationsFilter

This type represents a subscription filter related to notifications about NS faults. It shall comply with the provisions defined in table 8.5.3.2-1.

At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).

**Table 8.5.3.2-1: Definition of the FmNotificationsFilter data type**

Attribute name	Data type	Cardinality	Description
nsInstanceSubscriptionFilter	NsInstanceSubscriptionFilter	0..1	Filter criteria to select NS instances about which to notify.
notificationTypes	Enum (inlined)	0..N	Match particular notification types.  Permitted values: - AlarmNotification - AlarmClearedNotification - AlarmListRebuiltNotification See note.
faultyResourceTypes	FaultyResourceType	0..N	Match alarms related to NSs with a faulty resource type listed in this attribute.
perceivedSeverities	PerceivedSeverityType	0..N	Match alarms related to NSs with a perceived severity listed in this attribute.
eventTypes	EventType	0..N	Match alarms related to NSs with an event type listed in this attribute.
probableCauses	String	0..N	Match alarms related to NSs with a probable cause listed in this attribute.
NOTE: The permitted values of the "notificationTypes" attribute are spelled exactly as the names of the notification types to facilitate automated code generation systems.			

### 8.5.3.3 Type: FaultyResourceInfo

This type represents the faulty virtual resources that have a negative impact on an NS. It shall comply with the provisions defined in table 8.5.3.3-1.

**Table 8.5.3.3-1: Definition of the FaultyResourceInfo data type**

Attribute name	Data type	Cardinality	Description
faultyResource	ResourceHandle	1	Information that identifies the faulty resource instance and its managing entity.
faultyResourceType	FaultyResourceType	1	Type of the faulty resource.

### 8.5.3.4 Type: FaultyComponentInfo

This type represents the faulty component that has a negative impact on an NS. It shall comply with the provisions defined in table 8.5.3.4-1.

**Table 8.5.3.4-1: Definition of the FaultyComponentInfo data type**

Attribute name	Data type	Cardinality	Description
faultyNestedNsInstancelid	Identifier	0..1	Identifier of the faulty nested NS instance. See note.
faultyNsVirtualLinkInstancelid	Identifier	0..1	Identifier of the faulty NS virtual link instance. See note.
faultyVnfInstancelid	Identifier	0..1	Identifier of the faulty VNF instance. See note.

NOTE: At least one of the attributes shall be present.

## 8.5.4 Referenced simple data types and enumerations

### 8.5.4.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

### 8.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.2.

### 8.5.4.3 Enumeration: PerceivedSeverityType

The enumeration PerceivedSeverityType shall comply with the provisions defined in table 8.5.4.3-1. It indicates the relative level of urgency for operator attention.

**Table 8.5.4.3-1: Enumeration PerceivedSeverityType**

Enumeration value	Description
CRITICAL	The Critical severity level indicates that a service affecting condition has occurred and an immediate corrective action is required. Such a severity can be reported, for example, when a managed object becomes totally out of service and its capability needs to be restored (Recommendation ITU-T X.733 [14]).
MAJOR	The Major severity level indicates that a service affecting condition has developed and an urgent corrective action is required. Such a severity can be reported, for example, when there is a severe degradation in the capability of the managed object and its full capability needs to be restored (Recommendation ITU-T X.733 [14]).
MINOR	The Minor severity level indicates the existence of a non-service affecting fault condition and that corrective action should be taken in order to prevent a more serious (for example, service affecting) fault. Such a severity can be reported, for example, when the detected alarm condition is not currently degrading the capacity of the managed object (Recommendation ITU-T X.733 [14]).
WARNING	The Warning severity level indicates the detection of a potential or impending service affecting fault, before any significant effects have been felt. Action should be taken to further diagnose (if necessary) and correct the problem in order to prevent it from becoming a more serious service affecting fault (Recommendation ITU-T X.733 [14]).
INDETERMINATE	The Indeterminate severity level indicates that the severity level cannot be determined (Recommendation ITU-T X.733 [14]).
CLEARED	The Cleared severity level indicates the clearing of one or more previously reported alarms. This alarm clears all alarms for this managed object that have the same Alarm type, Probable cause and Specific problems (if given) (Recommendation ITU-T X.733 [14]).

### 8.5.4.4 Enumeration: EventType

The enumeration EventType represents those types of events that trigger an alarm. It shall comply with the provisions defined in table 8.5.4.4-1.

**Table 8.5.4.4-1: Enumeration EventType**

Enumeration value	Description
COMMUNICATIONS_ALARM	An alarm of this type is associated with the procedure and/or process required conveying information from one point to another (Recommendation ITU-T X.733 [14]).
PROCESSING_ERROR_ALARM	An alarm of this type is associated with a software or processing fault (Recommendation ITU-T X.733 [14]).
ENVIRONMENTAL_ALARM	An alarm of this type is associated with a condition related to an enclosure in which the equipment resides (Recommendation ITU-T X.733 [14]).
QOS_ALARM	An alarm of this type is associated with degradation in the quality of a service (Recommendation ITU-T X.733 [14]).
EQUIPMENT_ALARM	An alarm of this type is associated with an equipment fault (Recommendation ITU-T X.733 [14]).

### 8.5.4.5 Enumeration: FaultyResourceType

The enumeration FaultyResourceType represents those types of faulty resource. It shall comply with the provisions defined in table 8.5.4.5-1.

**Table 8.5.4.5-1: Enumeration FaultyResourceType**

Enumeration value	Description
COMPUTE	Virtual compute resource
STORAGE	Virtual storage resource
NETWORK	Virtual network resource

## 9 VNF Package Management interface

### 9.1 Description

This interface allows the OSS/BSS to invoke VNF package management operations towards the NFVO, and to subscribe to notifications regarding VNF package on-boarding or changes provided by the NFVO. Further, this interface allows API version information retrieval.

The operations provided through this interface are as follows:

- Create VNF Package Info
- Upload VNF Package
- Update VNF Package Info
- Delete VNF Package
- Query VNF Package, including obtaining the VNFD
- Fetch VNF Package
- Fetch VNF Package Artifacts
- Subscribe
- Query Subscription Info
- Notify
- Terminate Subscription

State changes of a VNF package are illustrated in clause B.2.



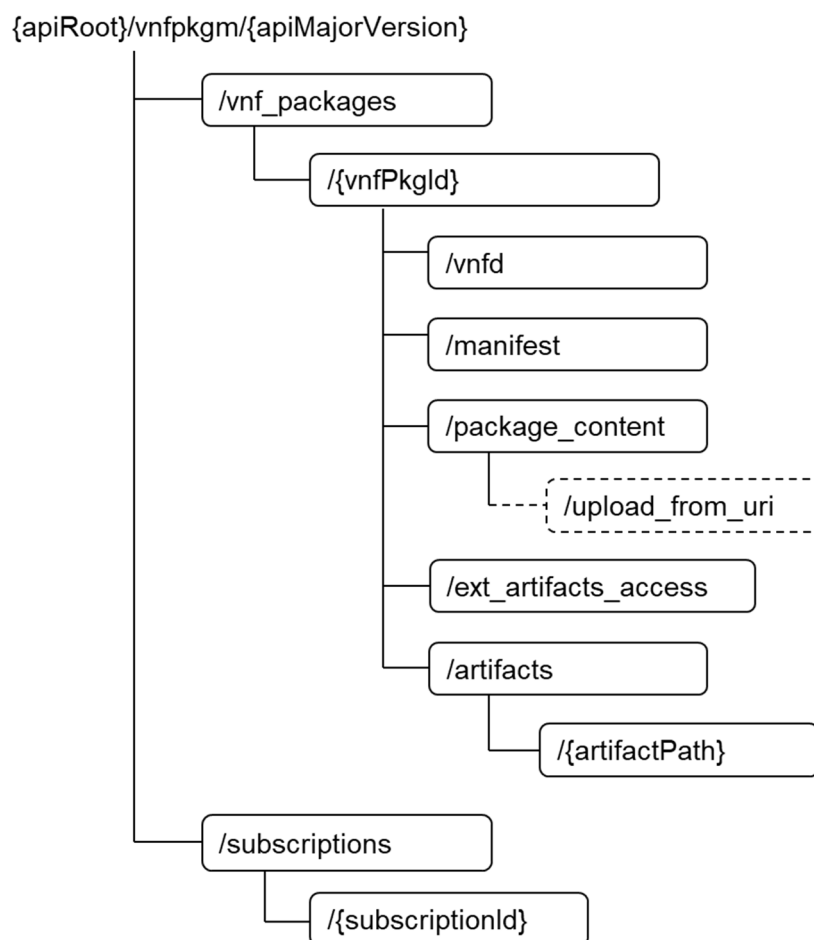
## 9.1a API version

For the VNF package management interface version as specified in the present document, the MAJOR version field shall be 2, the MINOR version field shall be 10, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [16] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v2".

## 9.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [16]. The string "vnfpgm" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 9.2-1 shows the overall resource URI structure defined for the VNF package management interface.



**Figure 9.2-1: Resource URI structure of the VNF Package Management Interface**

Table 9.2-1 lists the individual resources defined, and the applicable HTTP methods. The NFVO shall support responding to requests for all HTTP methods on the resources in table 9.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [16].

Table 9.2-1: Resources and methods overview of the VNF Package Management interface

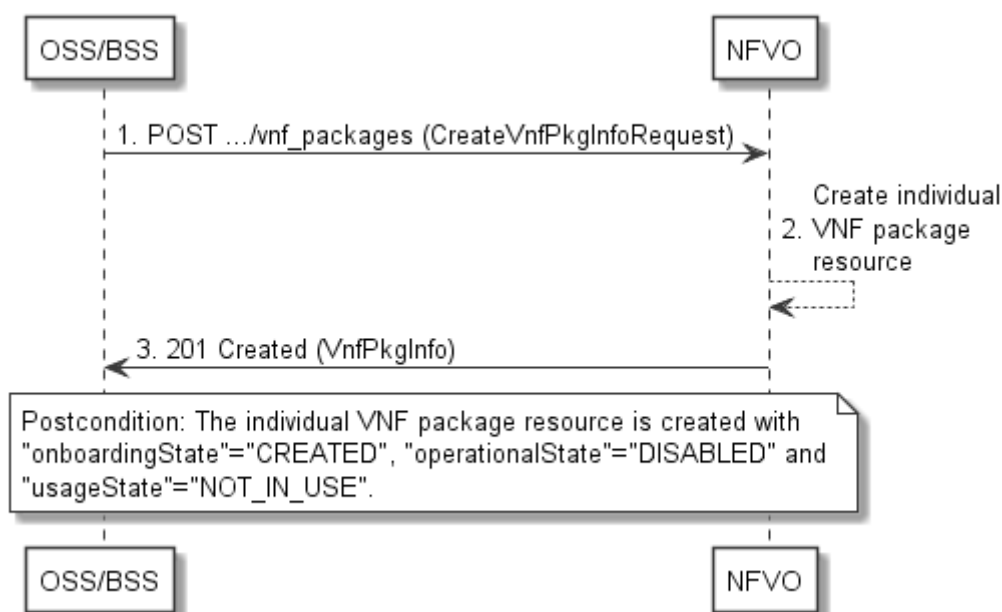
Resource name	Resource URI	HTTP Method	Cat	Meaning
VNF packages	/vnf_packages	GET	M	Query VNF packages information.
		POST	M	Create a new individual VNF package resource.
Individual VNF package	/vnf_packages/{vnfPkgId}	GET	M	Read information about an individual VNF package.
		PATCH	M	Update information about an individual VNF package.
		DELETE	M	Delete an individual VNF package.
VNFD of an individual VNF package	/vnf_packages/{vnfPkgId}/vnfd	GET	M	Read VNFD of an on-boarded VNF package.
Manifest of an individual VNF package	/vnf_packages/{vnfPkgId}/manifest	GET	M	Read the manifest of an on-boarded VNF package
VNF package content	/vnf_packages/{vnfPkgId}/package_content	GET	M	Fetch an on-boarded VNF package.
		PUT	M	Upload a VNF package by providing the content of the VNF package.
VNF package artifacts	/vnf_packages/{vnfPkgId}/artifacts/	GET	M	Bulk-fetch artifacts that are not images.
Upload VNF package from URI task	/vnf_packages/{vnfPkgId}/package_content/upload_from_uri	POST	M	Upload a VNF package by providing the address information of the VNF package.
Access configuration for external artifacts download	/vnf_packages/{vnfPkgId}/ext_artifacts_access	PUT	M	Provide access configuration information for the download of external artifacts.
		GET	M	Read access configuration information for the download of external artifacts.
Individual VNF package artifact	/vnf_packages/{vnfPkgId}/artifacts/{artifactPath}	GET	M	Fetch individual VNF package artifact.
Subscriptions	/subscriptions	POST	M	Subscribe to notifications related to on-boarding and/or changes of VNF packages.
		GET	M	Query multiple subscriptions.
Individual subscription	/subscriptions/{subscriptionId}	GET	M	Read information about an individual subscription.
		DELETE	M	Terminate a subscription.

Resource name	Resource URI	HTTP Method	Cat	Meaning
Notification endpoint	(provided by API consumer)	POST	See note	Notify about VNF package on-boarding or change. See note.
		GET	See note	Test the notification endpoint. See note.
NOTE: The NFVO shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the OSS/BSS. If the OSS/BSS supports invoking the POST method on the "Subscriptions" resource towards the NFVO, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.				

## 9.3 Sequence diagrams (informative)

### 9.3.1 Flow of the creation of an individual VNF package resource

This clause describes the procedure for creating an "Individual VNF package" resource.



**Figure 9.3.1-1: Flow of the creation of an individual VNF package resource**

Creation of an "Individual VNF package" resource, as illustrated in figure 9.3.1-1, consists of the following steps:

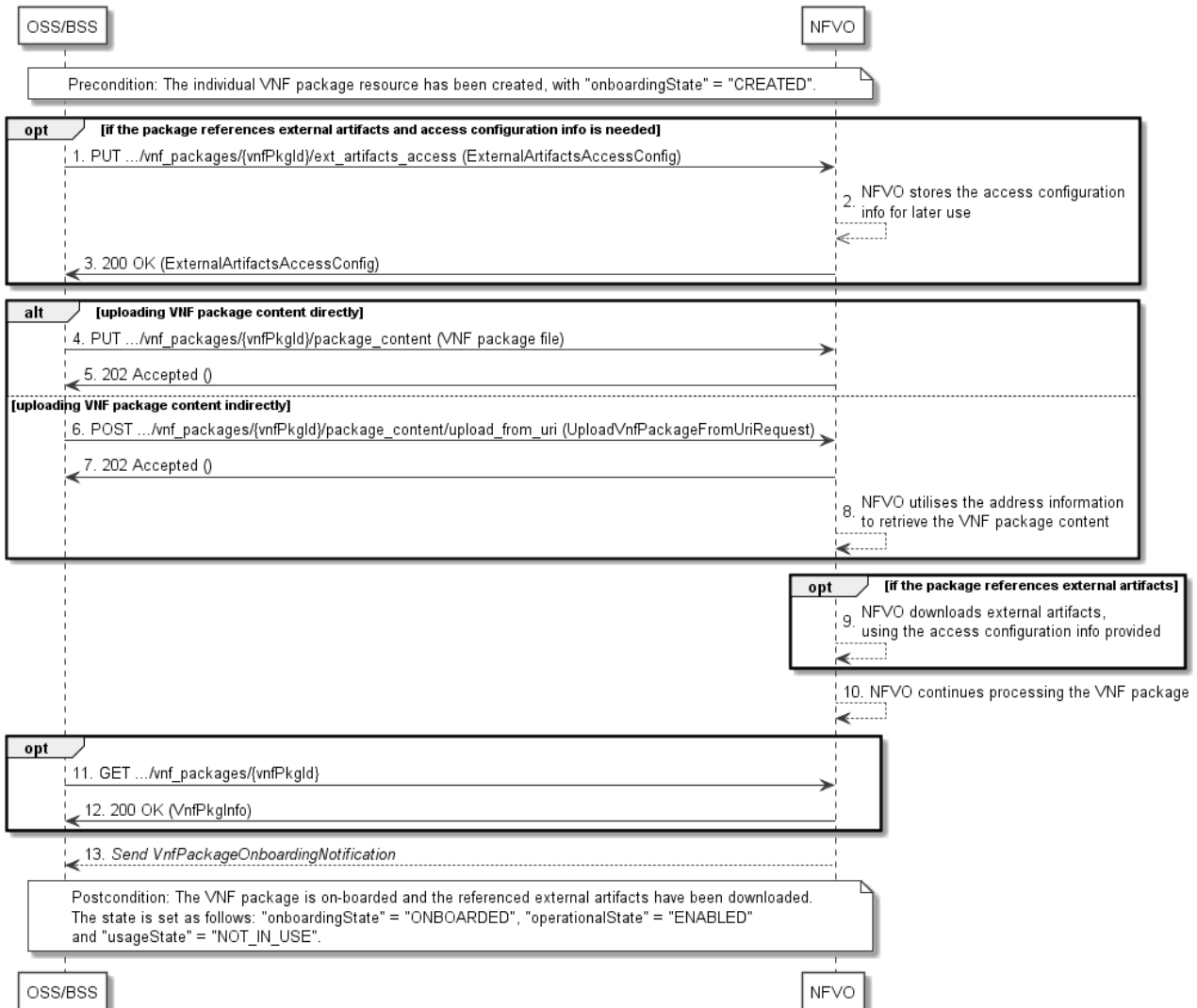
- 1) The OSS/BSS sends a POST request to the "VNF packages" resource including in the payload body a data structure of type "CreateVnfPkgInfoRequest".
- 2) The NFVO creates a new "Individual VNF package" resource.
- 3) The NFVO returns a "201 Created" response containing a representation of the "Individual VNF package" resource and a "Location" HTTP header that points to the new "Individual VNF package" resource.

**Postcondition:** Upon successful completion, the "Individual VNF package" resource is created with the value of the "onboardingState" attribute equals to "CREATED", the value of the "operationalState" attribute equals to "DISABLED" and the value of "usageState" attribute equals to "NOT\_IN\_USE".

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 9.3.2 Flow of the uploading of VNF package content

This clause describes the procedure of uploading the content of a VNF package.



**Figure 9.3.2-1: Flow of the uploading of VNF package content**

**Precondition:** The "Individual VNF package" resource has been created with the value of "onboardingState" attribute equals to "CREATED".

Uploading the content of a VNF package, as illustrated in figure 9.3.2-1, consists of the following steps:

- 1) If the package to be uploaded references external artifacts and access configuration information is not available to the NFVO, the OSS/BSS provides access configuration information for the subsequent download of the external VNF package artifacts to the NFVO by sending a PUT request to the "Access configuration for external artifacts download" resource including in the payload body a data structure of type "ExternalArtifactsAccessConfig".
- 2) In case step 1) was performed, the NFVO stores the information.
- 3) In case step 1) was performed, the NFVO returns a "200 OK" response with a copy of the stored information included in the payload body to indicate that the access configuration information was successfully stored.
- 4) If the OSS/BSS uploads the VNF package content directly to the NFVO, it sends a PUT request to the "VNF package content" resource including in the payload body a copy of the VNF package content. When the NFVO receives the PUT request, it sets the "onboardingState" to "UPLOADING". The NFVO then continues to receive the package data which might be a lengthy operation, depending in the size of the package.

- 5) The NFVO returns a "202 Accepted" response with an empty payload body.
- 6) If the OSS/BSS uploads the VNF package content indirectly to the NFVO, it sends a POST request to the "Upload VNF package from URI task" resource including in the payload body a data structure of type "UploadVnfPackageFromUriRequest".
- 7) The NFVO sets the "onboardingState" to "UPLOADING" and returns a "202 Accepted" response with an empty payload body to indicate the address information is successfully received.
- 8) The NFVO utilizes the address information to retrieve the VNF package content.
- 9) If the package references external artifacts, the NFVO utilizes the access configuration information and the information in the VNF package to download the external artifacts.
- 10) The NFVO sets the "onboardingState" to "PROCESSING". It then continues processing the VNF package (e.g. validation) after it has received the package content.
- 11) Optionally, the OSS/BSS can send a GET request to the "Individual VNF package" resource to check the onboarding state of the "Individual VNF package" resource.
- 12) The NFVO returns a "200 OK" response containing the information of the VNF package resource.
- 13) After successful processing, the NFVO sends a VnfPackageOnboardingNotification to the OSS/BSS to indicate the successful on-boarding of the VNF package content.

**Postcondition:** Upon successful completion, the content of the VNF package is on-boarded and the referenced external artifacts have been downloaded. The state of the VNF package is changed as follows: the value of the "onboardingState" attribute equals to "ONBOARDED", the value of the "operationalState" attribute equals to "ENABLED" and the value of the "usageState" attribute equals to "NOT\_IN\_USE".

**Error handling:** In case of failure, appropriate error information is provided in the response.

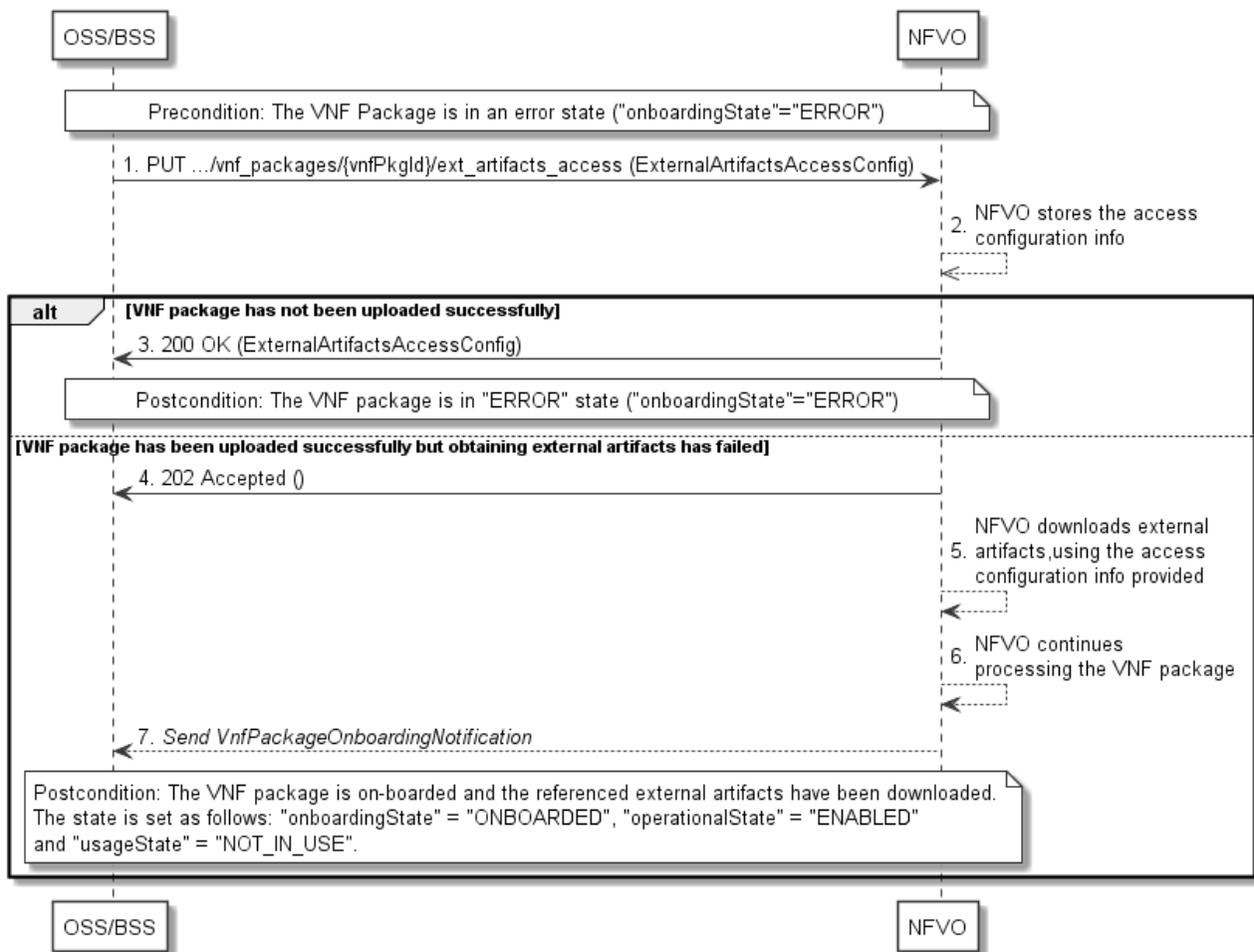
### 9.3.2a Flow of handling errors during upload of VNF package content

This clause describes how to handle errors during the uploading and processing of the content of a VNF package or external artifacts.

In case a long-running operation has failed, such as uploading of the VNF package, obtaining of external artifacts or processing/validation of the package, the "onboardingState" of the VNF package is set to "ERROR". Also, appropriate error information is provided in the "onboardingFailureDetails" attribute.

To handle the error, the OSS/BSS has the following options:

- 1) Delete the "Individual VNF package" resource and create a new one.
- 2) Upload the VNF package again, or upload a modified VNF package, by restarting the flow in clause 9.3.2 at step 4) or step 6).
- 3) Provide new or updated access configuration information for the external artifacts as illustrated in figure 9.3.2a-1.



**Figure 9.3.2a-1: Flow of re-downloading external VNF package artifacts in error case**

**Precondition:** The VNF package is in an error state.

Providing new or updated access configuration information for the external artifacts, is handled in error state as illustrated in figure 9.3.2a-1 and consists of the following steps:

- 1) The OSS/BSS provides access configuration information for the subsequent download of the external VNF package artifacts to the NFVO by sending a PUT request to the "Access configuration for external artifacts download" resource including in the payload body a data structure of type "ExternalArtifactsAccessConfig".
- 2) The NFVO stores the information.

In case the VNF package has not been uploaded successfully previously:

- 3) The NFVO returns a "200 OK" response with a copy of the stored information included in the payload body to indicate that the access configuration information was successfully stored.

In case the VNF package has been uploaded successfully previously:

- 4) The NFVO sets the "onboardingState" to "UPLOADING" and returns a "202 Accepted" response with an empty payload body.
- 5) The NFVO utilizes the access configuration information and the information in the VNF package to retry the download of those external artifacts that were not successfully downloaded previously.
- 6) The NFVO sets the "onboardingState" to "PROCESSING". It then continues processing the VNF package (e.g. validation).
- 7) After successful processing, the NFVO sends a VnfPackageOnboardingNotification to the OSS/BSS to indicate the successful on-boarding of the VNF package content.

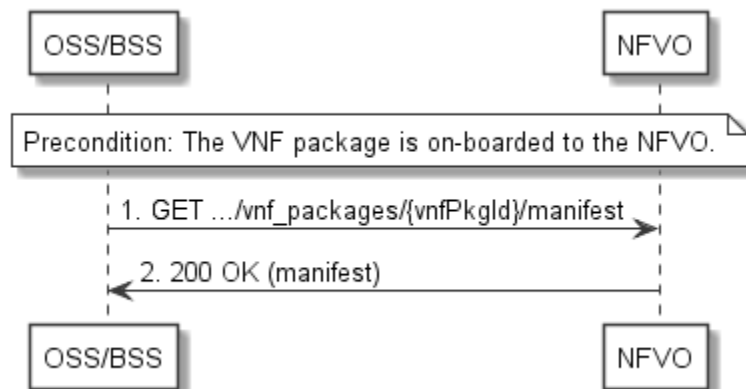
**Postcondition:**

- In case the VNF package was not successfully downloaded previously, the VNF package stays in "ERROR" state.
- In case the VNF package was successfully downloaded previously and this procedure was completed successfully, the content of the VNF package is on-boarded and the referenced external artifacts have been downloaded. The state of the VNF package is changed as follows: the value of the "onboardingState" attribute equals to "ONBOARDED", the value of the "operationalState" attribute equals to "ENABLED" and the value of the "usageState" attribute equals to "NOT\_IN\_USE".

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 9.3.2b Flow of fetching the VNF package manifest

This clause describes a sequence for fetching the VNF package manifest.



**Figure 9.3.2b-1: Flow of fetching the VNF package manifest**

**Precondition:** The VNF package is on-boarded to the NFVO.

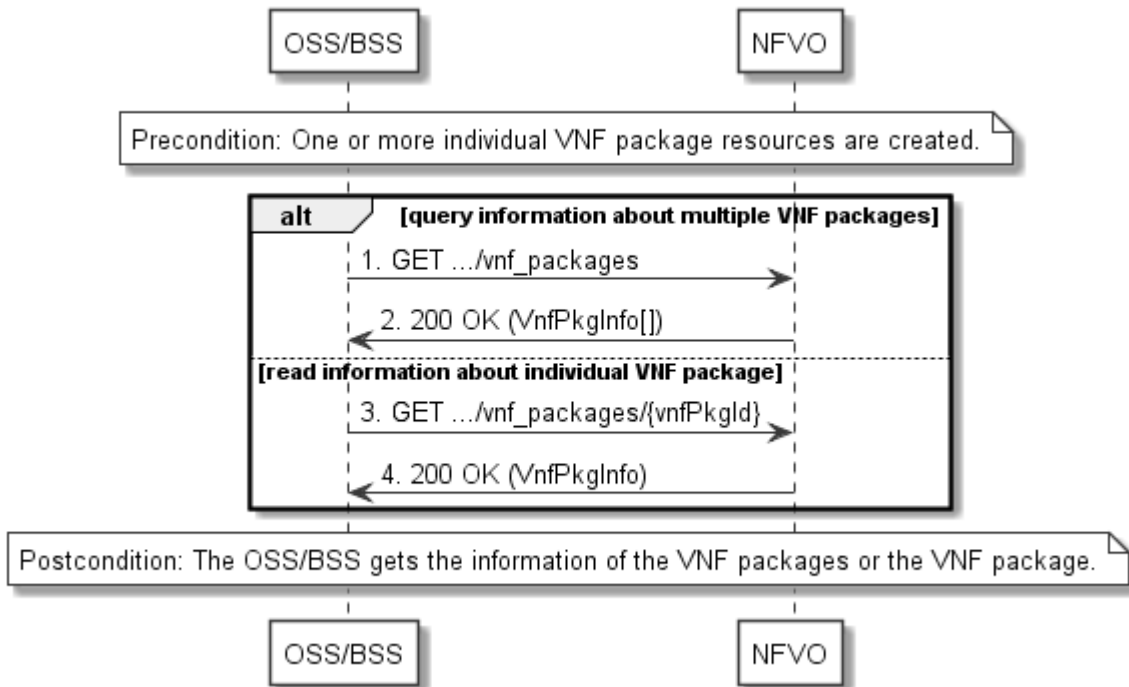
Reading the manifest of an on-boarded VNF package, as illustrated in figure 9.3.2b-1, consists of the following steps:

- 1) The OSS/BSS sends a GET request to the "Manifest of an individual VNF package" resource.
- 2) The NFVO returns a "200 OK" response with a payload body that contains a copy of the manifest file in the VNF package.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 9.3.3 Flow of querying/reading VNF package information

This clause describes a sequence for querying information about one or multiple VNF packages.



**Figure 9.3.3-1: Flow of querying/reading VNF package information**

**Precondition:** One or more individual VNF package resources are created.

VNF package information query, as illustrated in figure 9.3.3-1, consists of the following steps:

- 1) If the OSS/BSS intends to query information about multiple VNF packages, it sends a GET request to the "VNF packages" resource.
- 2) The NFVO returns a "200 OK" response, and includes in the payload body zero or more data structures of type "VnfPkgInfo".
- 3) If the OSS/BSS intends to read information about a particular VNF package, the OSS/BSS sends a GET request to the "Individual VNF package" resource, addressed by the appropriate VNF package identifier in its resource URI.
- 4) The NFVO returns a "200 OK" response, and includes in the payload body a data structure of type "VnfPkgInfo".

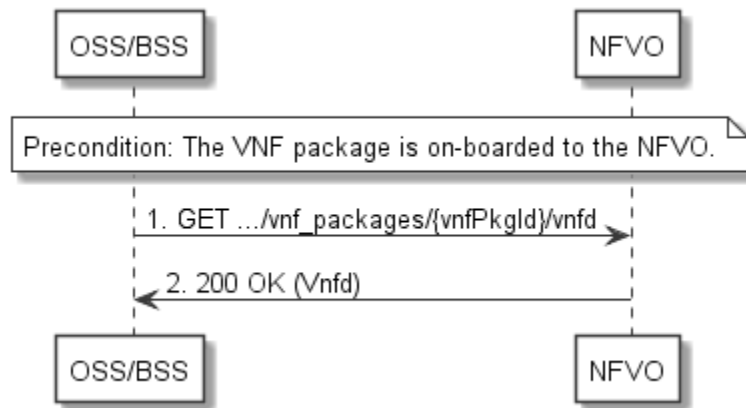
**Postcondition:** Upon successful completion, the OSS/BSS gets the information of the VNF packages or the VNF package.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 9.3.4 Flow of reading the VNFD of an on-boarded VNF package

This clause describes the procedure for reading the VNFD of an on-boarded VNF package.





**Figure 9.3.4-1: Flow of reading VNFD**

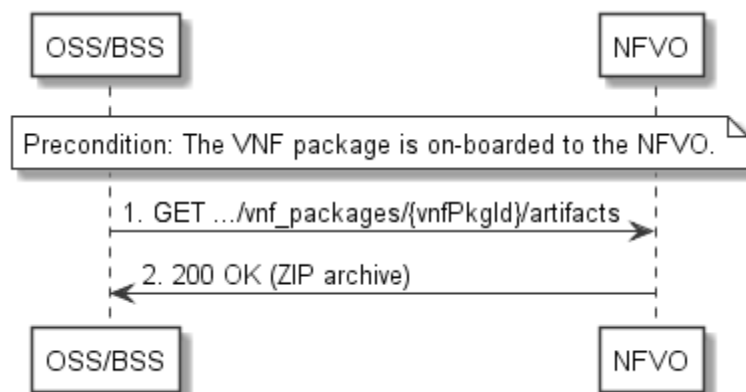
**Precondition:** The VNF package is on-boarded to the NFVO.

The procedure consists of the following steps as illustrated in figure 9.3.4-1:

- 1) The OSS/BSS sends a GET request to the "VNFD of an individual VNF package" resource.
- 2) The NFVO returns a "200 OK" response, and includes a copy of the VNFD from the VNF package in the payload body.

### 9.3.4a Flow of bulk-fetching VNF package artifacts that are not images

This clause describes a sequence for bulk-fetching artifacts that are not images from an on-boarded VNF package.



**Figure 9.3.4a-1: Flow of bulk-fetching VNF package artifacts**

**Precondition:** The VNF package is on-boarded to the NFVO.

Bulk-fetching of artifacts from an on-boarded VNF package allows the OSS/BSS to request all artifacts that are not images, and further may allow to specify additional filtering criteria for the artifacts to be included in that set.

**NOTE:** It is assumed that, due to their size, images are excluded from the bulk fetch operation. These can be fetched as individual artifacts.

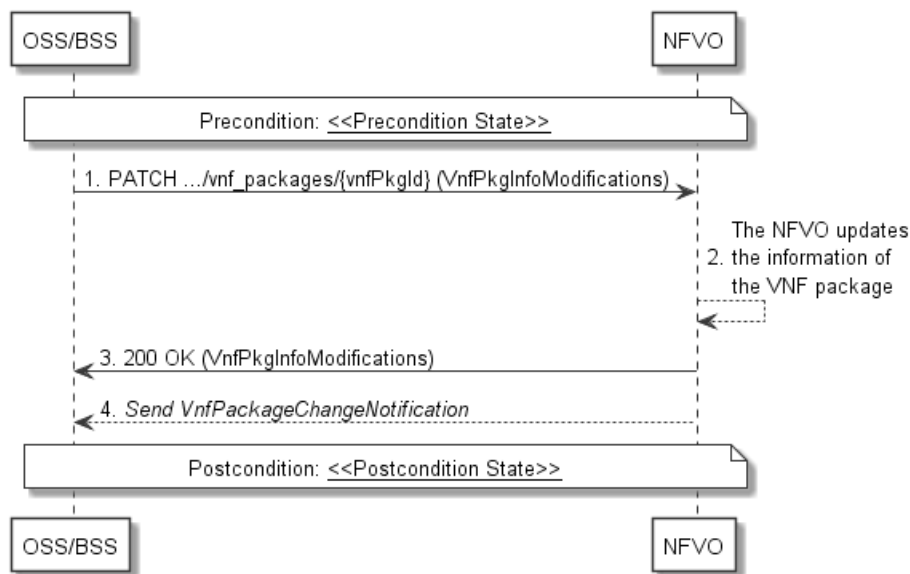
The procedure consists of the following steps, as illustrated in figure 9.3.4a-1:

- 1) The OSS/BSS sends a GET request to the "VNF package artifacts" resource and specifies, if supported, the appropriate URI query parameters to define the requested set of artifacts.
- 2) The NFVO returns a "200 OK" response with a payload body that contains a ZIP archive which contains the applicable artifacts, embedded in the appropriate directory structure in the VNF package.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 9.3.5 Flow of updating information of a VNF package

This clause describes the procedure for enabling/disabling/abort deletion of a VNF package.



**Figure 9.3.5-1: Flow of updating information of a VNF package**

**NOTE:** Due to possible race conditions, the 200 response and the VnfPackageChangeNotification can arrive in any order at the OSS/BSS.

**Precondition:** The VNF package is in <<Precondition State>>, the value of <<Precondition State>> depends on the actual requested operation, and is described in table 9.3.5-1.

The procedure consists of the following steps as illustrated in figure 9.3.5-1:

- 1) The OSS/BSS sends a PATCH request to the "Individual VNF package" resource including in the payload body a data structure of type "VnfPkgInfoModifications".
- 2) The NFVO updates the information of the VNF package.
- 3) The NFVO returns a "200 OK" response with a payload body containing a data structure of type "VnfPkgInfoModifications".
- 4) If the operational state of the VNF package is modified, the NFVO sends to OSS/BSS a VnfPackageChangeNotification to indicate the state change of the VNF package.

**Postcondition:** The VNF package is in << Postcondition State>>.

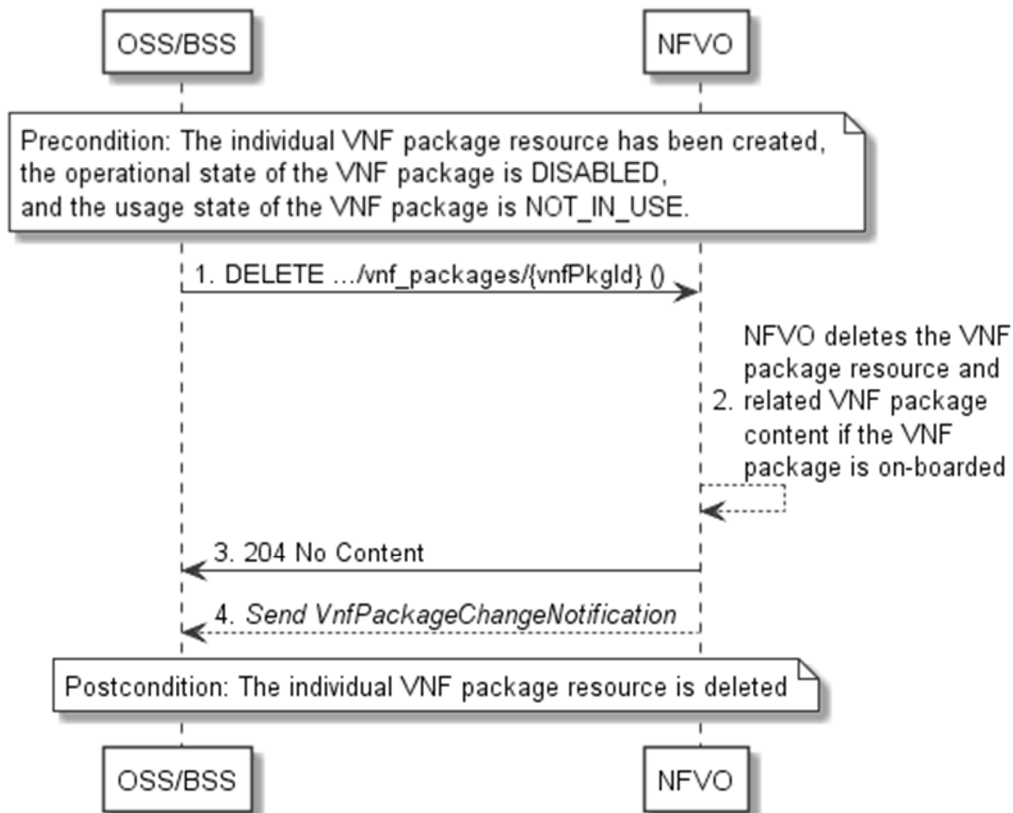
Table 9.3.5-1 describes how the <<Precondition State>> and << Postcondition State>> are parameterized in the above flow.

**Table 9.3.5-1: Parameterization of the flow for updating information of a VNF package**

Operation	<<Precondition State>>	<<PostconditionState>>
Enable a VNF package	The on-boarding state of the VNF package is ONBOARDED and the operational state of the VNF package is DISABLED	The operational state of the VNF package is ENABLED
Disable a VNF package	The on-boarding state of the VNF package is ONBOARDED and the operational state of the VNF package is ENABLED	The operational state of the VNF package is DISABLED
Update user defined data	The individual VNF package resource is created	The user defined data is updated

### 9.3.6 Flow of deleting a VNF package resource

This clause describes a sequence for deleting an "Individual VNF package" resource.



**Figure 9.3.6-1: Flow of deleting a VNF package resource**

**NOTE:** Due to possible race conditions, the 204 response and the `VnfPackageChangeNotification` can arrive in any order at the OSS/BSS.

**Precondition:** The "Individual VNF package" resource has been created, the operational state of the VNF package is DISABLED, and the usage state of the VNF package is NOT\_IN\_USE.

Deleting a VNF package resource, as illustrated in figure 9.3.6-1, consists of the following steps:

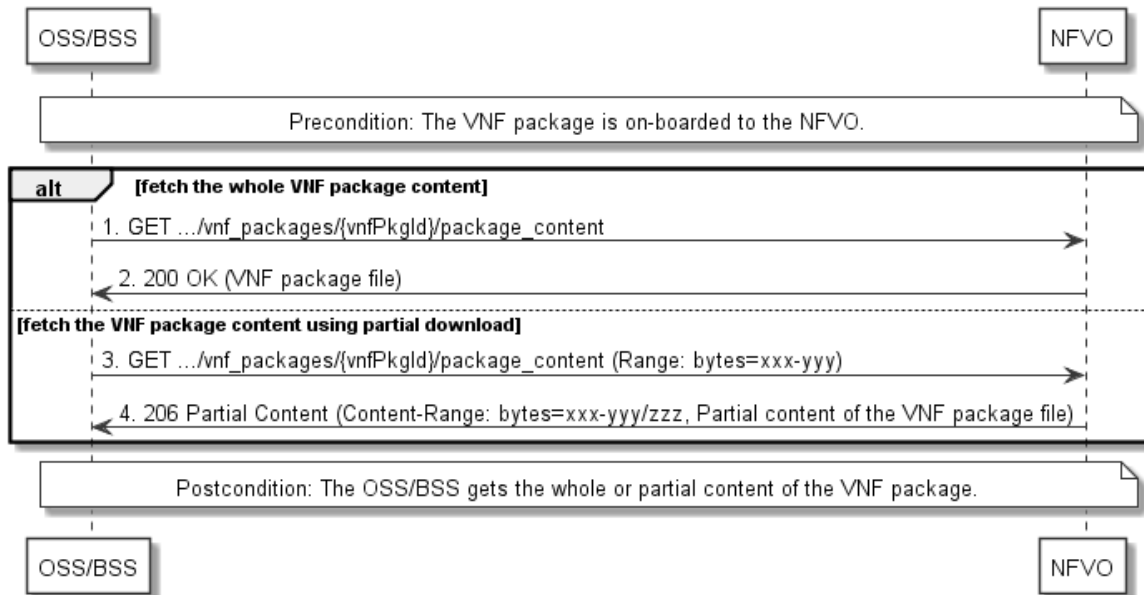
- 1) The OSS/BSS sends a DELETE request to the "Individual VNF package" resource.
- 2) The NFVO deletes the "Individual VNF package" resource and related VNF package content if the VNF package is on-boarded.
- 3) The NFVO returns a "204 No Content" response with an empty payload body.
- 4) The NFVO sends to the OSS/BSS a `VnfPackageChangeNotification` to indicate the deletion of the "Individual VNF package" resource.

**Postcondition:** Upon successful completion, the "Individual VNF package" resource is deleted.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 9.3.7 Flow of fetching an on-boarded VNF package

This clause describes a sequence for fetching the content of an on-boarded VNF package.



**Figure 9.3.7-1: Flow of fetching an on-boarded VNF package**

**Precondition:** The VNF package is on-boarded to the NFVO.

Fetching an on-boarded VNF package, as illustrated in figure 9.3.7-1, consists of the following steps:

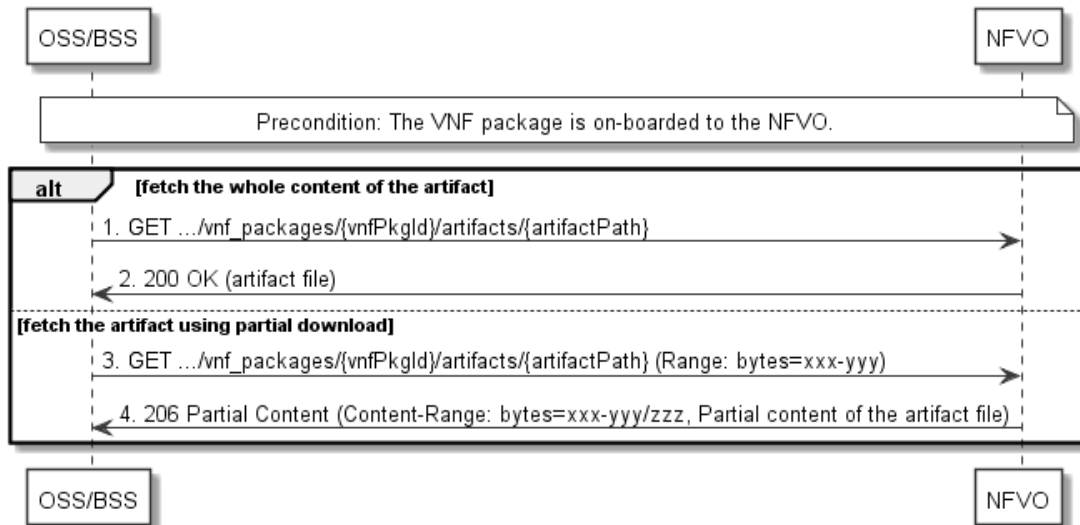
- 1) If fetching the whole VNF package content, the OSS/BSS sends a GET request to the "VNF package content" resource.
- 2) The NFVO returns a "200 OK" response, and includes a copy of the VNF package file in the payload body.
- 3) If fetching the VNF package content using partial download, the OSS/BSS sends a GET request to the "VNF package content" resource, and includes a "Range" HTTP header indicating the partition of the VNF package content needs to be transferred.
- 4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the VNF package, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the VNF package content.

**Postcondition:** Upon successful completion, the OSS/BSS gets the whole or partial content of the VNF package.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 9.3.8 Flow of fetching a VNF package artifact

This clause describes a sequence for fetching an individual artifact contained in an on-boarded VNF package.



**Figure 9.3.8-1: Flow of fetching a VNF package artifact**

**Precondition:** The VNF package is on-boarded to the NFVO.

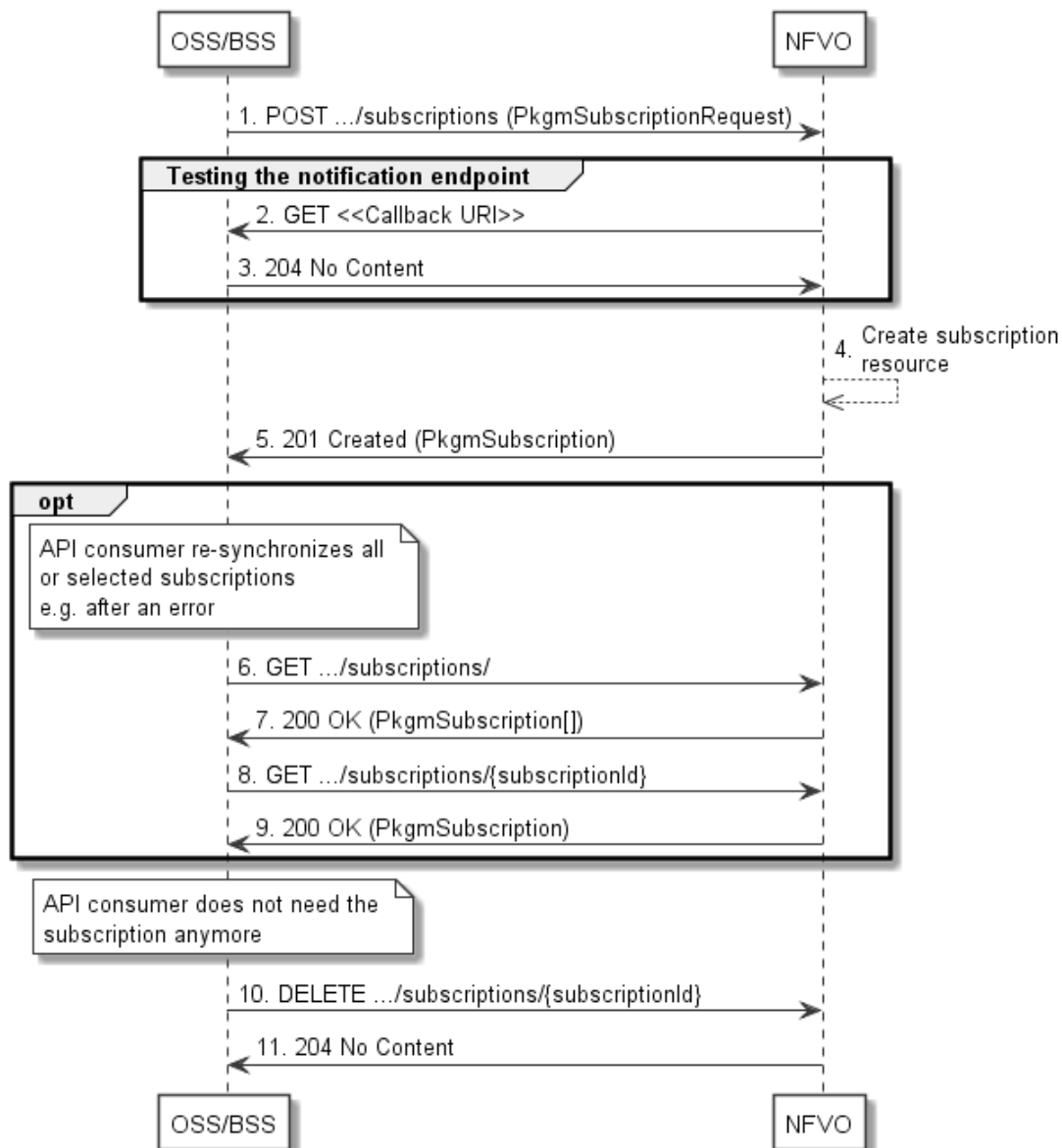
Fetching an individual artifact contained in an on-boarded VNF package, as illustrated in figure 9.3.8-1, consists of the following steps:

- 1) If fetching the whole content of the artifact, the OSS/BSS sends a GET request to the "Individual VNF package artifact" resource.
- 2) The NFVO returns a "200 OK" response, and includes a copy of the applicable artifact file from the VNF package in the payload body.
- 3) If fetching the artifact using partial download, the OSS/BSS sends a GET request to the "Individual VNF package artifact" resource, and includes a "Range" HTTP header indicating the partition of the artifact needs to be transferred.
- 4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the artifact file, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the artifact file.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 9.3.9 Flow of managing subscriptions

This clause describes the procedure for creating, reading and terminating subscriptions to notifications related to VNF package management.



**Figure 9.3.9-1: Flow of managing subscriptions**

The procedure consists of the following steps as illustrated in figure 9.3.9-1:

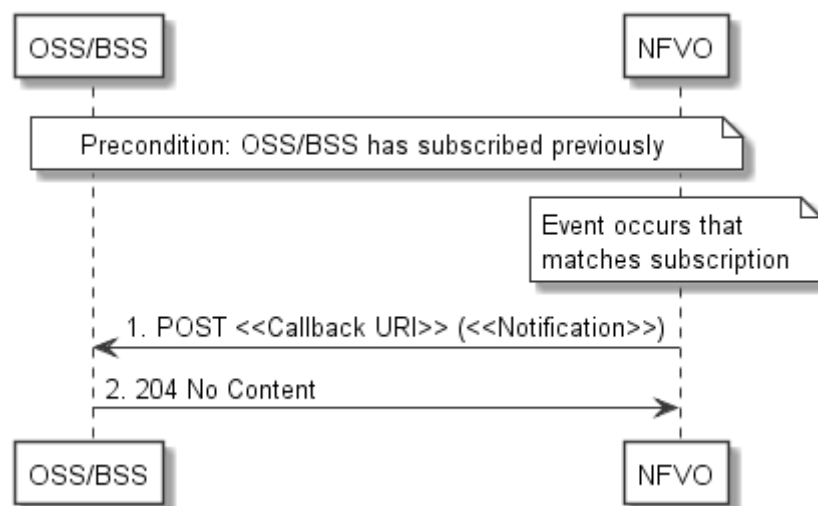
- 1) The OSS/BSS sends a POST request to the "Subscriptions" resource including in the payload body a data structure of type "PkgmSubscriptionRequest". That data structure contains filtering criteria and a callback URI to which the NFVO will subsequently send notifications about events that match the filter.
- 2) To test the notification endpoint that has been registered by the OSS/BSS as part of the subscription, the NFVO sends a GET request to the notification endpoint URI.
- 3) The OSS/BSS returns a "204 No Content" response to indicate success.
- 4) The NFVO creates a new subscription to notifications related to VNF package on-boarding or changes, and a resource that represents this subscription.
- 5) The NFVO returns a "201 Created" response containing a data structure of type "PkgmSubscription" representing the "Individual subscription" resource just created by the NFVO, and provides the URI of the newly-created resource in the "Location" HTTP header.
- 6) If desired, e.g. to recover from an error situation, the OSS/BSS can obtain information about its subscriptions by sending a GET request to the resource representing the subscriptions.

- 7) In that case, the NFVO returns a "200 OK" response that contains the list of representations of all existing subscriptions that were created by the OSS/BSS.
- 8) If desired, e.g. to recover from an error situation, the OSS/BSS can obtain information about a particular subscription by sending a GET request to the resource representing that individual subscription.
- 9) In that case, the NFVO returns a "200 OK" response that contains a representation of that individual subscription.
- 10) If the OSS/BSS does not need the subscription anymore, it terminates the subscription by sending a DELETE request to the resource that represents the individual subscription to remove.
- 11) The NFVO acknowledges the successful termination of the subscription by returning a "204 No Content" response.

**Error handling:** The NFVO rejects a subscription if the subscription information is not valid: endpoint cannot be reached, subscription information is malformed, etc.

### 9.3.10 Flow of sending notifications

This clause describes the procedure for sending notifications related to VNF package management.



**Figure 9.3.10-1: Flow of sending notifications**

**Precondition:** The OSS/BSS has subscribed previously for notifications related to VNF package management.

The procedure consists of the following steps as illustrated in figure 9.3.10-1:

- 1) If an event occurs that matches the filtering criteria which are part of the subscription, the NFVO generates a notification that includes information about the event, and sends it in the body of a POST request to the URI which the OSS/BSS has registered as part of the subscription request. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API (see clauses 9.5.2.8 and 9.5.2.9).
- 2) The OSS/BSS acknowledges the successful delivery of the notification by returning a "204 No Content" response.

**Error handling:** If the NFVO does not receive the "204 No Content" response from the OSS/BSS, it can retry sending the notification.

## 9.4 Resources

### 9.4.1 Introduction

This clause defines all the resources and methods provided by the VNF package management interface.

#### 9.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [16] are part of the VNF package management interface.

### 9.4.2 Resource: VNF packages

#### 9.4.2.1 Description

This resource represents VNF packages. The API consumer can use this resource to create individual VNF package resources, and to query information of the VNF packages.

#### 9.4.2.2 Resource definition

The resource URI is:

**{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf\_packages**

This resource shall support the resource URI variables defined in table 9.4.2.2-1.

**Table 9.4.2.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.

#### 9.4.2.3 Resource methods

##### 9.4.2.3.1 POST

The POST method creates a new individual VNF package resource.

Upon successful creation of the "Individual VNF package" resource, the NFVO shall set the "onboardingState" attribute in the "VnPkgInfo" structure to "CREATED", the "operationalState" attribute to "DISABLED", and the "usageState" attribute to "NOT\_IN\_USE".

This method shall follow the provisions specified in the tables 9.4.2.3.1-1 and 9.4.2.3.1-2 for URI query parameters, request and response data structures, and response codes.

**Table 9.4.2.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		



Table 9.4.2.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	CreateVnfPkgInfoRequest		1	Individual VNF package resource creation parameters, as defined in clause 9.5.2.2.
Response body	Data type	Cardinality	Response Codes	Description
	VnfPkgInfo	1	201 Created	<p>Shall be returned when an "Individual VNF package" resource has been created successfully.</p> <p>The response body shall contain a representation of the new individual VNF package resource, as defined in clause 9.5.2.5.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the individual VNF package resource.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 9.4.2.3.2 GET

The GET method queries the information of the VNF packages matching the filter.

This method shall follow the provisions specified in the tables 9.4.2.3.2-1 and 9.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>All attribute names that appear in the VnfPkgInfo and in data types referenced from it shall be supported by the NFVO in the filter expression.</p>
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_default	0..1	<p>Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details.</p> <p>The NFVO shall support this parameter.</p> <p>The following attributes shall be excluded from the VnfPkgInfo structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided:</p> <ul style="list-style-type: none"> <li>- softwareImages</li> <li>- additionalArtifacts</li> <li>- userDefinedData</li> <li>- checksum</li> <li>- onboardingFailureDetails</li> </ul>
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 9.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	VnfPkgInfo	0..N	200 OK	<p>Shall be returned when information about zero or more VNF packages has been queried successfully.</p> <p>The response body shall contain in an array the VNF package info representations that match the attribute filter, i.e. zero or more VNF package info representations as defined in clause 9.5.2.5.</p> <p>If the "filter" URI parameter or one of the "all_fields", "fields", "exclude_fields" or "exclude_default" URI parameters was supplied in the request and is supported, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 and 5.3.2 of ETSI GS NFV-SOL 013 [16], respectively.</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute selector.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>

#### 9.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 9.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 9.4.3 Resource: Individual VNF package

### 9.4.3.1 Description

This resource represents an individual VNF package. The API consumer can use this resource to read information of the VNF package, update information of the VNF package, or delete a VNF package.

### 9.4.3.2 Resource definition

The resource URI is:

**{apiRoot}/vnfpgm/{apiMajorVersion}/vnf\_packages/{vnfPkgId}**

This resource shall support the resource URI variables defined in table 9.4.3.2-1.

**Table 9.4.3.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "VnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.

### 9.4.3.3 Resource methods

#### 9.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.3.3.2 GET

The GET method reads the information of an individual VNF package.

This method shall follow the provisions specified in the tables 9.4.3.3.2-1 and 9.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 9.4.3.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 9.4.3.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	VnfPkgInfo	1	200 OK	<p>Shall be returned when information of the VNF package has been read successfully.</p> <p>The response body shall contain the VNF package info representation defined in clause 9.5.2.5.</p>

	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.
--	----------------	------------------------	---------	---

#### 9.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.3.3.4 PATCH

The PATCH method updates the information of a VNF package.

This method shall follow the provisions specified in the tables 9.4.3.3.4-1 and 9.4.3.3.4-2 for URI query parameters, request and response data structures, and response codes.

**Table 9.4.3.3.4-1: URI query parameters supported by the PATCH method on this resource**

Name	Cardinality	Description
none supported		

**Table 9.4.3.3.4-2: Details of the PATCH request/response on this resource**

	Data type	Cardinality	Description	
	Request body	VnfPkgInfoModifications	1	Parameters for VNF package information modifications.  The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [11].
Response body	Data type	Cardinality	Response Codes	Description
	VnfPkgInfoModifications	1	200 OK	Shall be returned when the operation has been accepted and completed successfully.  The response body shall contain attribute modifications for an "Individual VNF package" resource.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.  Typically, this is due to any of the following scenarios: <ul style="list-style-type: none"> <li>- Disable a VNF package resource of which the operational state is not "ENABLED"</li> <li>- Enable a VNF package resource of which the operational state is not "DISABLED"</li> </ul> The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 9.4.3.3.5 DELETE

The DELETE method deletes an individual VNF package resource.

A VNF Package may only be deleted once:

- there are no VNFs using it,
- there are no NSs using it,
- there are no NSDs referencing to it which have a value of "strict NSD constituent onboarding" set to True, and
- the operational state is DISABLED.

**NOTE:** An NS instance can be using a VNF Package even if no VNF instance is using it if the VNF is part of the selected NS deployment flavour but the current NS scale level determines zero instances for the VNF.

This method shall follow the provisions specified in the tables 9.4.3.3.5-1 and 9.4.3.3.5-2 for URI query parameters, request and response data structures, and response codes.

**Table 9.4.3.3.5-1: URI query parameters supported by the DELETE method on this resource**

Name	Cardinality	Description
none supported		

**Table 9.4.3.3.5-2: Details of the DELETE request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the VNF package has been deleted successfully.  The response body shall be empty.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.  Typically, this is due to the fact that the operational state of the VNF package resource is "ENABLED", or the usage state is "IN_USE" (i.e. "Individual VNF instance" resource created from the concerned VNF package exists).  The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 9.4.4 Resource: VNFD in an individual VNF package

### 9.4.4.1 Description

This resource represents the VNFD contained in an on-boarded VNF package. The API consumer can use this resource to obtain the content of the VNFD.

### 9.4.4.2 Resource definition

The resource URI is:

**{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf\_packages/{vnfPkgId}/vnfd**

This resource shall support the resource URI variables defined in table 9.4.4.2-1.

**Table 9.4.4.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.

### 9.4.4.3 Resource methods

#### 9.4.4.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.4.3.2 GET

The GET method reads the content of the VNFD within a VNF package.

The VNFD is implemented as a collection of one or more files. A ZIP archive embedding these files shall be returned when reading this resource.

The default format of the ZIP archive shall comply with the CSAR format as specified in ETSI GS NFV-SOL 004 [5] where only the files representing the VNFD and information needed to navigate the ZIP archive and to identify the file that is the entry point for parsing the VNFD and (if requested) further security information are included, and software images as well as other artifacts referenced from the YAML files are excluded. This means that the structure of the ZIP archive shall correspond to the directory structure used in the VNF package and that the archive shall contain the following files from the package:

- TOSCA.meta (if available in the package).
- The main TOSCA definitions YAML file (either as referenced by the Entry-Definitions keyword from TOSCA.meta or available as a file with the extension ".yaml" or ".yml" from the root of the archive).
- Other TOSCA YAML files, if any, as referenced by the Other-Definitions keyword from TOSCA.meta.
- Every component of the VNFD referenced (recursively) from the YAML files as mentioned above.

NOTE 1: For a VNFD based on TOSCA, it includes all the imported type definition files as indicated in the top level service template and in any of the lower level service template if it has any as described in ETSI GS NFV-SOL 001 [i.1].

NOTE 2: For a VNFD based on YANG, it includes the file as indicated by the "yang\_definitions" keyname in the metadata section of the main yaml file as described in ETSI GS NFV-SOL 004 [5].

- The related security information, if the "include\_signatures" URI parameter is provided, as follows:
  - the manifest file;
  - the singleton certificate file in the root of the VNF package (if available in the package);
  - the signing certificates of the individual files included in the ZIP archive (if available in the package);
  - the signatures of the individual files (if available in the package).

Three examples are provided below.

NOTE 3: These examples do not show the security related files.

EXAMPLE 1: Assuming a request is sent for the following VNF package (as described in clause A.1 in ETSI GS NFV-SOL 004 [5]):

```
!-----TOSCA-Metadata
!-----TOSCA.meta (metadata for navigating the ZIP file)

!-----Definitions
!----- MRF.yaml (main VNFD file)
!----- OtherTemplates (e.g. type definitions referenced by the main VNFD file or any
files indicated in the Other-Definitions keyword in the TOSCA.meta file if
it has any)

!-----Files
!----- ChangeLog.txt
!----- image(s)
!----- other artifacts

!-----Tests
!----- file(s)

!-----Licenses
!----- file(s)

!-----Scripts
!----- install.sh
!----- MRF.mf
```

The NFVO will return a ZIP file of the following format:

```
!-----TOSCA-Metadata
!-----TOSCA.meta

!-----Definitions
!----- MRF.yaml
!----- OtherTemplates
```

EXAMPLE 2: Assuming a request is sent for the following VNF package (a VNF package without a TOSCA-Metadata directory, as described in clause A.2 in ETSI GS NFV-SOL 004 [5]):

```
!-----MRF.yaml (main VNFD file)
!-----MRF.mf
!-----ChangeLog.txt

!-----Tests
!----- file(s)

!-----Licenses
!----- file(s)

!-----Artifacts
!----- install.sh
!----- start.yang
```

The NFVO will return a ZIP file of the following format:

```
!-----MRF.yaml
```

EXAMPLE 3: Assuming a request is sent for the following VNF package (a VNF package with the YANG VNFD without a TOSCA-Metadata directory, as described in clause A.3 in ETSI GS NFV-SOL 004 [5]):

```
!-----CompanyVNFD.yaml
!-----CompanyVNFD.xml
!-----CompanyVNFD.mf
!-----ChangeLog.txt

!-----Files

!-----Instance Data Files
!-----start.xml
```

```
!-----Licenses
!-----Scripts
!-----install.sh
```

The NFVO will return a ZIP file of the following format:

```
!----CompanyVNFD.yaml
!----CompanyVNFD.xml (indicated in the yang_definitions metadata in CompanyVNFD.yaml)
```

This method shall follow the provisions specified in the tables 9.4.4.3.2-1 and 9.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 9.4.4.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
include_signatures	0..1	If this parameter is provided, the NFVO shall include in the ZIP archive the security information as specified above.  This URI query parameter is a flag, i.e. it shall have no value.  The NFVO shall support this parameter.

**Table 9.4.4.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	Shall be returned when the content of the VNFD has been read successfully.  The payload body shall contain a ZIP archive that contains the files representing the VNFD, as specified above.  The "Content-Type" HTTP header shall be set to "application/zip".
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.  Typically, this is due to the fact that "onboardingState" of the VNF package has a value different from "ONBOARDED".  The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 9.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].



#### 9.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 9.4.4a Resource: Access configuration for external artifacts

#### 9.4.4a.1 Description

This resource represents access configuration information for the download of external artifacts. The OSS/BSS can use this resource to provide required credentials and address information for the downloading of external artifacts to the NFVO, enabling to retrieve the artifacts. The OSS/BSS can also read the information and can overwrite wrong information to trigger a retry of unsuccessful downloads.

#### 9.4.4a.2 Resource definition

The resource URI is:

**{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf\_packages/{vnfPkgId}/ext\_artifacts\_access**

This resource shall support the resource URI variables defined in table 9.4.4a.2-1.

**Table 9.4.4a.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new Individual VNF package resource.

#### 9.4.4a.3 Resource methods

##### 9.4.4a.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

##### 9.4.4a.3.2 GET

The GET method reads the access configuration information that is used by the NFVO to get the content of external VNF package artifacts.

This method shall follow the provisions specified in the tables 9.4.4a.3.2-1 and 9.4.4a.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 9.4.4a.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

Table 9.4.4a.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	ExternalArtifactsAccessConfig	1	200 OK	Shall be returned when the access configuration information has been read successfully. If no information has been provided previously, an empty array shall be returned.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 9.4.4a.3.3 PUT

The PUT method provides the access configuration information for the NFVO to download the content of external VNF package artifacts.

As precondition for invoking this method, the individual VNF package resource shall have been created, and the value of "onboardingState" attribute shall equal to "CREATED" or "ERROR".

The resource representation in the payload body of the PUT request shall replace the current state of the resource.

This method shall follow the provisions specified in the tables 9.4.4a.3.3-1 and 9.4.4a.3.3-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.4a.3.3-1: URI query parameters supported by the PUT method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.4a.3.3-2: Details of the PUT request/response on this resource

Request body	Data type	Cardinality	Description	
	ExternalArtifactsAccessConfig	1	The payload body contains the access configuration information based on which the NFVO can obtain the external VNF package artifact files.	
Response body	Data type	Cardinality	Response Codes	Description
	ExternalArtifactsAccessConfig	1	200 OK	<p>Shall be returned when the attribute "onboardingState" = "CREATED" or when the attribute "onboardingState" = "ERROR" and the VNF package has <b>not</b> been uploaded successfully previously, to indicate that the access configuration information has been stored successfully by the NFVO.</p> <p>The response body shall contain a representation of the state of the resource.</p>
	n/a		202 Accepted	<p>Shall be returned when the attribute "onboardingState" = "ERROR" and the VNF package has been uploaded successfully previously, to indicate that the access configuration information has been stored successfully by the NFVO and the NFVO now starts downloading external artifacts.</p> <p>The response body shall be empty.</p> <p>Prior to returning the 202 response, the NFVO shall set the "onboardingState" attribute to "UPLOADING".</p> <p>Subsequently to returning the 202 response, the NFVO shall retry the download of those external artifacts that have failed downloading, or that were downloaded successfully previously and for which modified access configuration information has been provided in the request.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the "onboardingState" attribute contains a value different from "CREATED" or "ERROR".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

#### 9.4.4a.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.4a.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 9.4.4b Resource: Manifest in an individual VNF package

### 9.4.4b.1 Description

This resource represents the manifest contained in an on-boarded VNF package. The API consumer can use this resource to obtain the content of the manifest.

### 9.4.4b.2 Resource definition

The resource URI is:

**{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf\_packages/{vnfPkgId}/manifest**

This resource shall support the resource URI variables defined in table 9.4.4b.2-1.

**Table 9.4.4b.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.

### 9.4.4b.3 Resource methods

#### 9.4.4b.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.4b.3.2 GET

The GET method reads the content of the manifest within a VNF package.

This method shall follow the provisions specified in the tables 9.4.4b.3.2-1 and 9.4.4b.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 9.4.4b.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
include_signatures	0..1	<p>If this parameter is provided, the NFVO shall return the manifest and related security information (such as certificate) in a ZIP archive.</p> <p>If this parameter is not given, the NFVO shall provide only a copy of the manifest file.</p> <p>This URI query parameter is a flag, i.e. it shall have no value.</p> <p>The NFVO shall support this parameter.</p>

Table 9.4.4b.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
n/a				
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the content of the manifest has been read successfully.</p> <p>If the "include_signatures" URI query parameter was absent in the request, or if the manifest file has all security-related information embedded (i.e. there is no separate certificate file), the payload body shall contain a copy of the manifest file of the VNF package, and the "Content-Type" HTTP header shall be set to "text/plain".</p> <p>If the "include_signatures" URI query parameter was present in the related request and the manifest file does not have all the security-related information embedded (i.e. there is a separate certificate file), the "Content-Type" HTTP header shall be set to "application/zip" and the payload body shall contain a ZIP archive which includes:</p> <ul style="list-style-type: none"> <li>- a copy of the manifest file of the VNF package;</li> <li>- a copy of the related individual certificate file.</li> </ul>
	ProblemDetails	0..1	406 Not Acceptable	<p>If the related request contained an "Accept" header not compatible with the Content type "application/zip" but the "include_signatures" flag was provided, the NFVO shall respond with this response code.</p> <p>The "ProblemDetails" structure may be included with the "detail" attribute providing more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "onboardingState" of the VNF package has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 9.4.4b.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 9.4.4b.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 9.4.4b.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 9.4.5 Resource: VNF package content

### 9.4.5.1 Description

This resource represents the content of VNF package identified by the VNF package identifier allocated by the NFVO. The API consumer can use this resource to fetch and to upload the content of the VNF package.

### 9.4.5.2 Resource definition

The resource URI is:

**{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf\_packages/{vnfPkgId}/package\_content**

This resource shall support the resource URI variables defined in table 9.4.5.2-1.

**Table 9.4.5.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.

### 9.4.5.3 Resource methods

#### 9.4.5.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.5.3.2 GET

The GET method fetches the content of a VNF package identified by the VNF package identifier allocated by the NFVO.

The content of the package is provided as onboarded, i.e. depending on the security option used, the CSAR or the CSAR wrapped in a ZIP archive together with an external signature is returned, as defined in clause 5.1 of ETSI GS NFV-SOL 004 [5].

NOTE: Information about the applicable security option can be obtained by evaluating the "packageSecurityOption" attribute in the "VnfPkgInfo" structure.

This method shall follow the provisions specified in the tables 9.4.5.3.2-1 and 9.4.5.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 9.4.5.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

Table 9.4.5.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			<p>The request may contain a "Range" HTTP header to obtain single range of bytes from the VNF package file. This can be used to continue an aborted transmission.</p> <p>If the Range header is present in the request and the NFVO does not support responding to range requests with a 206 response, it shall return a 200 OK response instead as defined below.</p>
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the whole content of the VNF package file has been read successfully.</p> <p>The response body shall include a copy of the VNF package file.</p> <p>The "Content-Type" HTTP header shall be set according to the type of the file, i.e. to "application/zip" for a VNF Package as defined in ETSI GS NFV-SOL 004 [5].</p>
	n/a	1	206 Partial Content	<p>If the NFVO supports range requests, this response shall be returned when a single consecutive byte range from the content of the VNF package file has been read successfully according to the request.</p> <p>The response body shall contain the requested part of the VNF package file.</p> <p>The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [10].</p> <p>The "Content-Type" HTTP header shall be set as defined above for the "200 OK" response.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "onboardingState" of the VNF package has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	0..1	416 Range Not Satisfiable	<p>Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the VNF package file (e.g. "access after end of file").</p> <p>The response body may contain a ProblemDetails structure.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>	

### 9.4.5.3.3 PUT

The PUT method uploads the content of a VNF package.

This method shall follow the provisions specified in the tables 9.4.5.3.3-1 and 9.4.5.3.3-2 for URI query parameters, request and response data structures, and response codes.

Upon start of the upload of the package, the NFVO shall set the "onboardingState" attribute in the "VnfPkgInfo" structure to "UPLOADING". Upon successful upload of the package, if the package references external artifacts, the NFVO shall obtain the external artifacts. Subsequently, upon success, the NFVO shall set that attribute to "PROCESSING" and shall process the package, which shall include checking package consistency. Upon successful processing, the NFVO shall set the "onboardingState" attribute to "ONBOARDED", the "operationalState" attribute to "ENABLED", and the "usageState" attribute to "NOT\_IN\_USE". In addition, the NFVO shall set the value of the attributes in the "VnfPkgInfo" that are copied from the VNFD (refer to clause 9.5.2.5). If an error occurs during uploading the package, downloading the external artifacts or processing the package, the NFVO shall set the "onboardingState" attribute to "ERROR" and shall populate the "onboardingFailureDetails" attribute in "VnfPkgInfo".

If there are on-boarded NSDs that refer to the VNFD included in this VNF Package, their respective NsdInfo records are updated to refer to the on-boarded VNF Package.

**Table 9.4.5.3.3-1: URI query parameters supported by the PUT method on this resource**

Name	Cardinality	Description
none supported		

**Table 9.4.5.3.3-2: Details of the PUT request/response on this resource**

	Data type	Cardinality	Description	
Request body	n/a	1	The payload body contains a ZIP file that represents the VNF package.  The "Content-Type" HTTP header shall be set according to the type of the file, i.e. to "application/zip" for a VNF Package as defined in ETSI GS NFV-SOL 004 [5].	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	Shall be returned when the VNF package has been accepted for uploading, but the processing has not been completed. It is expected to take some time for processing.  The response body shall be empty. See note.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.  Typically, this is due to the fact that the onboarding state of the VNF package resource is not "CREATED" or "ERROR".  The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.
NOTE: The API consumer can track the uploading progress by receiving the "VnfPackageOnBoardingNotification" from the NFVO or by reading the status of the individual VNF package resource using the GET method.				

#### 9.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].



## 9.4.5a Resource: VNF package artifacts

### 9.4.5a.1 Description

This resource represents the artifacts contained in a VNF package. The API consumer can use this resource to bulk-fetch the artifacts.

Optional filters allow to restrict the set of artifacts included in the resource representation. In the present version of the present document, image artifacts are excluded from the representation of this resource.

### 9.4.5a.2 Resource definition

The possible resource URI is:

**{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf\_packages/{vnfPkgId}/artifacts**

This resource shall support the resource URI variables defined in table 9.4.5a.2-1.

**Table 9.4.5a.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.

### 9.4.5a.3 Resource methods

#### 9.4.5a.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.5a.3.2 GET

The GET method shall return an archive that contains a set of artifacts according to the provisions for inclusion/exclusion defined below, embedded in a directory structure being the same as in the VNF package.

The criteria for exclusion/inclusion of an artifact in the archive are defined as follows:

- Artifacts that are software images shall be excluded from the archive.
- Artifacts that are not software images and that are external to the VNF package shall be excluded from the archive unless the URI query parameter "include\_external\_artifacts" has been provided. External artifacts shall be included in the archive using the content of the "artifactPath" attribute as the path.
- All additional artifacts included in the VNF package that are MANO artifacts shall be included in the archive, unless the URI query parameter "exclude\_all\_mano\_artifacts" has been provided, in which case such artifacts shall be excluded.
- All additional artifacts included in the VNF package that are non-MANO artifacts shall be included in the archive, unless:
  - the URI query parameter "exclude\_all\_non\_mano\_artifacts" has been provided, in which case such artifacts shall be excluded;
  - the URI query parameter "select\_non\_mano\_artifact\_sets" has been provided and is supported by the NFVO, in which case only those non-MANO artifacts shall be included whose non-MANO artifact set identifier matches one of the values of the query parameter.

Package metadata such as manifest file or VNFD shall not be included in the archive.

This method shall follow the provisions specified in the tables 9.4.5a.3.2-1 and 9.4.5a.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 9.4.5a.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
exclude_all_mano_artifacts	0..1	Flag (i.e. parameter without value) that instructs the NFVO to exclude the set of additional MANO artifacts (i.e. those that are not images) from the response payload body.  The NFVO shall support this parameter. The OSS/BSS may supply this parameter.
exclude_all_non_mano_artifacts	0..1	Flag (i.e. parameter without value) that instructs the NFVO to exclude the set of non-MANO artifacts from the response payload body.  The NFVO shall support this parameter. The OSS/BSS may supply this parameter.
include_external_artifacts	0..1	Flag (i.e. parameter without value) that instructs the NFVO to include external artifacts in the response payload body. It shall not be treated as an error if this flag is provided but there is no external artifact to include in the result. If this parameter is missing, no external artifacts shall be included.  The NFVO shall support this parameter. The OSS/BSS may supply this parameter.
select_non_mano_artifact_sets	0..1	Comma-separated list of non-MANO artifact set identifiers for which the artifacts are to be included in the response body.  The NFVO should support this parameter. If the NFVO does not support this parameter, it shall ignore it, i.e. provide a response as if no parameter was provided. The OSS/BSS may supply this parameter.
include_signatures	0..1	If this parameter is provided, the NFVO shall include in the ZIP archive the individual signatures and, if provided, related certificates for the included artifacts, in the format in which they are provided in the VNF package.  If this parameter is not given, the NFVO shall only provide copies of the artifact files.  This URI query parameter is a flag, i.e. it shall have no value. The NFVO shall support this parameter.

**Table 9.4.5a.3.2-2: Details of the GET request/response on this resource**

	Data type	Cardinality	Description
Request body	n/a		The "Accept" HTTP header shall be set to "application/zip".
			The request may contain a "Range" HTTP header to obtain single range of bytes from the archive containing the artifacts. This can be used to continue an aborted transmission.  If the "Range" header is present in the request and the NFVO does not support responding to range requests with a 206 response, it shall return a 200 OK response instead as defined below.

	Data type	Cardinality	Response Codes	Description
<b>Response body</b>	n/a	1	200 OK	<p>Shall be returned when the whole content of the archive containing the artifact files has been read successfully.</p> <p>The payload body shall be a ZIP archive containing the requested set of artifacts selected according to the provisions specified above in this clause, and, if the flag "include_signatures" was provided in the related request, the applicable signature files and, if available, the separate certificate files from the VNF package.</p> <p>The "Content-Type" HTTP header shall be set to "application/zip".</p>
	n/a	1	206 Partial Content	<p>If the NFVO supports range requests, this response shall be returned when a single consecutive byte range from the content of the archive that would have been returned in a "200 OK" response has been read successfully according to the request.</p> <p>The response body shall contain the requested part of the archive.</p> <p>The "Content-Type" HTTP header shall be set to "application/zip".</p> <p>The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [10].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid URI query parameters, including the following situations:</p> <ul style="list-style-type: none"> <li>- "exclude_all_non_mano_artifacts" and "select_non_mano_artifact_sets" are both present in the request;</li> <li>- "exclude_all_non_mano_artifacts" and "exclude_all_mano_artifacts" are both present in the request;</li> <li>- one or more of the values provided in "select_non_mano_artifact_sets" are not defined in the manifest of the VNF package.</li> </ul> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "onboardingState" of the VNF package has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	0..1	416 Range Not Satisfiable	<p>Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the archive file (e.g. "access after end of file").</p> <p>The response body may contain a ProblemDetails structure.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>

### 9.4.5a.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 9.4.5a.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 9.4.5a.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 9.4.6 Resource: Upload VNF package from URI task

### 9.4.6.1 Description

This task resource represents the "Upload VNF package from URI" operation. The API consumer can use this resource to request the uploading of a VNF package by providing address information to the NFVO for retrieving the content of the VNF package.

### 9.4.6.2 Resource definition

The resource URI is:

**{apiRoot}/vnfpgm/{apiMajorVersion}/vnf\_packages/{vnfPkgId}/package\_content/upload\_from\_uri**

This resource shall support the resource URI variables defined in table 9.4.6.2-1.

**Table 9.4.6.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new Individual VNF package resource.

### 9.4.6.3 Resource methods

#### 9.4.6.3.1 POST

The POST method provides the information for the NFVO to get the content of a VNF package.

This method shall follow the provisions specified in the tables 9.4.6.3.1-1 and 9.4.6.3.1-2 for URI query parameters, request and response data structures, and response codes.

Upon start of obtaining the package, the NFVO shall set the "onboardingState" attribute in the "VnfPkgInfo" structure to "UPLOADING". Upon successfully obtaining the package, if the package references external artifacts, the NFVO shall obtain the external artifacts. Subsequently, upon success, the NFVO shall set that attribute to "PROCESSING" and shall process the package, which shall include checking package consistency. Upon successful processing, the NFVO shall set the "onboardingState" attribute to "ONBOARDED", the "operationalState" attribute to "ENABLED", and the "usageState" attribute to "NOT\_IN\_USE". In addition, the NFVO shall set the value of the attributes in the "VnfPkgInfo" that are copied from the VNFD (refer to clause 9.5.2.5). If an error occurs during obtaining the package, downloading the external artifacts or processing the package, the NFVO shall set the "onboardingState" attribute to "ERROR" and shall populate the "onboardingFailureDetails" attribute in "VnfPkgInfo".

Table 9.4.6.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.6.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	UploadVnfPkgFromUri Request		1	The payload body contains the address information based on which the NFVO can obtain the content of the VNF package.
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the information about the VNF package has been received successfully, but the on-boarding has not been completed. It is expected to take some time for processing.</p> <p>The response body shall be empty. See note.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the on-boarding state of the VNF package resource is not "CREATED" or "ERROR".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.
NOTE: The API consumer can track the uploading progress by receiving the "VnfPackageOnBoardingNotification" from the NFVO or by reading the status of the individual on-boarded VNF package resource using the GET method.				

#### 9.4.6.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.6.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.6.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 9.4.7 Resource: Individual VNF package artifact

### 9.4.7.1 Description

This resource represents an individual artifact contained in a VNF package. The API consumer can use this resource to fetch the content of the artifact.

### 9.4.7.2 Resource definition

The resource URI is:

**{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf\_packages/{vnfPkgId}/artifacts/{artifactPath}**

This resource shall support the resource URI variables defined in table 9.4.7.2-1.

**Table 9.4.7.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note 1.
artifactPath	<p>For an artifact contained as a file in the VNF package, this variable shall contain a sequence of one or more path segments representing the path of the artifact within the VNF package, relative to the root of the package. See note 3.</p> <p>EXAMPLE:      foo/bar/m%40ster.sh</p> <p>For an external artifact represented as a URI in the VNF package manifest, this variable shall contain a sequence of one or more path segments as synthesized by the NFVO (see clause 9.5.3.3), representing this artifact.</p> <p>See note 2 and note 3.</p>
<p>NOTE 1: This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.</p> <p>NOTE 2: This identifier can be retrieved from the "artifactPath" attribute of the applicable "additionalArtifacts" entry in the body of the response to a GET request querying the "Individual VNF package" or the "VNF packages" resource.</p> <p>NOTE 3: Since multiple path segments are allowed to be contained in this variable, the "/" character that separates these segments is not percent-encoded. Each individual segment is percent-encoded if necessary as defined in clause 4.1 of ETSI GS NFV-SOL 013 [16].</p>	

### 9.4.7.3 Resource methods

#### 9.4.7.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.7.3.2 GET

The GET method fetches the content of an artifact within a VNF package.

This method shall follow the provisions specified in the tables 9.4.7.3.2-1 and 9.4.7.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.7.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
include_signatures	0..1	<p>If this parameter is provided, the NFVO shall return the artifact and related security information (such as signature and optional certificate) in a ZIP archive.</p> <p>If this parameter is not given, the NFVO shall provide only a copy of the artifact file.</p> <p>This URI query parameter is a flag, i.e. it shall have no value.</p> <p>The NFVO shall support this parameter.</p>

Table 9.4.7.3.2-2: Details of the GET request/response on this resource

	Data type	Cardinality	Description	
	Request body	n/a		<p>The request may contain a "Range" HTTP header to obtain single range of bytes from an artifact file. This can be used to continue an aborted transmission.</p> <p>If the Range header is present in the request and the NFVO does not support responding to range requests with a 206 response, it shall return a 200 OK response instead as defined below.</p>
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the whole content of the artifact file has been read successfully.</p> <p>If the "include_signatures" request URI parameter was not provided in the related request, the payload body shall contain a copy of the artifact file from the VNF package, as defined by ETSI GS NFV-SOL 004 [5], and the "Content-Type" HTTP header shall be set according to the content type of the artifact file. If the artifact is encrypted, the header shall be set to the value "application/cms" (IETF RFC 7193 [17]). If the content type cannot be determined, the header shall be set to the value "application/octet-stream".</p> <p>If the "include_signatures" request URI parameter was provided in the related request, the "Content-Type" HTTP header shall be set to "application/zip" and the payload body shall contain a ZIP archive which includes:</p> <ul style="list-style-type: none"> <li>- a copy of the artifact file from the VNF package, as defined by ETSI GS NFV-SOL 004 [5];</li> <li>- the related security information (individual signature file and optional related individual certificate file).</li> </ul>
	n/a	1	206 Partial Content	<p>If the NFVO supports range requests and the "include_signatures" request URI parameter was not present in the related request, this response shall be returned when a single consecutive byte range from the content of the artifact file has been read successfully according to the request.</p> <p>The response body shall contain the requested part of the artifact file from the VNF package, as defined by ETSI GS NFV-SOL 004 [5].</p> <p>The "Content-Type" HTTP header shall be set according to the content type of the artifact file. If the content type cannot be determined, the header shall be set to the value "application/octet-stream". The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [10].</p>

	ProblemDetails	0..1	406 Not Acceptable	If the related request contained an "Accept" header not compatible with the Content type "application/zip" but the "include_signatures" flag was provided, the NFVO shall respond with this response code.  The "ProblemDetails" structure may be included with the "detail" attribute providing more information about the error.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.  Typically, this is due to the fact that "onboardingState" of the VNF package has a value different from "ONBOARDED".  The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	416 Range Not Satisfiable	Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the artifact file (e.g. "access after end of file").  The response body may contain a ProblemDetails structure.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 9.4.7.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 9.4.7.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 9.4.7.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 9.4.8 Resource: Subscriptions

### 9.4.8.1 Description

This resource represents subscriptions. The API consumer can use this resource to subscribe to notifications related to the VNF package management, and to query its subscriptions.

### 9.4.8.2 Resource definition

The resource URI is:

**{apiRoot}/vnfpgm/{apiMajorVersion}/subscriptions**

This resource shall support the resource URI variables defined in table 9.4.8.2-1.



**Table 9.4.8.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.

### 9.4.8.3 Resource methods

#### 9.4.8.3.1 POST

The POST method creates a new subscription.

This method shall follow the provisions specified in the tables 9.4.8.3.1-1 and 9.4.8.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual subscription" resource as defined in clause 9.4.9 shall have been created. This method shall not trigger any notification.

Creation of two "Individual subscription" resources with the same callback URI and the same filter can result in performance degradation and will provide duplicates of notifications to the OSS/BSS, and might make sense only in very rare use cases. Consequently, the NFVO may either allow creating a new "Individual subscription" resource if another "Individual subscription" resource with the same filter and callback URI already exists (in which case it shall return the "201 Created" response code), or may decide to not create a duplicate "Individual subscription" resource (in which case it shall return a "303 See Other" response code referencing the existing "Individual subscription" resource with the same filter and callback URI).

**Table 9.4.8.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 9.4.8.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	PkgmSubscriptionRequest	1	Details of the subscription to be created.	
Response body	Data type	Cardinality	Response Codes	Description
	PkgmSubscription	1	201 Created	<p>Shall be returned when the subscription has been created successfully.</p> <p>The response body shall contain a representation of the created "Individual subscription" resource.</p> <p>The HTTP response shall include a "Location" HTTP header that points to the created resource.</p>
	n/a		303 See Other	<p>Shall be returned when a subscription with the same callback URI and the same filter already exists and the policy of the NFVO is to not create redundant subscriptions.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource.</p> <p>The response body shall be empty.</p>

	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 9.4.10.3.2 and the test has failed. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 9.4.8.3.2 GET

The GET method queries the list of active subscriptions of the functional block that invokes the method. It can be used e.g. for resynchronization after error situations.

This method shall follow the provisions specified in the tables 9.4.8.3.2-1 and 9.4.8.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 9.4.8.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>All attribute names that appear in the PkgmSubscription and in data types referenced from it shall be supported by the NFVO in the filter expression.</p>
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 9.4.8.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	PkgmSubscription	0..N	200 OK	<p>Shall be returned when the list of subscriptions has been queried successfully.</p> <p>The response body shall contain in an array the representations of all active subscriptions of the functional block that invokes the method, i.e. zero or more representations of VNF package management subscriptions, as defined in clause 9.5.2.7.</p> <p>If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [16].</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>

#### 9.4.8.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.8.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.8.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 9.4.9 Resource: Individual subscription

### 9.4.9.1 Description

This resource represents an individual subscription. The API consumer can use this resource to read and to terminate a subscription to notifications related to the VNF package management.

### 9.4.9.2 Resource definition

The resource URI is:

**{apiRoot}/vnfpkgm/{apiMajorVersion}/subscriptions/{subscriptionId}**

This resource shall support the resource URI variables defined in table 9.4.9.2-1.

**Table 9.4.9.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
subscriptionId	Identifier of this subscription. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual subscription" resource. It can also be retrieved from the "id" attribute in the payload body of that response.

### 9.4.9.3 Resource methods

#### 9.4.9.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 9.4.9.3.2 GET

The GET method reads an individual subscription.

This method shall follow the provisions specified in the tables 9.4.9.3.2-1 and 9.4.9.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 9.4.9.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 9.4.9.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	PkgmSubscription	1	200 OK	Shall be returned when information about an individual subscription has been read successfully.  The response body shall contain a representation of the "Individual subscription" resource.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 9.4.9.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 9.4.9.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 9.4.9.3.5 DELETE

The DELETE method terminates an individual subscription.

This method shall follow the provisions specified in the tables 9.4.9.3.5-1 and 9.4.9.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual subscription" resource shall not exist any longer. This means that no notifications for that subscription shall be sent to the formerly-subscribed API consumer.

**NOTE:** Due to race conditions, some notifications might still be received by the formerly-subscribed API consumer for a certain time period after the deletion.

**Table 9.4.9.3.5-1: URI query parameters supported by the DELETE method on this resource**

Name	Cardinality	Description
none supported		

**Table 9.4.9.3.5-2: Details of the DELETE request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the "Individual subscription" resource has been deleted successfully.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 9.4.10 Resource: Notification endpoint

### 9.4.10.1 Description

This resource represents a notification endpoint.

The API producer can use this resource to send notifications related to VNF package management events to a subscribed API consumer, which has provided the URI of this resource during the subscription process.

### 9.4.10.2 Resource definition

The resource URI is provided by the API consumer when creating the subscription.

This resource shall support the resource URI variables defined in table 9.4.10.2-1.

**Table 9.4.10.2-1: Resource URI variables for this resource**

Name	Definition
none supported	

### 9.4.10.3 Resource methods

#### 9.4.10.3.1 POST

The POST method delivers a notification from the API producer to an API consumer. The API consumer shall have previously created an "individual subscription resource" with a matching filter.

This method shall follow the provisions specified in the tables 9.4.10.3.1-1 and 9.4.10.3.1-2 for URI query parameters, request and response data structures, and response codes.

**Table 9.4.10.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

Each notification request body shall include exactly one of the alternatives defined in table 9.4.10.3.1-2.

**Table 9.4.10.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	VnfPackageOnboardingNotification	1	A notification about on-boarding of a VNF package.	
VnfPackageChangeNotification	1	A notification about changes of status in a VNF package.		
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the notification has been delivered successfully.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 9.4.10.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during subscription.

This method shall follow the provisions specified in the tables 9.4.10.3.2-1 and 9.4.10.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 9.4.10.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 9.4.10.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
n/a				
Response body	Data type	Cardinality	Response Codes	Description
n/a			204 No Content	Shall be returned to indicate that the notification endpoint has been tested successfully. The response body shall be empty.
ProblemDetails		See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 9.4.10.3.3 PUT

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 9.4.10.3.4 PATCH

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 9.4.10.3.5 DELETE

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 9.5 Data model

### 9.5.1 Introduction

This clause defines the request and response data structures of the VNF package management interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error, and may choose to ignore them.

### 9.5.2 Resource and notification data types

#### 9.5.2.1 Introduction

This clause defines data structures to be used in resource representations and notifications.

#### 9.5.2.2 Type: CreateVnfPkgInfoRequest

This type represents the request parameters for creating a new individual VNF package resource. It shall comply with the provisions defined in table 9.5.2.2-1.

**Table 9.5.2.2-1: Definition of the CreateVnfPkgInfoRequest data type**

Attribute name	Data type	Cardinality	Description
userDefinedData	KeyValuePairs	0..1	User defined data for the VNF package.

### 9.5.2.3 Type: VnfPkgInfoModifications

This type represents modifications to the information of a VNF package. It shall comply with the provisions defined in table 9.5.2.3-1.

**Table 9.5.2.3-1: Definition of the VnfPkgInfoModifications data type**

Attribute name	Data type	Cardinality	Description
operationalState	PackageOperationalStateType	0..1	New value of the operational state of the on-boarded instance of the VNF package. See note.
userDefinedData	KeyValuePairs	0..1	User defined data to be updated. For existing keys, the value is replaced. See note.
NOTE: At least one of the two parameters shall be present. If the VNF package is not on-boarded, the operation is used only to update existing or add additional user defined data using the userDefinedData attribute.			

### 9.5.2.4 Type: UploadVnfPackageFromUriRequest

This type represents the request parameters for uploading the content of a VNF package. The NFVO can obtain the VNF package content through the information provided in the request parameters. It shall comply with the provisions defined in table 9.5.2.4-1.

**Table 9.5.2.4-1: Definition of the UploadVnfPackageFromUriRequest data type**

Attribute name	Data type	Cardinality	Description
addressInformation	Uri	1	Address information of the VNF package content. The NFVO can use this address to obtain the VNF package.
authType	Enum (inlined)	0..1	Defines the type of authentication/authorization for downloading the VNF package.  Permitted values: <ul style="list-style-type: none"> <li>- BASIC: Only the "username" and "password" attributes shall be present.</li> <li>- OAUTH2_CLIENT_CREDENTIALS: Only the "paramsOauth2ClientCredentials" attribute shall be present.</li> </ul> This attribute shall not be present if the credentials were provisioned out of band.
userName	String	0..1	User name to be used for authentication.
password	String	0..1	Password to be used for authentication.
paramsOauth2ClientCredentials	Structure (inlined)	0..1	Parameters for authentication/authorization using OAuth 2.0.
>clientId	String	1	Client identifier to be used in the access token request of the OAuth 2.0 client credentials grant type.
>clientPassword	String	1	Client password to be used in the access token request of the OAuth 2.0 client credentials grant type.
>tokenEndpoint	Uri	1	The token endpoint from which the access token can be obtained.

### 9.5.2.5 Type: VnfPkgInfo

This type represents the information of a VNF package. It shall comply with the provisions defined in table 9.5.2.5-1.



Table 9.5.2.5-1: Definition of the VnfPkgInfo data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the VNF package. This identifier is allocated by the NFVO.
vnfdId	Identifier	0..1	This identifier, which is managed by the VNF provider, identifies the VNF package and the VNFD in a globally unique way. It is copied from the VNFD of the on-boarded VNF package. It shall be present after the VNF package content has been on-boarded and absent otherwise.
vnfdExtInvariantId	Identifier	0..1	Identifies a VNFD in a version independent manner. This attribute is invariant across versions of the VNFD that fulfil certain conditions related to the external connectivity and management of the VNF. It shall be present after the VNF package content has been on-boarded if it is included in the VNFD and shall be absent otherwise. If present it is copied from the VNFD of the on-boarded VNF package.
vnfProvider	String	0..1	Provider of the VNF package and the VNFD. This information is copied from the VNFD. It shall be present after the VNF package content has been on-boarded and absent otherwise.
vnfProductName	String	0..1	Name to identify the VNF product. Invariant for the VNF product lifetime. This information is copied from the VNFD. It shall be present after the VNF package content has been on-boarded and absent otherwise.
vnfSoftwareVersion	Version	0..1	Software version of the VNF. This is changed when there is any change to the software included in the VNF package. This information is copied from the VNFD. It shall be present after the VNF package content has been on-boarded and absent otherwise.
vnfdVersion	Version	0..1	The version of the VNFD. This information is copied from the VNFD. It shall be present after the VNF package content has been on-boarded and absent otherwise.
compatibleSpecification Versions	Version	0..N	Indicates which versions of the ETSI GS NFV-SOL 004 [5] specification the package complies to, as defined in the manifest of the package. Each entry shall be formatted as defined in clause 4.3.2 of ETSI GS NFV-SOL 004 [5].
checksum	Checksum	0..1	Checksum of the on-boarded VNF package. Permitted hash algorithms are defined in ETSI GS NFV-SOL 004 [5]. It shall be present after the VNF package content has been on-boarded and absent otherwise.
packageSecurityOption	Enum (inlined)	0..1	Signals the security option used by the package as defined in clause 5.1 of ETSI GS NFV-SOL 004 [5]. It shall be present after the VNF package content has been on-boarded and absent otherwise.  Valid values: - OPTION_1 - OPTION_2
signingCertificate	String	0..1	The singleton signing certificate if it is included as a file in the VNF package.

Attribute name	Data type	Cardinality	Description
softwareImages	VnfPackageSoftwareImageInfo	0..N	Information about VNF package artifacts that are software images.  Every local and external software image referenced from the VNFD shall be included. No other artifacts shall be included. This attribute shall not be present before the VNF package content is on-boarded. Otherwise, this attribute shall be present unless it has been requested to be excluded per attribute selector.
additionalArtifacts	VnfPackageArtifactInfo	0..N	Information about VNF package artifacts contained in the VNF package that are not software images.  Every local and external artifact declared in the manifest shall be included, except the software images and the files that make up the parts of the VNFD (see clause 9.4.4.3.2).  Signature files and certificate files are not considered as artifacts, however, the content of the "Licenses" and "Testing" directories in the VNF package is.  This attribute shall not be present before the VNF package content is on-boarded. Otherwise, this attribute shall be present if the VNF package contains additional artifacts.
onboardingState	PackageOnboardingStateType	1	On-boarding state of the VNF package. See note 3.
operationalState	PackageOperationalStateType	1	Operational state of the VNF package.  See note 1 and note 3.
usageState	PackageUsageStateType	1	Usage state of the VNF package.  See note 2 and note 3.
vnfmInfo	String	1..N	Specifies VNFMs compatible with the VNF. This information is copied from the VNFD. See note 4.
userDefinedData	KeyValuePairs	0..1	User defined data for the VNF package.
onboardingFailureDetails	ProblemDetails	0..1	Failure details of current onboarding procedure. See clause 6.3 of ETSI GS NFV-SOL 013 [16] for the details of "ProblemDetails" structure.  If "onboardingState" is "ERROR", this attribute shall be present and contain error information (such as failed onboarding or processing operation, affected artifact, etc.), unless it has been requested to be excluded via an attribute selector.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>vnfd	Link	1	Link to the "VNFD in an individual VNF package" resource.
>packageContent	Link	1	Link to the "VNF package content" resource.
NOTE 1: If the value of the onboardingState attribute is not equal to "ONBOARDED", the value of the operationalState attribute shall be equal to "DISABLED".			
NOTE 2: If the value of the onboardingState attribute is not equal to "ONBOARDED", the value of the usageState attribute shall be equal to "NOT_IN_USE".			
NOTE 3: State changes of a VNF package are illustrated in clause B.2.			
NOTE 4: ETSI GS NFV-SOL 001 [i.1] specifies the structure and format of the VNFD based on TOSCA specifications.			

### 9.5.2.6 Type: PkgmSubscriptionRequest

This type represents a subscription request related to VNF package management notifications about VNF package on-boarding or changes. It shall comply with the provisions defined in table 9.5.2.6-1.

**Table 9.5.2.6-1: Definition of the PkgmSubscriptionRequest data type**

Attribute name	Data type	Cardinality	Description
filter	PkgmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of authorization when sending notifications corresponding to this subscription, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [16].  This attribute shall only be present if the subscriber requires authorization of notifications.

### 9.5.2.7 Type: PkgmSubscription

This type represents a subscription related to notifications about VNF package management. It shall comply with the provisions defined in table 9.5.2.7-1.

**Table 9.5.2.7-1: Definition of the PkgmSubscription data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this "Individual subscription" resource.
filter	PkgmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.

### 9.5.2.8 Type: VnfPackageOnboardingNotification

This type represents a VNF package management notification, which informs the receiver that the onboarding process of a VNF package is complete and the package is ready for use. A change of the on-boarding state before the VNF package is on-boarded is not reported. It shall comply with the provisions defined in table 9.5.2.8-1. The support of this notification is mandatory. The notification shall be triggered by the NFVO when the value of the "onboardingState" attribute of a new VNF package has changed to "ONBOARDED".

**Table 9.5.2.8-1: Definition of the VnfPackageOnboardingNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "VnfPackageOnboardingNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.

Attribute name	Data type	Cardinality	Description
vnfPkgId	Identifier	1	Identifier of the VNF package. This identifier is allocated by the NFVO.  Its value is the same as the value of the "id" attribute of the related "Individual VNF package" resource.
vnfdId	Identifier	1	This identifier, which is managed by the VNF provider, identifies the VNF package and the VNFD in a globally unique way. It is copied from the VNFD of the on-boarded VNF package.
vnfmInfo	String	1..N	Specifies VNFMs compatible with the VNF. This information is copied from the VNFD. See table 9.5.2.5-1.
_links	PkgmLinks	1	Links to resources related to this notification.

### 9.5.2.9 Type: VnfPackageChangeNotification

This type represents a VNF package management notification, which informs the receiver of a change of the status in an on-boarded VNF package. Only changes in the "operationalState" attribute of an on-boarded VNF package and the deletion of the VNF package will be reported. Changes in the "usageState" and "onboardingState" attributes are not reported. The notification shall comply with the provisions defined in table 9.5.2.9-1. The support of this notification is mandatory. The notification shall be triggered by the NFVO when there is a change in the status of an onboarded VNF package, as follows:

- The "operationalState" attribute of a VNF package has changed, and the "onboardingState" attribute of the package has the value "ONBOARDED" (i.e. the package has been onboarded previously).
- The on-boarded VNF package has been deleted, and the "onboardingState" attribute of the deleted package had the value "ONBOARDED".

**Table 9.5.2.9-1: Definition of the VnfPackageChangeNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "VnfPackageChangeNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
vnfPkgId	Identifier	1	Identifier of the VNF package. This identifier is allocated by the NFVO.  Its value is the same as the value of the "id" attribute of the related "Individual VNF package" resource.
vnfdId	Identifier	1	Identifier of the VNFD contained in the VNF package, which also identifies the VNF package. This identifier is allocated by the VNF provider and copied from the VNFD.
changeType	PackageChangeType	1	The type of change of the VNF package.
operationalState	PackageOperationalStateType	0..1	New operational state of the VNF package. Only present when changeType is OP_STATE_CHANGE.
_links	PkgmLinks	1	Links to resources related to this notification.

### 9.5.2.10 Type: ExternalArtifactsAccessConfig

This type represents the access configuration information for downloading external VNF package artifacts. The NFVO can obtain the external VNF package artifact file through the information provided in this structure, together with information provided in the manifest/VNFD. The data structure shall comply with the provisions defined in table 9.5.2.10-1. If the data structure is part of a response body, security-sensitive attributes shall be excluded as specified in table 9.5.2.10-1.

**Table 9.5.2.10-1: Definition of the ExternalArtifactsAccessConfig data type**

Attribute name	Data type	Cardinality	Description
artifact	Structure (inlined)	0..N	Access configuration information for an external artifact.
>artifactUri	Uri	1	The artifactUri attribute as defined in the VNF package manifest that identifies the external artifact for which the access configuration information is provided, and that provides the default download location for this artifact.
>overrideUri	Uri	0..1	The Uri from which the external artifact can be downloaded. If this attribute is present, the NFVO shall download from this address. If this attribute is not present, the NFVO shall download from the address provided in "artifactUri".
>authType	Enum (inlined)	0..1	Defines the type of authentication/authorization for downloading the VNF package.  Permitted values: <ul style="list-style-type: none"> <li>- BASIC: Only the "username" and "password" attributes shall be present.</li> <li>- OAUTH2_CLIENT_CREDENTIALS: Only the "paramsOauth2ClientCredentials" attribute shall be present.</li> </ul> This attribute shall not be present if no credentials are provided for the artifact.
>userName	String	0..1	User name to be used for authentication.
>password	String	0..1	Password to be used for authentication. Shall not be present in response bodies.
>paramsOauth2ClientCredentials	Structure (inlined)	0..1	Parameters for authentication/authorization using OAUTH2_CLIENT_CREDENTIALS.
>>clientId	String	1	Client identifier to be used in the access token request of the OAuth 2.0 client credentials grant type.
>>clientPassword	String	1	Client password to be used in the access token request of the OAuth 2.0 client credentials grant type. Shall not be present in response bodies.
>>tokenEndpoint	Uri	1	The token endpoint from which the access token can be obtained.

## 9.5.3 Referenced structured data types

### 9.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but are not resource representations.

### 9.5.3.2 Type: VnfPackageSoftwareImageInfo

This type represents an artifact contained in or external to a VNF package which represents a software image. It shall comply with provisions defined in table 9.5.3.2-1.

**Table 9.5.3.2-1: Definition of the VnfPackageSoftwareImageInfo data type**

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnfd	1	Identifier of the software image.
name	String	1	Name of the software image.
provider	String	1	Provider of the software image.
version	Version	1	Version of the software image.
checksum	Checksum	1	Checksum of the software image file. Permitted hash algorithms are defined in ETSI GS NFV-SOL 004 [5].
isEncrypted	Boolean	1	Reflects whether the image is encrypted (true) or not (false).
containerFormat	Enum (inlined)	1	<p>Container format indicates whether the software image is in a file format that also contains metadata about the actual software.</p> <p>Permitted values:</p> <ul style="list-style-type: none"> <li>- AKI: a kernel image format</li> <li>- AMI: a machine image format</li> <li>- ARI: a ramdisk image format</li> <li>- BARE: the image does not have a container or metadata envelope</li> <li>- DOCKER: docker container format</li> <li>- OVA: OVF package in a tarfile</li> <li>- OVF: OVF container format</li> </ul> <p>See note 1.</p>
diskFormat	Enum (inlined)	1	<p>Disk format of a software image is the format of the underlying disk image.</p> <p>Permitted values:</p> <ul style="list-style-type: none"> <li>- AKI: a kernel image format</li> <li>- AMI: a machine image format</li> <li>- ARI: a ramdisk image format</li> <li>- ISO: an archive format for the data contents of an optical disc, such as CD-ROM</li> <li>- QCOW2: a common disk image format, which can expand dynamically and supports copy on write</li> <li>- RAW: an unstructured disk image format</li> <li>- VDI: a common disk image format</li> <li>- VHD: a common disk image format</li> <li>- VHDX: enhanced version of VHD format</li> <li>- VMDK: a common disk image format</li> </ul> <p>See note 2.</p>
createdAt	DateTime	1	Time when this software image was created.
minDisk	UnsignedInt	1	The minimal disk for this software image in bytes.
minRam	UnsignedInt	1	The minimal RAM for this software image in bytes.
size	UnsignedInt	1	Size of this software image in bytes.
userMetadata	KeyValuePairs	0..1	User-defined data.

Attribute name	Data type	Cardinality	Description
imagePath	String	0..1	<p>Path which identifies the image artifact and also allows to access a copy of the image artifact.</p> <p>For a software image contained as a file in the VNF package, this attribute shall be present, and the value of this attribute shall start with the name of the first segment in the path in the package, i.e. it shall not be prefixed by path separator characters such as "." and "/".</p> <p>EXAMPLE:       foo/bar/m%40ster.vhd</p> <p>For an external software image represented as a URI in the VNF descriptor, this attribute shall be present if the image artifact has been downloaded by the NFVO and shall be absent otherwise. If present, it shall contain the artifactPath under which the image artifact can be obtained using the "Individual artifact in a VNF package" resource defined in clause 9.4.7. It is the responsibility of the NFVO to synthesize this path in a manner that avoids any collision of the synthesized artifact path with the paths and names of image artifacts included in the package.</p>
imageUri	Uri	0..1	<p>URI of the image artifact as defined in the VNF package manifest. Shall be present if the image artifact is external to the VNF package and shall be absent otherwise.</p> <p>EXAMPLE:       https://example.com/m%40ster.vhd</p>
NOTE 1: The list of permitted values was taken from "Container formats" in [i.2].			
NOTE 2: The list of permitted values was adapted from "Disk formats" in [i.2].			

### 9.5.3.3 Type: VnfPackageArtifactInfo

This type represents an artifact other than a software image which is contained in or external to a VNF package. It shall comply with provisions defined in table 9.5.3.3-1.

**Table 9.5.3.3-1: Definition of the VnfPackageArtifactInfo data type**

Attribute name	Data type	Cardinality	Description
artifactPath	String	0..1	<p>Path which identifies the artifact and also allows to access a copy of the artifact.</p> <p>For an artifact contained as a file in the VNF package, this attribute shall be present, and the value of this attribute shall start with the name of the first segment in the path in the package, i.e. it shall not be prefixed by path separator characters such as "." and "/".</p> <p>EXAMPLE:       foo/bar/m@ster.sh</p> <p>For an external artifact represented as a URI in the VNF descriptor, this attribute shall be present if the artifact has been downloaded by the NFVO and shall be absent otherwise. If present, it shall contain the artifactPath under which the artifact can be obtained using the "Individual artifact in a VNF package" resource defined in clause 9.4.7. It is the responsibility of the NFVO to synthesize this path in a manner that avoids any collision of the synthesized artifact path with the paths and names of artifacts included in the package.</p>

Attribute name	Data type	Cardinality	Description
artifactURI	Uri	0..1	URI of the artifact as defined in the VNF package manifest. Shall be present if the artifact is external to the package and shall be absent otherwise.  EXAMPLE: <a href="https://example.com/m%40ster.sh">https://example.com/m%40ster.sh</a>
checksum	Checksum	1	Checksum of the artifact file. Permitted hash algorithms are defined in ETSI GS NFV-SOL 004 [5].
isEncrypted	Boolean	1	Reflects whether the artifact is encrypted (true) or not (false).
nonManoArtifactSetId	String	0..1	Non-MANO artifact set identifier of the non-MANO artifact set to which the artifact belongs, as defined in clause 4.3.7 of ETSI GS NFV-SOL 004 [5]. Shall be provided if the artifact is a non-MANO artifact, and shall be omitted otherwise.
artifactClassification	Enum (inlined)	0..1	Marks specific types of artifacts as defined in the VNF package. If none of the specific classes listed below applies, the attribute shall not be present.  Valid values: <ul style="list-style-type: none"> <li>- HISTORY: a history artifact as per clause 4.3.3 in ETSI GS NFV-SOL 004 [5]</li> <li>- TESTING: a testing artifact as per clause 4.3.4 in ETSI GS NFV-SOL 004 [5]</li> <li>- LICENSE: a license artifact as per clause 4.3.5 in ETSI GS NFV-SOL 004 [5]</li> </ul>
metadata	KeyValuePairs	0..1	The metadata of the artifact that are available in the VNF package, such as Content type, size, creation date, etc.

#### 9.5.3.4 Type: PkgmNotificationsFilter

This type represents a subscription filter related to notifications related to VNF package management. It shall comply with the provisions defined in table 9.5.3.4-1.

At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).

**Table 9.5.3.4-1: Definition of the PkgmNotificationsFilter data type**

Attribute name	Data type	Cardinality	Description
notificationTypes	Enum (inlined)	0..N	Match particular notification types.  Permitted values: <ul style="list-style-type: none"> <li>- VnfPackageOnboardingNotification</li> <li>- VnfPackageChangeNotification</li> </ul> See note 1.
vnfProductsFromProviders	Structure (inlined)	0..N	If present, match VNF packages that contain VNF products from certain providers. See note 2.
>vnfProvider	String	1	Name of the VNF provider to match.
>vnfProducts	Structure (inlined)	0..N	If present, match VNF packages that contain VNF products with certain product names, from one particular provider.
>>vnfProductName	String	1	Name of the VNF product to match.
>>versions	Structure (inlined)	0..N	If present, match VNF packages that contain VNF products with certain versions and a certain product name, from one particular provider.
>>>vnfSoftwareVersion	Version	1	VNF software version to match.



Attribute name	Data type	Cardinality	Description
>>>vnfdVersions	Version	0..N	If present, match VNF packages that contain VNF products with certain VNFD versions, a certain software version and a certain product name, from one particular provider.
vnfdId	Identifier	0..N	Match VNF packages with a VNFD identifier listed in the attribute. See note 2.
vnfPkgId	Identifier	0..N	Match VNF packages with a package identifier listed in the attribute.  May be present if the "notificationTypes" attribute contains the value "VnfPackageChangeNotification", and shall be absent otherwise. See note 2.
operationalState	PackageOperationalStateType	0..N	Match particular operational states of the VNF package.  May be present if the "notificationTypes" attribute contains the value "VnfPackageChangeNotification", and shall be absent otherwise.
usageState	PackageUsageStateType	0..N	Match particular usage states of the VNF package.  May be present if the "notificationTypes" attribute contains the value "VnfPackageChangeNotification", and shall be absent otherwise.
vnfmInfo	String	0..N	Match strings that specify VNFM compatible with the VNF. See table 9.5.2.5-1.
NOTE 1: The permitted values of the "notificationTypes" attribute are spelled exactly as the names of the notification types to facilitate automated code generation systems.			
NOTE 2: The attributes "vnfProductsFromProviders", "vnfdId", and "vnfPkgId" are alternatives to reference particular VNF packages in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.			

### 9.5.3.5 Type: PkgmLinks

This type represents the links to resources that a VNF package management notification can contain. It shall comply with the provisions defined in table 9.5.3.5-1.

**Table 9.5.3.5-1: Definition of the PkgmLinks data type**

Attribute name	Data type	Cardinality	Description
vnfPackage	NotificationLink	1	Link to the resource representing the VNF package to which the notified change applies, i.e. the "Individual VNF package" resource that represents the VNF package.
subscription	NotificationLink	1	Link to the related subscription.

### 9.5.3.6 Void

## 9.5.4 Referenced simple data types and enumerations

### 9.5.4.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

### 9.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.2.

### 9.5.4.3 Enumeration: PackageOnboardingStateType

The enumeration PackageOnboardingStateType shall comply with the provisions defined in table 9.5.4.3-1.

**Table 9.5.4.3-1: Enumeration PackageOnboardingStateType**

Enumeration value	Description
CREATED	The VNF package resource has been created.
UPLOADING	The associated VNF package content is being uploaded.
PROCESSING	The associated VNF package content is being processed, e.g. validation.
ONBOARDED	The associated VNF package content has been on-boarded successfully.
ERROR	There was an error during upload of the VNF package content or external artifacts, or during VNF package processing.

### 9.5.4.4 Enumeration: PackageOperationalStateType

The enumeration PackageOperationalStateType shall comply with the provisions defined in table 9.5.4.4-1.

**Table 9.5.4.4-1: Enumeration PackageOperationalStateType**

Enumeration value	Description
ENABLED	The VNF package is enabled, i.e. it can be used for the creation of new "Individual VNF instance" resources.
DISABLED	The VNF package is disabled, i.e. it shall not be used for the creation of further "Individual VNF instance" resources (unless and until the VNF package is re-enabled).

### 9.5.4.5 Enumeration: PackageUsageStateType

The enumeration PackageUsageStateType shall comply with the provisions defined in table 9.5.4.5-1.

**Table 9.5.4.5-1: Enumeration PackageUsageStateType**

Enumeration value	Description
IN_USE	"Individual VNF instance" resources created from this VNF package exist.
NOT_IN_USE	No "individual VNF instance" resource created from this VNF package exists.

### 9.5.4.6 Enumeration: PackageChangeType

The enumeration PackageChangeType shall comply with the provisions defined in table 9.5.4.6-1.

**Table 9.5.4.6-1: Enumeration PackageChangeType**

Enumeration value	Description
OP_STATE_CHANGE	The "operationalState" attribute has been changed.
PKG_DELETE	The VNF package has been deleted.

---

## 10 NFVI Capacity Information interface

### 10.1 Description

This interface allows providing of NFVI capacity information (including capacity shortage notifications) to the OSS/BSS.

The operations provided through this interface are as follows:

- Query NFVI capacity
- Subscribe
- Notify

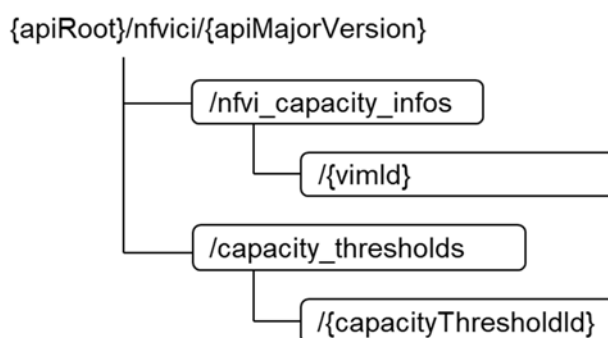
## 10.1a API version

For the NFVI capacity information interface version as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 10, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [16] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v1".

## 10.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [16]. The string "nfvici" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 10.2-1 shows the overall resource URI structure defined for the NFVI capacity information interface.



**Figure 10.2-1: Resource URI structure of the NFVI Capacity Information interface**

Table 10.2-1 lists the individual resources defined, and the applicable HTTP methods. The NFVO shall support responding to requests for all HTTP methods on the resources in table 10.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [16].

**Table 10.2-1: Resources and methods overview of the NFVI Capacity Information interface**

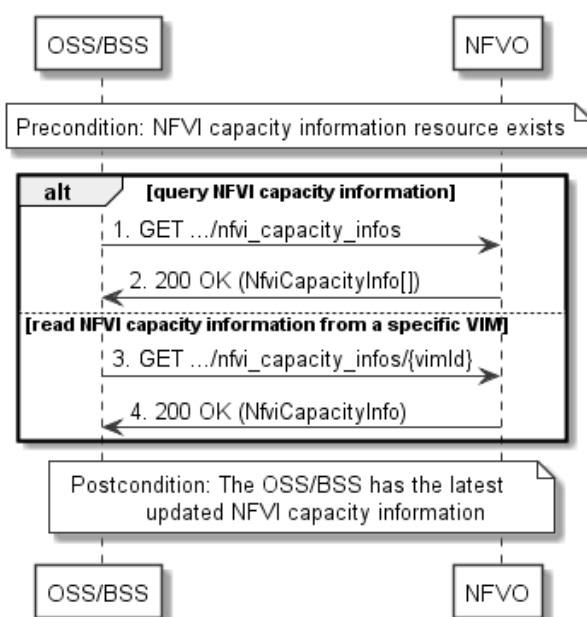
Resource name	Resource URI	HTTP Method	Cat	Meaning
NFVI capacity information	/nfvi_capacity_infos	GET	M	Query NFVI capacity information.
Individual VIM's NFVI capacity information	/nfvi_capacity_infos/{vimId}	GET	M	Read NFVI capacity information from a specific VIM.
Capacity thresholds	/capacity_thresholds	POST	M	Create an NFVI capacity threshold.
		GET	M	Query NFVI capacity thresholds.
Individual capacity threshold	/capacity_thresholds/{capacityThresholdId}	GET	M	Read a single NFVI capacity threshold.
		PATCH	M	Update NFVI capacity threshold callback.
		DELETE	M	Delete a single NFVI capacity threshold.

Resource name	Resource URI	HTTP Method	Cat	Meaning
Notification endpoint	(provided by API consumer)	POST	See note	Notify about NFVI capacity. See note.
		GET	See note	Test the notification endpoint. See note.
NOTE: The NFVO shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the OSS/BSS. If the OSS/BSS supports invoking the POST method on the "Capacity thresholds" resource towards the NFVO, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.				

## 10.3 Sequence diagrams (informative)

### 10.3.1 Flow of querying/reading NFVI capacity information

This clause describes a sequence flow for querying/reading NFVI capacity information.



**Figure 10.3.1-1: Flow of querying/reading NFVI capacity information**

**Precondition:** NFVI capacity information resource exists.

The procedure consists of the following steps as illustrated in figure 10.3.1-1:

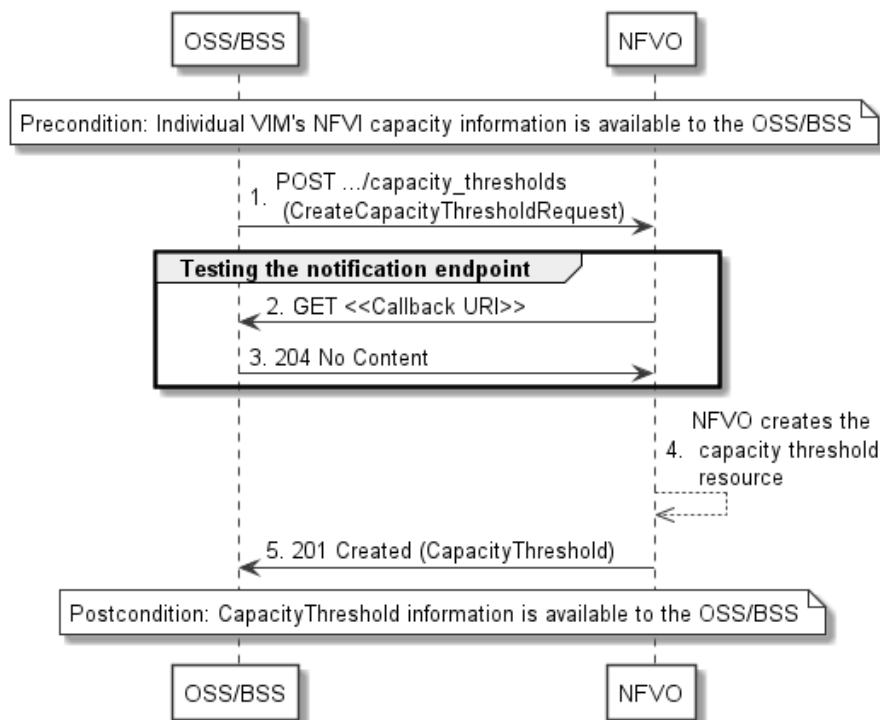
- 1) If the OSS/BSS intends to query all NFVI capacity information, it sends a GET request to the "NFVI capacity Information" resource.
- 2) In case of step 1), the NFVO returns a "200 OK" response to the OSS/BSS, and includes zero or more data structures of type "NfviCapacityInfo" in the payload body.
- 3) If the OSS/BSS intends to read information about a particular VIM's NFVI capacity information, it sends a GET request to the "Individual VIM's NFVI capacity information" resource, addressed by the appropriate VIM identifier of NFVI capacity information in its resource URI.
- 4) In case of step 3), the NFVO returns a "200 OK" response to the OSS/BSS, and includes one data structure of type "NfviCapacityInfo" in the payload body.

**Postcondition:** The OSS/BSS has the latest updated NFVI capacity information.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 10.3.2 Flow of creating a capacity threshold

This clause describes the procedure for creating a capacity threshold.



**Figure 10.3.2-1: Flow of creating a capacity threshold**

**Precondition:** Individual VIM's NFVI capacity information is available to the OSS/BSS.

The creation of a capacity threshold, as illustrated in figure 10.3.2-1, consists of the following steps:

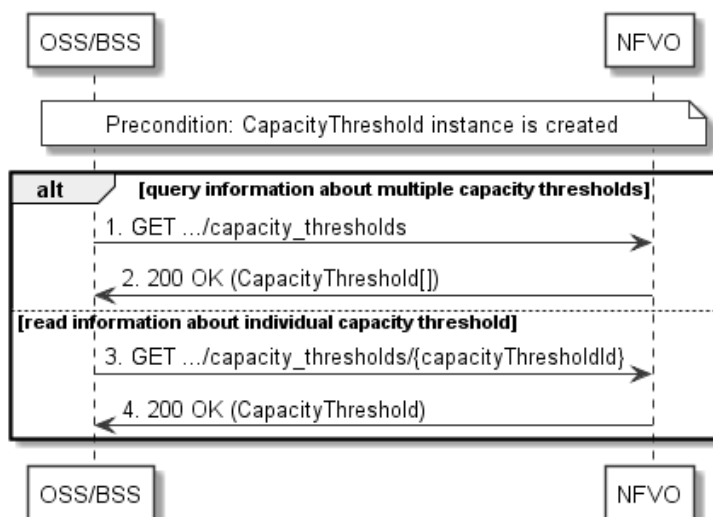
- 1) The OSS/BSS sends a POST request to the "Capacity thresholds" resource, including a data structure of type "CreateCapacityThresholdRequest" in the payload body.
- 2) To test the notification endpoint that was registered by the OSS/BSS during the capacity threshold creation, the NFVO send a GET request to the notification endpoint URI.
- 3) The OSS/BSS returns a "204 No Content" response to indicate success.
- 4) The NFVO creates a capacity threshold resource.
- 5) The NFVO returns a "201 Created" response to the OSS/BSS, and includes in the payload body a representation of the capacity threshold just created.

**Postcondition:** CapacityThreshold information is available to the OSS/BSS.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 10.3.3 Flow of querying/reading capacity thresholds

This clause describes the procedure for querying/reading capacity thresholds.



**Figure 10.3.3-1: Flow of capacity thresholds query/read**

**Precondition:** CapacityThreshold instance is created.

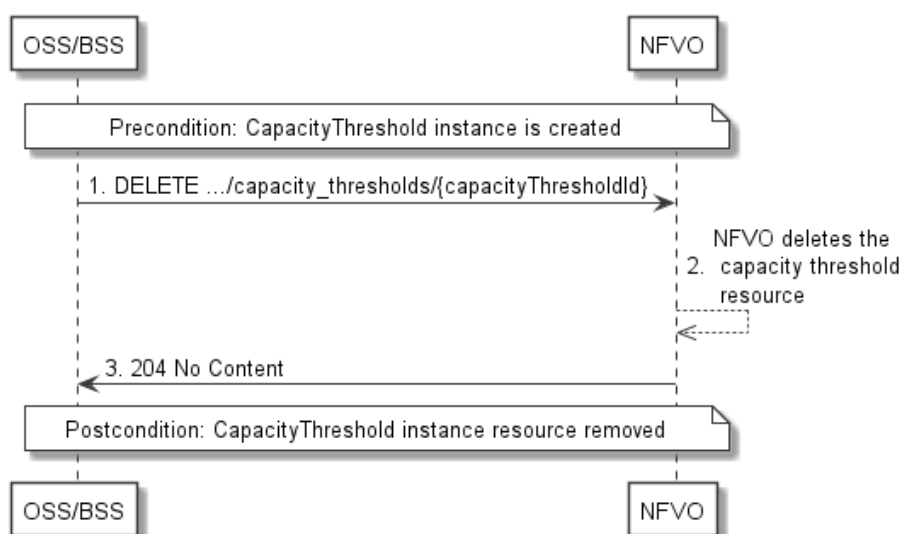
The query/read capacity thresholds, as illustrated in figure 10.3.3-1, consists of the following steps:

- 1) If the OSS/BSS intends to query all capacity thresholds, it sends a GET request to the "Capacity thresholds" resource.
- 2) In case of step 1), the NFVO returns a "200 OK" response to the OSS/BSS, and includes zero or more data structures of type "CapacityThreshold" in the payload body.
- 3) If the OSS/BSS intends to read information about a particular capacity threshold, it sends a GET request to the "Individual capacity threshold" resource address by the appropriate capacity threshold identifier in its resource URI.
- 4) In case of step 3), the NFVO returns a "200 OK" response to the OSS/BSS, and includes a data structure of type "CapacityThreshold" in the payload body.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 10.3.4 Flow of deleting a capacity threshold

This clause describes the procedure for deleting a capacity threshold.



**Figure 10.3.4-1: Flow of capacity threshold deletion**

**Precondition:** CapacityThreshold instance is created.

The deletion of a capacity threshold, as illustrated in figure 10.3.4-1, consists of the following steps:

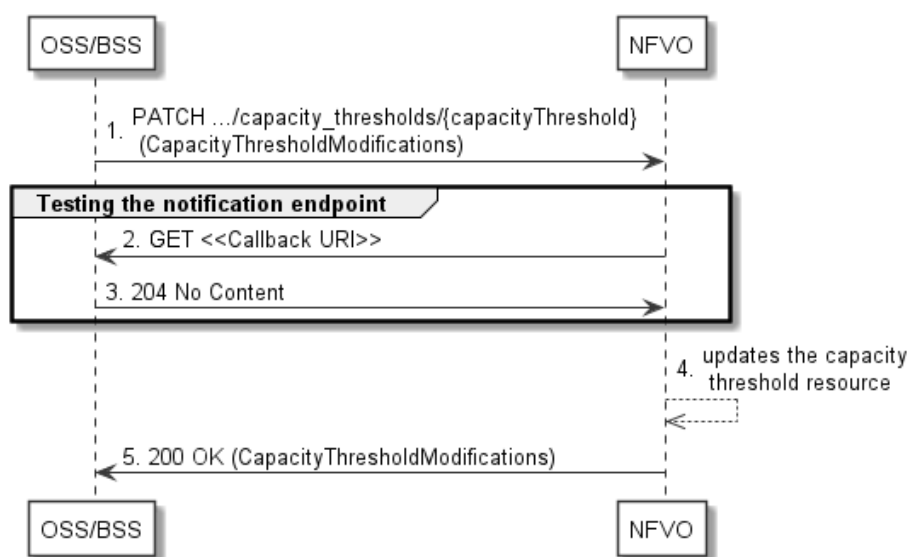
- 1) The OSS/BSS sends a DELETE request to the "Individual capacity threshold" resource, addressed by the appropriate capacity threshold identifier in its resource URI.
- 2) The NFVO deletes the capacity threshold resource.
- 3) The NFVO returns a "204 No Content" response code to the NFVO with an empty payload body.

**Error handling:** In case of failure, appropriate error information is provided in the response.

**Postcondition:** CapacityThreshold information resource removed.

### 10.3.5 Flow of updating the callback URI of a threshold

This clause describes a sequence flow for updating the callback URI in an NFVI capacity threshold.



**Figure 10.3.5-1: Flow of updating the callback URI of a threshold**

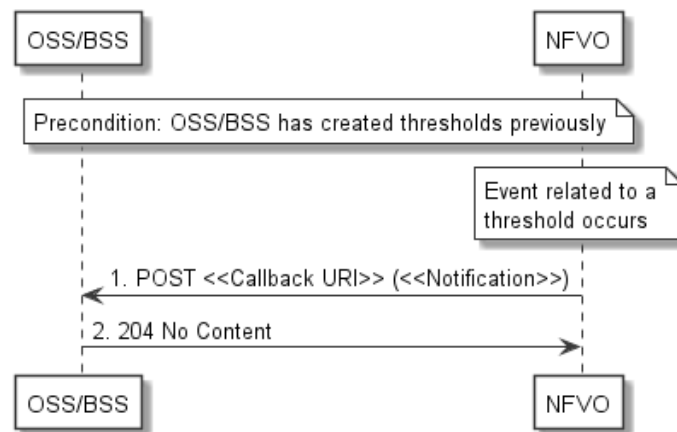
The procedure consists of the following steps as illustrated in figure 10.3.5-1:

- 1) If the OSS/BSS intends to update the callback URI in a capacity threshold, it sends a PATCH request to the "Individual capacity threshold" resource including in the payload body a data structure of type "CapacityThresholdModifications".
- 2) To test the notification endpoint that is addressed by the new callback URI, the NFVO sends a GET request to the notification endpoint URI.
- 3) The OSS/BSS returns a "204 No Content" response to indicate success.
- 4) The NFVO updates the capacity threshold resource.
- 5) The NFVO returns a "200 OK" response to the OSS/BSS and includes in the payload body a data structure of type "CapacityThresholdModifications" to indicate the performed modifications.

**Error handling:** In case of failure, including an invalid notification endpoint, appropriate error information is provided in the response.

### 10.3.6 Flow of sending notifications

This clause describes the procedure for sending notifications related to NFVI capacity information.



**Figure 10.3.6-1: Flow of sending notifications**

**Precondition:** The OSS/BSS has previously created thresholds which trigger notifications related to NFVI capacity information.

The procedure consists of the following steps as illustrated in figure 10.3.6-1:

- 1) If an event occurs that matches the filtering criteria which are part of the subscription, the NFVO generates a notification that includes information about the event, and sends it in the body of a POST request to the URI which the OSS/BSS has registered as part of the subscription request. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API (see clause 10.5.2.9).
- 2) The OSS/BSS acknowledges the successful delivery of the notification by returning a "204 No Content" response.

**Error handling:** If the NFVO does not receive the "204 No Content" response from the NFVO, it can retry sending the notification.

## 10.4 Resources

### 10.4.1 Introduction

This clause defines all the resources and methods provided by the NFVI capacity information API.

#### 10.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [16] are part of the NFVI capacity information interface.

### 10.4.2 Resource: NFVI capacity information

#### 10.4.2.1 Description

This resource represents NFVI capacity information. The API consumer can use this resource to query NFVI capacity information.

#### 10.4.2.2 Resource definition

The resource URI is:

**{apiRoot}/nfvi/{apiMajorVersion}/nfvi\_capacity\_infos**

This resource shall support the resource URI variables defined in table 10.4.2.2-1.



**Table 10.4.2.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 10.1a.

### 10.4.2.3 Resource methods

#### 10.4.2.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 10.4.2.3.2 GET

The API consumer can use this method to retrieve information about NFVI capacity information.

This method shall follow the provisions specified in the tables 10.4.2.3.2-1 and 10.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 10.4.2.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].  The NFVO shall support receiving this parameter as part of the URI query string.  The OSS/BSS may supply this parameter. If the OSS/BSS does not supply a filter value for the "timeInterval" attribute in the NfviCapacityInfo, a query of the current NFVI capacity information available by the NFVO is requested.  The following attribute names in the NfviCapacityInfo shall be supported by the NFVO in the filter expression: <ul style="list-style-type: none"> <li>- resourceZoneId</li> <li>- vimId</li> <li>- timeInterval</li> <li>- resourceType</li> <li>- capacityMeasurementName</li> <li>- capacity values: {totalCapacity, allocatedCapacity, reservedCapacity, availableCapacity}</li> </ul>
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.  The following attributes shall be excluded from the NfviCapacityInfo structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: <ul style="list-style-type: none"> <li>- capacityInfoPerZone</li> <li>- totalCapacityInfo</li> <li>- timeInterval</li> </ul>
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 10.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	NfviCapacityInfo	0..N	200 OK	<p>Shall be returned when information about NFVI capacity information has been queried successfully.</p> <p>The response body shall contain NFVI capacity information, as defined in clause 10.5.2.4.</p> <p>If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 of ETSI GS NFV-SOL 013 [16].</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute selector.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 10.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 10.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 10.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 10.4.3 Resource: Individual VIM's NFVI capacity information

### 10.4.3.1 Description

This resource represents an individual VIM's NFVI capacity information. The API consumer can use this resource to read NFVI capacity information from a specific VIM.

### 10.4.3.2 Resource definition

The resource URI is:

**{apiRoot}/nfvi/{apiMajorVersion}/nfvi\_capacity\_infos/{vimId}**

This resource shall support the resource URI variables defined in table 10.4.3.2-1.

**Table 10.4.3.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 10.1a.
vimId	Identifier of the VIM with associated NFVI capacity information. See note.
NOTE:	This identifier can be retrieved from the "vimId" attribute in the CapacityShortageNotification. It can also be retrieved from the "vimId" attribute in the payload body of the response.

### 10.4.3.3 Resource methods

#### 10.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 10.4.3.3.2 GET

The API consumer can use this method for reading an individual VIM's NFVI capacity information.

This method shall follow the provisions specified in the tables 10.4.3.3.2-1 and 10.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 10.4.3.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this parameter as part of the URI query string.</p> <p>The OSS/BSS may supply this parameter. If the OSS/BSS does not supply a filter value for the timeInterval attribute in the NfviCapacityInfo, a query of the current NFVI capacity information available by the NFVO is requested.</p> <p>The following attribute names in the NfviCapacityInfo shall be supported by the NFVO in the filter expression:</p> <ul style="list-style-type: none"> <li>- resourceZoneld</li> <li>- timeInterval</li> <li>- resourceType</li> <li>- capacityMeasurementName</li> <li>- capacity values: {totalCapacity, allocatedCapacity, reservedCapacity, availableCapacity}</li> </ul>

Table 10.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	NfviCapacityInfo	1	200 OK	Shall be returned when information of an individual VIM's NFVI capacity has been read successfully.  The response body shall contain a representation of the NFVI capacity information, as defined in clause 10.5.2.4.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 10.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 10.4.3.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 10.4.3.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 10.4.4 Resource: Capacity thresholds

#### 10.4.4.1 Description

This resource represents NFVI capacity thresholds. The API consumer can use this resource to create and query NFVI capacity thresholds.

#### 10.4.4.2 Resource definition

The resource URI is:

**{apiRoot}/nfvi/{apiMajorVersion}/capacity\_thresholds**

This resource shall support the resource URI variables defined in table 10.4.4.2-1.

Table 10.4.4.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 10.1a.

### 10.4.4.3 Resource methods

#### 10.4.4.3.1 POST

The POST method creates a new NFVI capacity threshold.

As a result of successfully executing this method, a new "Individual capacity threshold" resource as defined in clause 10.4.5 shall have been created.

This method shall follow the provisions specified in the tables 10.4.4.3.1-1 and 10.4.4.3.1-2 for URI query parameters, request and response data structures, and response codes.

**Table 10.4.4.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 10.4.4.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	CreateCapacityThresholdRequest	1	Request parameters to create a new "Individual capacity threshold" resource.	
Response body	Data type	Cardinality	Response Codes	Description
	CapacityThreshold	1	201 Created	<p>Shall be returned when a capacity threshold has been created successfully.</p> <p>The response body shall contain a representation of the created "Individual capacity threshold" resource, as defined in clause 10.5.2.8.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created resource.</p>
	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 10.4.6.3.2 and the test has failed.</p> <p>In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

## 10.4.4.3.2 GET

The API consumer can use this method to query information about NFVI capacity thresholds.

This method shall follow the provisions specified in the tables 10.4.4.3.2-1 and 10.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 10.4.4.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].  The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.  All attribute names that appear in the CapacityThreshold data type and in data types referenced from it shall be supported by the NFVO in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

NOTE: There are no attribute selectors defined for this resource as the threshold attributes with cardinality 0..1 or 0..N are not structurally complex in nature.

**Table 10.4.4.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	CapacityThreshold	0..N	200 OK	Shall be returned when information about zero or more capacity thresholds has been queried successfully.  If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [16].  The response body shall contain in an array the representations of zero or more capacity thresholds, as defined in clause 10.5.2.8.  If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression.  The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big.  If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 10.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 10.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 10.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 10.4.5 Resource: Individual capacity threshold

#### 10.4.5.1 Description

This resource represents an individual NFVI capacity threshold. The API consumer can use this resource to query information about the NFVI capacity threshold or delete the "Individual capacity threshold" resource.

#### 10.4.5.2 Resource definition

The resource URI is:

**{apiRoot}/nfvi/{apiMajorVersion}/capacity\_thresholds/{capacityThresholdId}**

This resource shall support the resource URI variables defined in table 10.4.5.2-1.

**Table 10.4.5.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 10.1a.
capacityThresholdId	Identifier of the capacity threshold. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual capacity threshold" resource. It can also be retrieved from the "id" attribute in the payload body of that response.

#### 10.4.5.3 Resource methods

##### 10.4.5.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

##### 10.4.5.3.2 GET

The API consumer can use this method for reading information about an NFVI capacity threshold.

This method shall follow the provisions specified in the tables 10.4.5.3.2-1 and 10.4.5.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 10.4.5.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

Table 10.4.5.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	CapacityThreshold	0..N	200 OK	Shall be returned when information about an individual capacity threshold has been read successfully.  The response body shall contain in a representation of the capacity threshold, as defined in clause 10.5.2.8.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 10.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 10.4.5.3.4 PATCH

This method allows to modify an "Individual capacity threshold" resource.

This method shall follow the provisions specified in the tables 10.4.5.3.4-1 and 10.4.5.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 10.4.5.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 10.4.5.3.4-2: Details of the PATCH request/response on this resource

Request body	Data type	Cardinality	Description	
	CapacityThresholdModifications	1		Parameters for the capacity threshold modification.  The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [11].
Response body	Data type	Cardinality	Response Codes	Description
	CapacityThresholdModifications	1	200 OK	Shall be returned when the request has been processed successfully.  The response body shall contain a data structure of type "CapacityThresholdModifications".
	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled.  Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity.  The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.



	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 10.4.6.3.2 and the test has failed.</p> <p>In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 10.4.5.3.5 DELETE

This method allows to delete an NFVI capacity threshold.

As a result of successfully executing this method, the "Individual capacity threshold" resource shall not exist any longer.

This method shall follow the provisions specified in the tables 10.4.5.3.5-1 and 10.4.5.3.5-2 for URI query parameters, request and response data structures, and response codes.

**Table 10.4.5.3.5-1: URI query parameters supported by the DELETE method on this resource**

Name	Cardinality	Description
none supported		

**Table 10.4.5.3.5-2: Details of the DELETE request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	<p>Shall be returned when the NFVI capacity threshold has been deleted successfully.</p> <p>The response body shall be empty.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 10.4.6 Resource: Notification endpoint

### 10.4.6.1 Description

This resource represents a notification endpoint for NFVI capacity information.

The API producer can use this resource to send notifications related to NFVI capacity information events to an API consumer which has provided the URI of this resource during the capacity threshold creation process.

### 10.4.6.2 Resource definition

The resource URI is provided by the API consumer when creating the capacity threshold.

This resource shall support the resource URI variables defined in table 10.4.6.2-1.

**Table 10.4.6.2-1: Resource URI variables for this resource**

Name	Definition
none supported	

### 10.4.6.3 Resource methods

#### 10.4.6.3.1 POST

The POST method delivers a notification regarding an NFVI capacity information event from the API producer to an API consumer. The API consumer shall have previously created an "Individual capacity threshold" resource.

This method shall follow the provisions specified in the tables 10.4.6.3.1-1 and 10.4.6.3.1-2 for URI query parameters, request and response data structures, and response codes.

**Table 10.4.6.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 10.4.6.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	CapacityShortageNotification	1	Notification about the available NFVI capacity having crossed below a threshold value or having re-covered from a capacity shortage.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the notification has been delivered successfully.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 10.4.6.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during creation of the capacity threshold resource.

This method shall follow the provisions specified in the tables 10.4.6.3.2-1 and 10.4.6.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 10.4.6.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 10.4.6.3.2-2: Details of the GET request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned to indicate that the notification endpoint has been tested successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 10.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 10.4.6.3.4 PATCH

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 10.4.6.3.5 DELETE

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 10.5 Data Model

### 10.5.1 Introduction

This clause defines the request and response data structures of the NFVI capacity information interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error, and may choose to ignore them.

### 10.5.2 Resource and notification data types

#### 10.5.2.1 Introduction

This clause defines the data structures to be used in resource representations and notifications.

#### 10.5.2.2 Type: CreateCapacityThresholdRequest

This type represents a request to create a capacity threshold. It shall comply with the provisions defined in table 10.5.2.2-1.

**Table 10.5.2.2-1: Definition of the CreateCapacityThresholdRequest data type**

Attribute name	Data type	Cardinality	Description
objectInstanceId	Identifier	1	Identifier of the VIM's NFVI capacity information instance (measured object instance) associated with this capacity threshold.
subObjectInstanceIds	IdentifierInVim	0..N	Identifiers of the sub-object instances of the measured object instance associate with this capacity threshold.  If this attribute is absent, measurements will be taken for all sub-object instances of the measured object instance.
criteria	CapacityThresholdCriteria	1	Criteria that define this capacity threshold.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this capacity threshold, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [16].  This attribute shall only be present if the API consumer requires authorization of notifications.

### 10.5.2.3 Type: NfviCapacityInfo

This type defines the format of the NFVI capacity information. The type shall comply with the provisions defined in table 10.5.2.3-1.

**Table 10.5.2.3-1: Definition of the NfviCapacityInfo data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this NFVI capacity information. Its value shall be the same as vimId.
vimId	Identifier	1	Identifies the VIM to which the NFVI capacity information is associated.
capacityInfoPerZone	NfviCapacityInfoPerZone	1..N	Capacity information on a per resource zone basis under control by the associated VIM.
totalCapacityInfo	NfviCapacityMeasurement	1..N	Total capacity information associated to the VIM.
timeInterval	TimeInterval	0..1	Time interval of the NFVI capacity information.

### 10.5.2.4 Type: NfviCapacityInfoPerZone

This type defines the format of the NFVI capacity information on a per resource zone basis. The type shall comply with the provisions defined in table 10.5.2.4-1.

**Table 10.5.2.4-1: Definition of the NfviCapacityInfoPerZone data type**

Attribute name	Data type	Cardinality	Description
resourceZoneId	Identifier	0..1	Identifies the resource zone of the applicable NFVI capacity information.
capacityMeasurements	NfviCapacityMeasurement	1..N	Capacity measurement on a per resource type basis.

### 10.5.2.5 Type: NfviCapacityMeasurement

This type defines the format of the NFVI capacity information on a per resource type basis. The type shall comply with the provisions defined in table 10.5.2.5-1.

**Table 10.5.2.5-1: Definition of the NfviCapacityMeasurement data type**

Attribute name	Data type	Cardinality	Description
resourceType	NfviCapacityResourceTypeEnumeration	1	Type of resource.
capacityMeasurementName	String	1	Name of the capacity measurement. Different resource types can have different associated capacity measurements, typically associated to different sub-types of the resource type. See note.
totalCapacity	(any type)	1	The total capacity.
allocatedCapacity	(any type)	1	The allocated/used capacity.
reservedCapacity	(any type)	0..1	The reserved capacity. Shall be present if the capacity measurement relates to a resource type and (and sub-type) that is reservable.
availableCapacity	(any type)	1	The available capacity.
NOTE: The present document and referred documents do not specify the capacity measurements, thus the capacity measurement names are not specified in the present document version.			

### 10.5.2.6 Type: TimeInterval

This type defines the format of a time interval. The type shall comply with the provisions defined in table 10.5.2.6-1.

**Table 10.5.2.6-1: Definition of the TimeInterval data type**

Attribute name	Data type	Cardinality	Description
aTime	DateTime	1	First date and time of the interval. See note.
bTime	DateTime	0..1	Second date and time of the interval. Shall be present when a time interval is provided, and absent otherwise. When provided, the bTime shall be greater than aTime. See note.
NOTE: When only the start time is present, there is no time interval being defined, and therefore the provided timing information refers to a specific point in time.			

### 10.5.2.7 Type: CapacityThreshold

This type represents a capacity threshold. It shall comply with the provisions defined in table 10.5.2.7-1.

**Table 10.5.2.7-1: Definition of the CapacityThreshold data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this capacity threshold resource.
objectInstanceId	Identifier	1	Identifier of the VIM's NFVI capacity information instance (measured object instance) associated with the capacity threshold. See note 1.
subObjectInstanceIds	IdentifierInVim	0..N	Identifiers of the sub-object instances of the measured object instance associate with this capacity threshold.  If this attribute is absent, measurements are taken for all sub-object instances of the measured object instance.  See note 2.
criteria	CapacityThresholdCriteria	1	Criteria that define this capacity threshold.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.
>object	Link	0..1	Link to a resource representing the VIM's NFVI capacity information. Shall be present if the VIM's NFVI capacity information is accessible as a resource.

Attribute name	Data type	Cardinality	Description
NOTE 1: The "objectInstancelid" aims to identify the "Individual VIM's NFVI capacity information", which is associated to a VIM instance.			
NOTE 2: The "subObjectInstancelids" aim to identify the resource zones in which the available NFVI capacity crosses a threshold value.			

### 10.5.2.8 Type: CapacityThresholdModifications

This type represents modifications to a capacity threshold. It shall comply with the provisions defined in table 10.5.2.8-1.

**Table 10.5.2.8-1: Definition of the CapacityThresholdModifications data type**

Attribute name	Data type	Cardinality	Description
callbackUri	Uri	0..1	New value of the "callbackUri" attribute. The value "null" is not permitted. See note.
authentication	SubscriptionAuthentication	0..1	New value of the "authentication" attribute, or "null" to remove the attribute. If present in a request body, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [11]).  This attribute shall not be present in response bodies. See note.
NOTE: At least one of the attributes defined in this type shall be present in request bodies.			

### 10.5.2.9 Type: CapacityShortageNotification

This notification informs the receiver that the available NFVI capacity has crossed below a threshold value or has re-covered from a capacity shortage. It shall comply with the provisions defined in table 10.5.2.9-1.

**Table 10.5.2.9-1: Definition of the CapacityShortageNotification data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "CapacityShortageNotification" for this notification type.
thresholdId	Identifier	1	Identifier of the threshold which has been crossed.
timeStamp	DateTime	1	Date and time of the generation of the notification.
objectInstancelid	Identifier	1	Identifies the VIM's NFVI capacity information instance (measured object instance) in which the available NFVI capacity has crossed a threshold value. See note.
subObjectInstancelid	IdentifierInVim	0..1	Identifier of the sub-object of the measured object to which the measurement applies. See note.
direction	Enum {UP; DOWN}	1	Specifies if the threshold has been crossed in UP or DOWN direction.

Attribute name	Data type	Cardinality	Description
capacityInformation	NfviCapacityMeasurement	1	Information about the available, reserved, allocated/used, and total capacity of the NFVI. If the threshold creation process does specify a resource zone and/or vimId, the information is provided for the resource zone/vimId where the NFVI capacity has crossed the threshold as indicated by the "subObjectInstanceid" and "objectInstanceid" attributes.
_links	Structure (inlined)	1	Links to resources related to this notification.
>objectInstance	NotificationLink	0..1	Link to the resource representing the measured object instance to which the notified change applies. Shall be present if the measured object instance information is accessible as a resource.
>threshold	NotificationLink	1	Link to the resource that represents the threshold that was crossed.
NOTE: Refer to the definition of the "CapacityThreshold" in clause 10.5.2.7.			

## 10.5.3 Referenced structured data types

### 10.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but can neither be resource representations nor bound to any subscribe/notify mechanism.

### 10.5.3.2 Type: CapacityThresholdCriteria

This type represents criteria that define a capacity threshold. It shall comply with the provisions defined in table 10.5.3.2-1.

**Table 10.5.3.2-1: Definition of the CapacityThresholdCriteria data type**

Attribute name	Data type	Cardinality	Description
capacityMetric	Structure (inlined)	1	Defines the capacity metric for which the threshold applies.
>resourceType	NfviCapacityResourceTypeEnumeration	1	Type of resource.
>capacityMeasurementName	String	1	Name of the capacity measurement. Different resource types can have different associated capacity measurements, typically associated to different sub-types of the resource type.
>capacityType	Enum (inlined)	1	The type of capacity for the threshold.  Permitted values: <ul style="list-style-type: none"> <li>- TOTAL: for total capacity.</li> <li>- ALLOCATED: for allocated/used capacity.</li> <li>- RESERVED: for reserved capacity.</li> <li>- AVAILABLE: for available capacity.</li> </ul>
thresholdType	Enum (inlined)	1	Type of capacity threshold. This attribute determines which other attributes are present in the data structure.  Permitted values: <ul style="list-style-type: none"> <li>- SIMPLE: Single-valued static threshold.</li> </ul> See note 1.
simpleThresholdDetails	Structure (inlined)	0..1	Details of a simple threshold. Shall be present if thresholdType="SIMPLE".
>thresholdValue	Number	1	The threshold value. Shall be represented as a floating point number.

Attribute name	Data type	Cardinality	Description
>hysteresis	Number	1	<p>The hysteresis of the threshold.</p> <p>Shall be represented as a non-negative floating point number.</p> <p>A notification with crossing direction "UP" will be generated if the measured value reaches or exceeds "thresholdValue" + "hysteresis".</p> <p>A notification with crossing direction "DOWN" will be generated if the measured value reaches or undercuts "thresholdValue" - "hysteresis".</p> <p>See note 2.</p>
<p>NOTE 1: In the present document, simple thresholds are defined. The definition of additional threshold types is left for future specification.</p> <p>NOTE 2: The hysteresis is defined to prevent storms of threshold crossing notifications. When processing a request to create a threshold, implementations should enforce a suitable minimum value for this attribute (e.g. override the value or reject the request).</p>			

## 10.5.4 Referenced simple data types and enumerations

### 10.5.4.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

### 10.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.2.

### 10.5.4.3 Enumeration: NfviCapacityResourceTypeEnumeration

The enumeration NfviCapacityResourceTypeEnumeration shall comply with the provisions in table 10.5.4.3-1.

**Table 10.5.4.3-1: Enumeration NfviCapacityResourceTypeEnumeration**

Enumeration value	Description
VR_COMPUTE	For virtualised compute resource.
VR_NETWORK	For virtualised network resource.
VR_STORAGE	For virtualised storage resource.
HOST_COMPUTE	For host compute resource.

# 11 VNF Snapshot Package Management interface

## 11.1 Description

This interface allows the OSS/BSS to query and update the VNF snapshot package information and to fetch, create, upload, build, extract, and delete VNF snapshot packages. The "Create VNF snapshot package" operation is designed as a 2-step operation, whereby first a new "Individual VNF snapshot package" resource is created based on the VNF Snapshot metadata. Then, second, either a new VNF snapshot package is built or an existing VNF snapshot package is uploaded.

The operations provided through this interface are:

- Create VNF Snapshot Package Info
- Build VNF Snapshot Package



- Upload VNF Snapshot Package
- Extract VNF Snapshot Package
- Fetch VNF Snapshot Package
- Fetch VNF Snapshot Package Artifacts
- Query VNF Snapshot Package Information
- Delete VNF Snapshot Package
- Update VNF Snapshot Package

This interface also enables to invoke error handling procedures (Cancel operation) of an ongoing VNF snapshot package management operation, and API version information retrieval.

## 11.1a API version

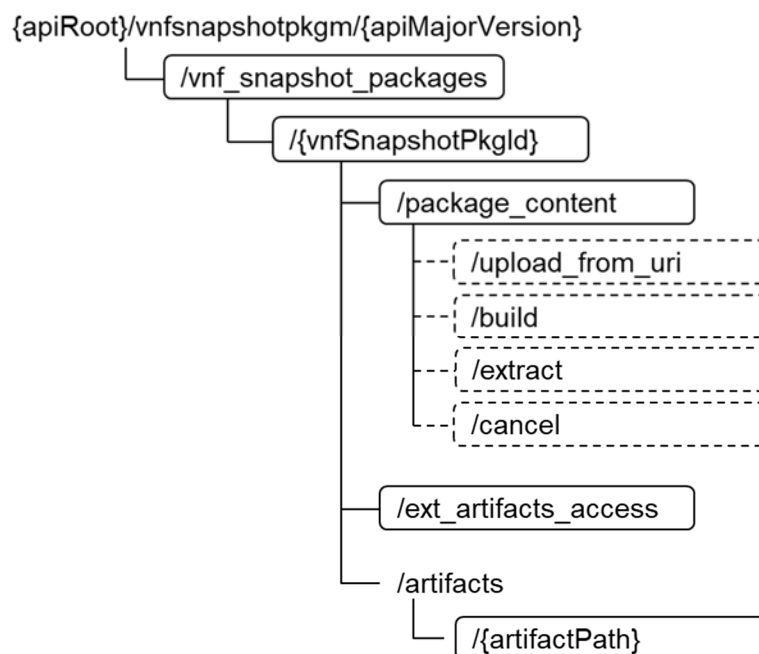
For the VNF snapshot package management interface as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 10 and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [16] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v1".

## 11.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [16].

The string "vnfsnapshotpkgm" shall be used to represent {apiName}. All resource URIs in clauses below are defined relative to the above base URI.

Figure 11.2-1 shows the overall resource URI structure defined for the VNF snapshot package management interface.



**Figure 11.2-1: Resource URI structure of the VNF snapshot package management interface**

Table 11.2-1 lists the individual resources defined, and the applicable HTTP methods.

The NFVO shall support responding to requests for all HTTP methods on the resources in table 11.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [16].

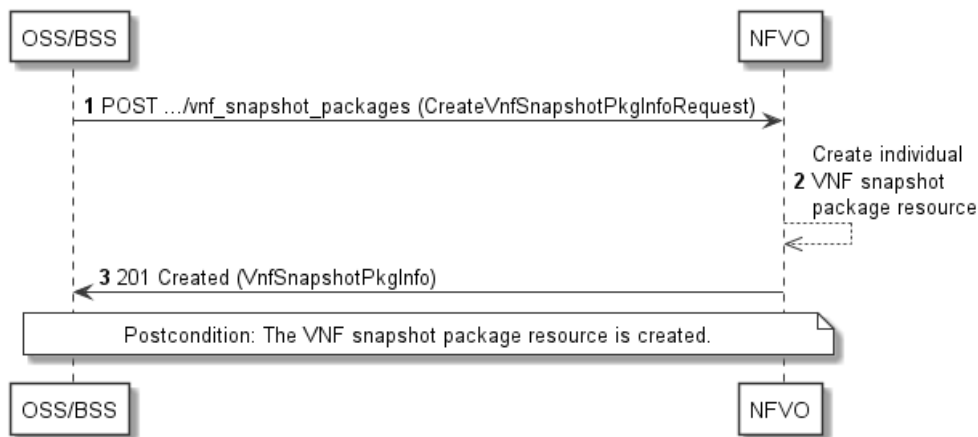
**Table 11.2-1: Resources and methods overview of the VNF snapshot package management interface**

Resource name	Resource URI	HTTP Method	Cat	Meaning
VNF snapshot packages	/vnf_snapshot_packages	GET	M	Query multiple VNF snapshot packages information.
		POST	M	Create a new "Individual VNF snapshot package" resource.
Individual VNF snapshot package	/vnf_snapshot_packages/{vnfSnapshotPkgId}	GET	M	Read an "Individual VNF snapshot package" resource.
		PATCH	M	Update information about an individual VNF snapshot package.
		DELETE	M	Delete an individual VNF snapshot package.
VNF snapshot package content	/vnf_snapshot_packages/{vnfSnapshotPkgId}/package_content	GET	M	Fetch VNF snapshot package.
		PUT	M	Upload a VNF snapshot package by providing the content of the VNF snapshot package.
Upload VNF snapshot package from URI task	/vnf_snapshot_packages/{vnfSnapshotPkgId}/package_content/upload_from_uri	POST	M	Upload a VNF snapshot package by providing the address information of the VNF snapshot package.
Build VNF snapshot package content task	/vnf_snapshot_packages/{vnfSnapshotPkgId}/package_content/build	POST	M	Build the content of a VNF snapshot package by providing a reference to the VNF snapshot to package.
Extract VNF snapshot package content task	/vnf_snapshot_packages/{vnfSnapshotPkgId}/package_content/extract	POST	M	Extract the content of a VNF snapshot package.
Cancel VNF snapshot package operation task	/vnf_snapshot_packages/{vnfSnapshotPkgId}/package_content/cancel	POST	M	Cancel any current operation with the content of a VNF snapshot package.
Access configuration for external artifacts download	/vnf_snapshot_packages/{vnfSnapshotPkgId}/ext_artifacts_access	PUT	M	Provide access configuration information for the download of external artifacts.
		GET	M	Read access configuration information for the download of external artifacts.
Individual VNF snapshot package artifact	/vnf_snapshot_packages/{vnfSnapshotPkgId}/artifacts/{artifactPath}	GET	M	Fetch individual VNF snapshot package artifact.

## 11.3 Sequence diagrams (informative)

### 11.3.1 Flow of the creation of an "Individual VNF snapshot package" resource

This clause describes the procedure for creating an "Individual VNF snapshot package" resource.



**Figure 11.3.1-1: Flow of the creation of an "Individual VNF snapshot package" resource**

Creation of an "Individual VNF snapshot package" resource, as illustrated in figure 11.3.1-1, consists of the following steps:

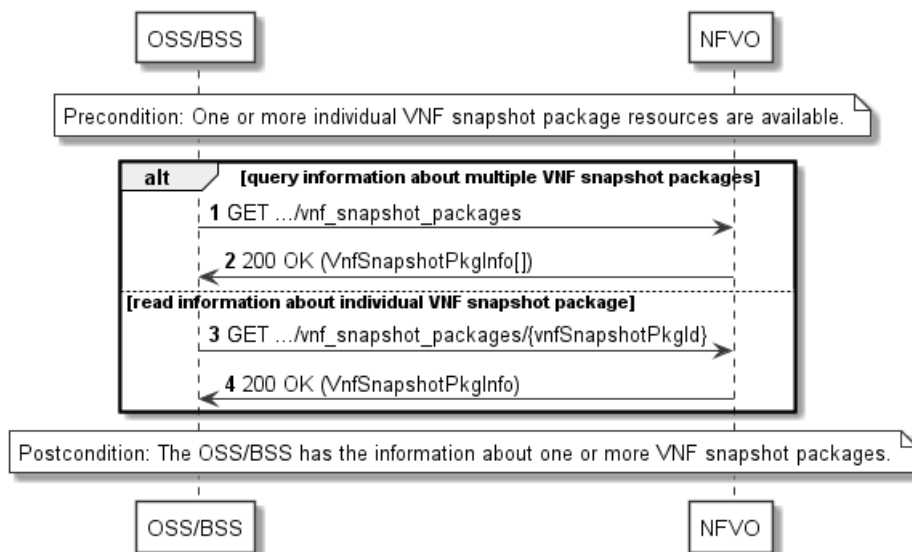
- 1) The OSS/BSS sends to the NFVO a POST request to the "VNF snapshot packages" resource including in the payload body a data structure of type "CreateVnfSnapshotPkgInfoRequest".
- 2) The NFVO creates a new "Individual VNF snapshot package" resource.
- 3) The NFVO returns a "201 Created" response containing a representation of the "Individual VNF snapshot package" resource and a "Location" HTTP header that points to the new "Individual VNF snapshot package" resource.

**Postcondition:** Upon successful completion, the "Individual VNF snapshot package" resource is created with the value of the "state" attribute equals to "CREATED".

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 11.3.2 Flow of querying/reading VNF snapshot package information

This clause describes the procedure for querying information about one or multiple VNF snapshot packages.



**Figure 11.3.2-1: Flow of querying/reading VNF snapshot package information**

**Precondition:** One or more "Individual VNF package" resources are created.

VNF snapshot package information query, as illustrated in figure 11.3.2-1, consists of the following steps:

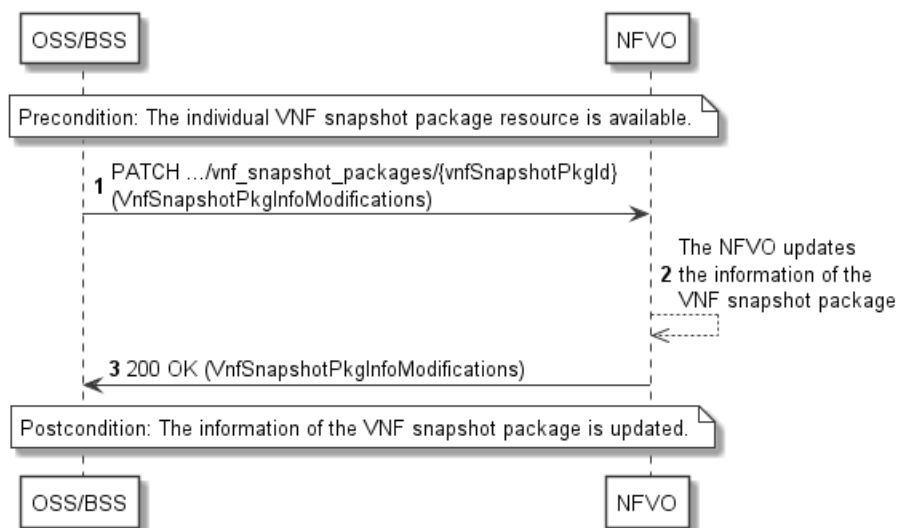
- 1) If the OSS/BSS intends to query information about multiple VNF snapshot packages, it sends a GET request to the "VNF snapshot packages" resource.
- 2) The NFVO returns a "200 OK" response and includes in the payload body zero or more data structures of type "VnfSnapshotPkgInfo".
- 3) If the OSS/BSS intends to read information about a particular VNF snapshot package, the OSS/BSS sends a GET request to the "Individual VNF snapshot package" resource, addressed by the appropriate VNF snapshot package identifier in its resource URI.
- 4) The NFVO returns a "200 OK" response and includes in the payload body a data structure of type "VnfSnapshotPkgInfo".

**Postcondition:** Upon successful completion, the OSS/BSS gets the information about one or more VNF snapshot packages.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 11.3.3 Flow of updating information of a VNF snapshot package

This clause describes the procedure for updating information of a VNF snapshot package.



**Figure 11.3.3-1: Flow of updating information of a VNF snapshot package**

**Precondition:** The VNF snapshot package is available.

The procedure consists of the following steps as illustrated in figure 11.3.3-1:

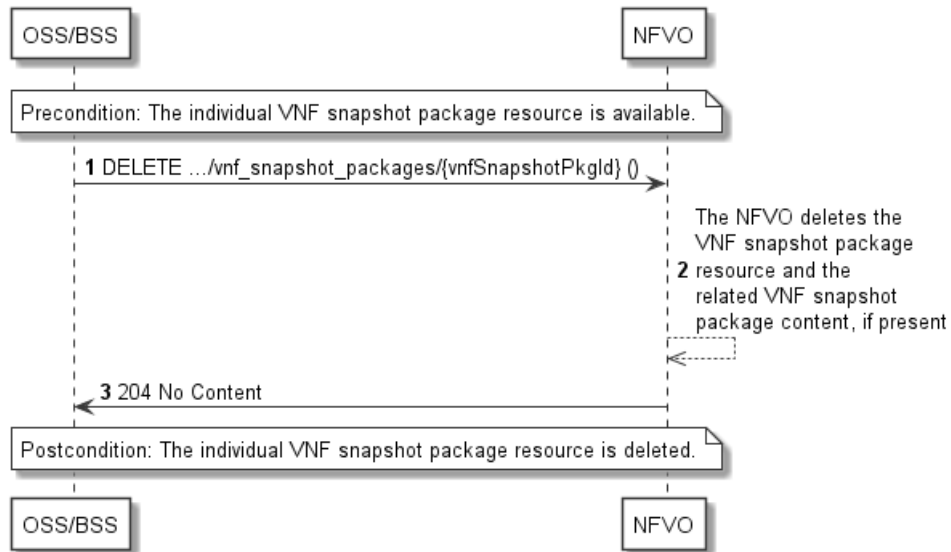
- 1) The OSS/BSS sends a PATCH request to the "Individual VNF snapshot package" resource including in the payload body a data structure of type "VnfSnapshotPkgInfoModifications".
- 2) The NFVO updates the information of the VNF snapshot package.
- 3) The NFVO returns a "200 OK" response with a payload body containing a data structure of type "VnfSnapshotPkgInfoModifications".

**Postcondition:** The information of the VNF snapshot package is updated.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 11.3.4 Flow of deleting a VNF snapshot package resource

This clause describes a sequence for deleting a VNF package resource.



**Figure 11.3.4-1: Flow of deleting a VNF package resource**

**Precondition:** The "Individual VNF snapshot package" resource is available.

Deleting a VNF snapshot package resource, as illustrated in figure 11.3.4-1, consists of the following steps:

- 1) The OSS/BSS sends a DELETE request to the "Individual VNF snapshot package" resource.
- 2) The NFVO deletes the "Individual VNF snapshot package" resource and related VNF snapshot package content, if present.
- 3) The NFVO returns a "204 No Content" response with an empty payload body.

**Postcondition:** Upon successful completion, the "Individual VNF snapshot package" resource is deleted.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 11.3.5 Flow of the uploading of VNF snapshot package content

This clause describes the procedure of uploading the content of a VNF snapshot package.

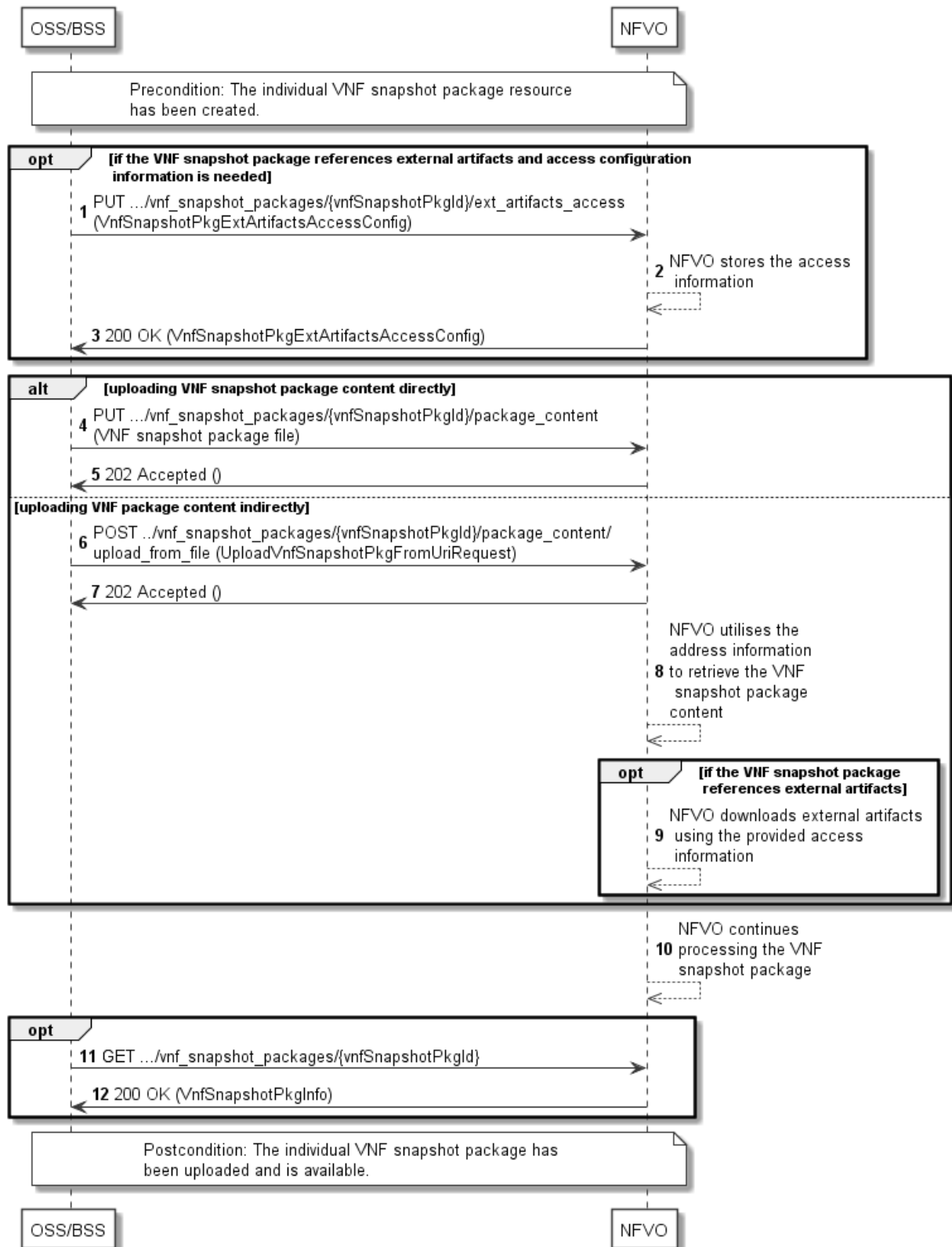


Figure 11.3.5-1: Flow of the uploading of VNF snapshot package content

**Precondition:** The "Individual VNF snapshot package" resource has been created.

Uploading the content of a VNF snapshot package, as illustrated in figure 11.3.5-1, consists of the following steps:

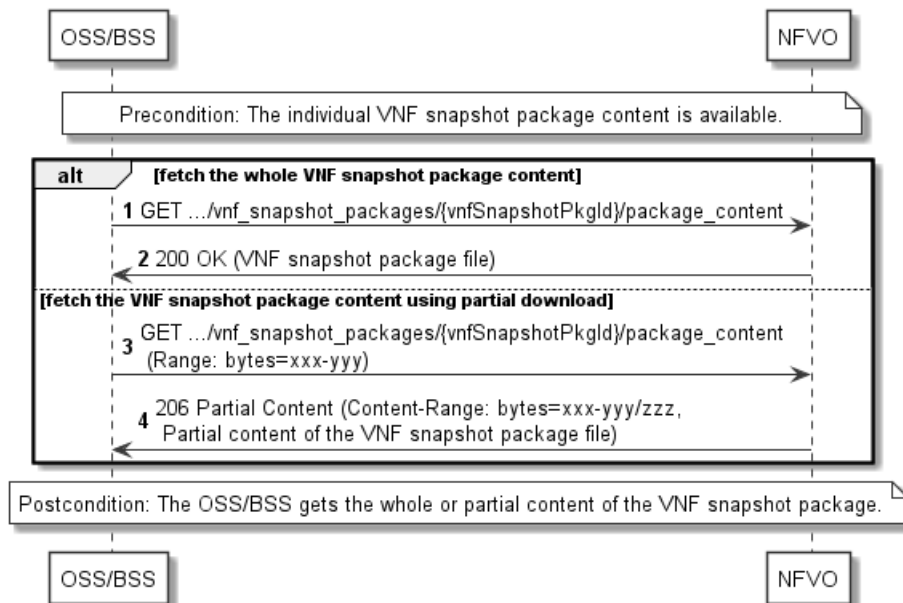
- 1) If the VNF snapshot package to be uploaded references external artifacts and access configuration information is not available to the NFVO, the OSS/BSS provides access configuration information for the subsequent download of the external VNF snapshot package artifacts to the NFVO by sending a PUT request to the "Access configuration for external artifacts" resource including in the payload body a data structure of type "VnfSnapshotPkgExtArtifactsAccessConfig".
- 2) In case step 1) was performed, the NFVO stores the information.
- 3) In case step 1) was performed, the NFVO returns a "200 OK" response with a copy of the stored information included in the payload body to indicate that the access configuration information was successfully stored.
- 4) If the OSS/BSS uploads the VNF package content directly to the NFVO, it sends a PUT request to the "VNF snapshot package content" resource including in the payload body a copy of the VNF snapshot package content.
- 5) The NFVO returns a "202 Accepted" response with an empty payload body.
- 6) If the OSS/BSS uploads the VNF snapshot package content indirectly to the NFVO, it sends a POST request to the "Upload VNF snapshot package from URI task" resource including in the payload body a data structure of type "UploadVnfSnapshotPkgFromUriRequest".
- 7) The NFVO returns a "202 Accepted" response with an empty payload body to indicate the address information is successfully received.
- 8) The NFVO utilizes the address information to retrieve the VNF snapshot package content.
- 9) If the VNF snapshot package references external artifacts, the NFVO utilizes the access configuration information and the information in the VNF snapshot package to download the external artifacts.
- 10) The NFVO continues processing the VNF snapshot package (e.g. validation) after it retrieves the package content.
- 11) Optionally, the OSS/BSS can send a GET request to the "Individual VNF snapshot package" resource to check the uploading state of the VNF snapshot package resource.
- 12) The NFVO returns a "200 OK" response containing the information of the VNF snapshot package resource.

**Postcondition:** Upon successful completion, the content of the VNF snapshot package is uploaded and is available.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 11.3.6 Flow of fetching a VNF snapshot package

This clause describes the procedure for fetching the content of an available VNF snapshot package.



**Figure 11.3.6-1: Flow of fetching a VNF snapshot package**

**Precondition:** The VNF snapshot package is available on the NFVO.

Fetching an available VNF snapshot package, as illustrated in figure 11.3.6-1, consists of the following steps:

- 1) If fetching the whole VNF snapshot package content, the OSS/BSS sends a GET request to the "VNF snapshot package content" resource.
- 2) The NFVO returns a "200 OK" response and includes a copy of the VNF snapshot package file in the payload body.
- 3) If fetching the VNF snapshot package content using partial download, the OSS/BSS sends a GET request to the "VNF snapshot package content" resource and includes a "Range" HTTP header indicating the partition of the VNF snapshot package content needs to be transferred.
- 4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the VNF snapshot package, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the VNF snapshot package content.

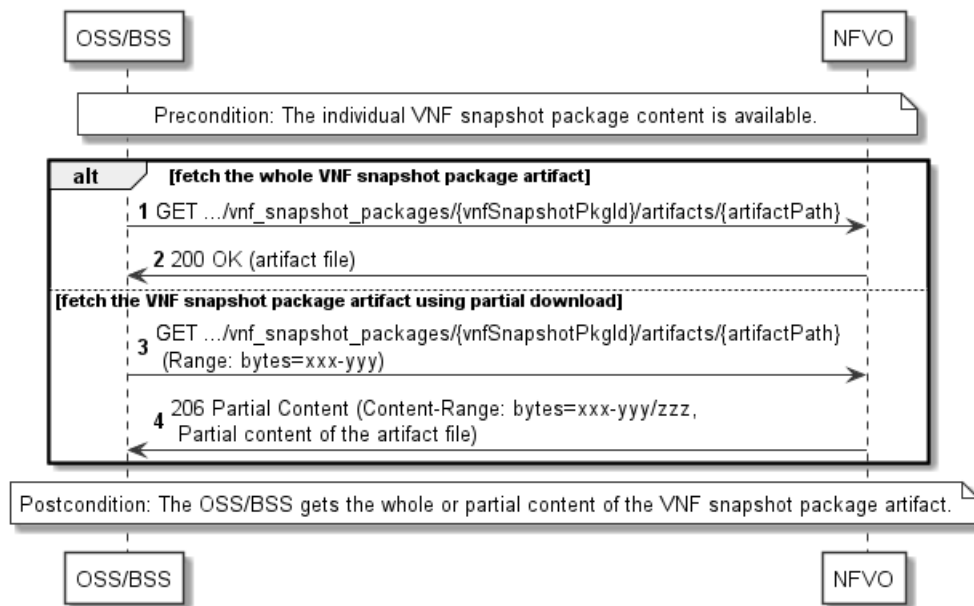
**Postcondition:** Upon successful completion, the OSS/BSS gets the whole or partial content of the VNF snapshot package.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 11.3.7 Flow of fetching a VNF snapshot package artifact

This clause describes the procedure for fetching an individual artifact contained in an available VNF snapshot package.





**Figure 11.3.7-1: Flow of fetching a VNF snapshot package artifact**

**Precondition:** The VNF snapshot package is available on the NFVO.

Fetching an individual artifact contained in an available VNF snapshot package, as illustrated in figure 11.3.7-1, consists of the following steps:

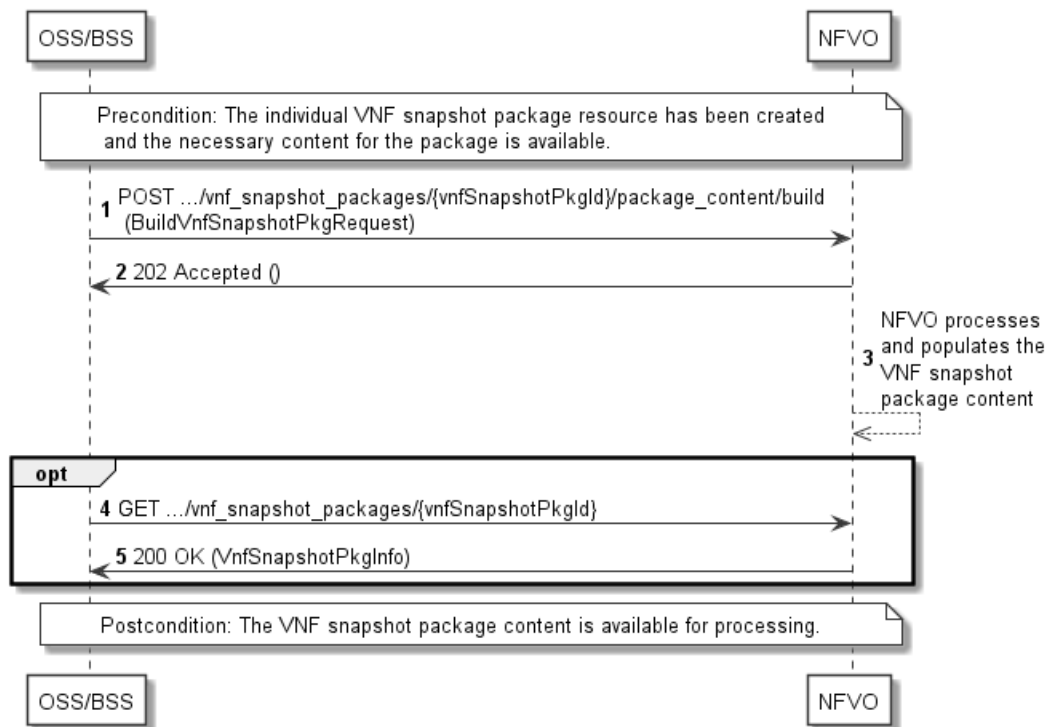
- 1) If fetching the whole content of the artifact, the OSS/BSS sends a GET request to the "Individual VNF snapshot package artifact" resource.
- 2) The NFVO returns a "200 OK" response and includes a copy of the applicable artifact file from the VNF snapshot package in the payload body.
- 3) If fetching the artifact using partial download, the OSS/BSS sends a GET request to the "Individual VNF snapshot package artifact" resource and includes a "Range" HTTP header indicating the partition of the artifact needs to be transferred.
- 4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the artifact file, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the artifact file.

**Postcondition:** The OSS/BSS gets the whole or partial content of the VNF snapshot package artifact.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 11.3.8 Flow of building the content of VNF snapshot package

This clause describes the procedure of building the content of a VNF snapshot package.



**Figure 11.3.8-1: Flow of building the content of VNF snapshot package**

**Precondition:** The "Individual VNF snapshot package" resource has been created and the necessary content for the package is available, i.e. a VNF snapshot.

Building the content of a VNF snapshot package, as illustrated in figure 11.3.8-1, consists of the following steps:

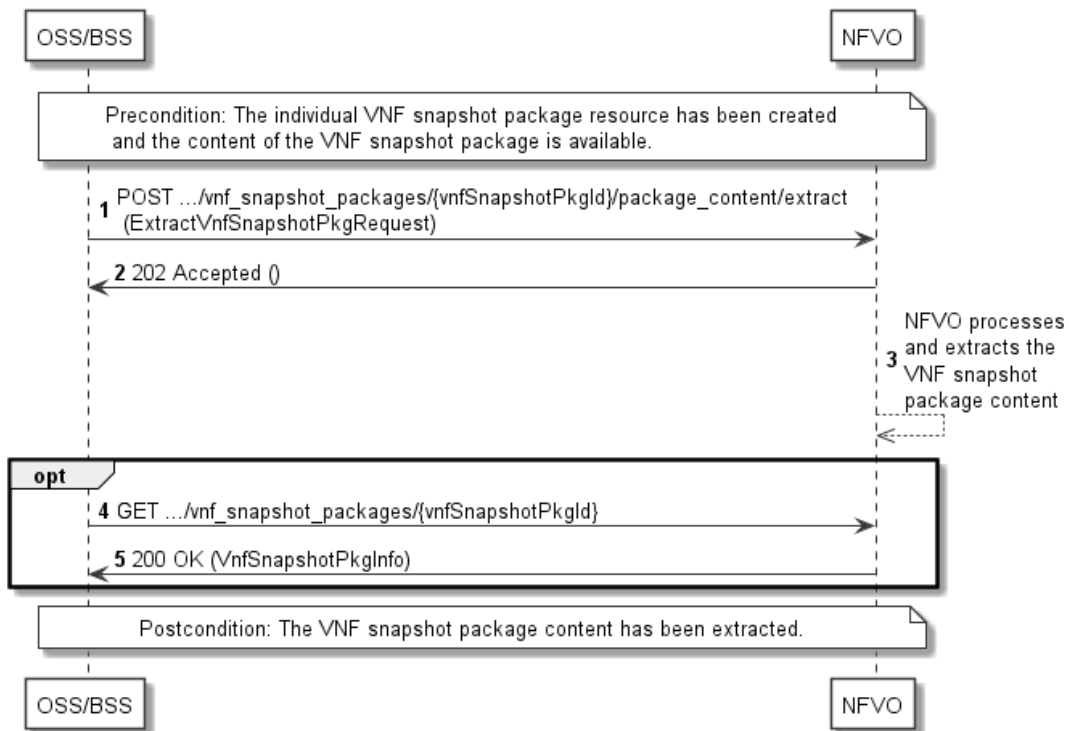
- 1) The OSS/BSS sends a POST request to the "Build VNF snapshot package content task" resource including in the payload body a data structure of type "BuildVnfSnapshotPkgRequest".
- 2) The NFVO returns a "202 Accepted" response with an empty payload body.
- 3) The NFVO continues processing the building of the VNF snapshot package by populating the VNF snapshot package content.
- 4) Optionally, the OSS/BSS can send a GET request to the "Individual VNF snapshot package" resource to check the building state of the VNF snapshot package resource.
- 5) In case of step 4), the NFVO returns a "200 OK" response containing the information of the VNF snapshot package resource.

**Postcondition:** Upon successful completion, the content of the VNF snapshot package has been built and is available for processing.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 11.3.9 Flow of extracting the content of VNF snapshot package

This clause describes the procedure of extracting the content of a VNF snapshot package.



**Figure 11.3.9-1: Flow of extracting the content of a VNF snapshot package**

**Precondition:** The "Individual VNF snapshot package" resource has been created and the content of the VNF snapshot package is available.

Extracting the content of a VNF snapshot package, as illustrated in figure 11.3.9-1, consists of the following steps:

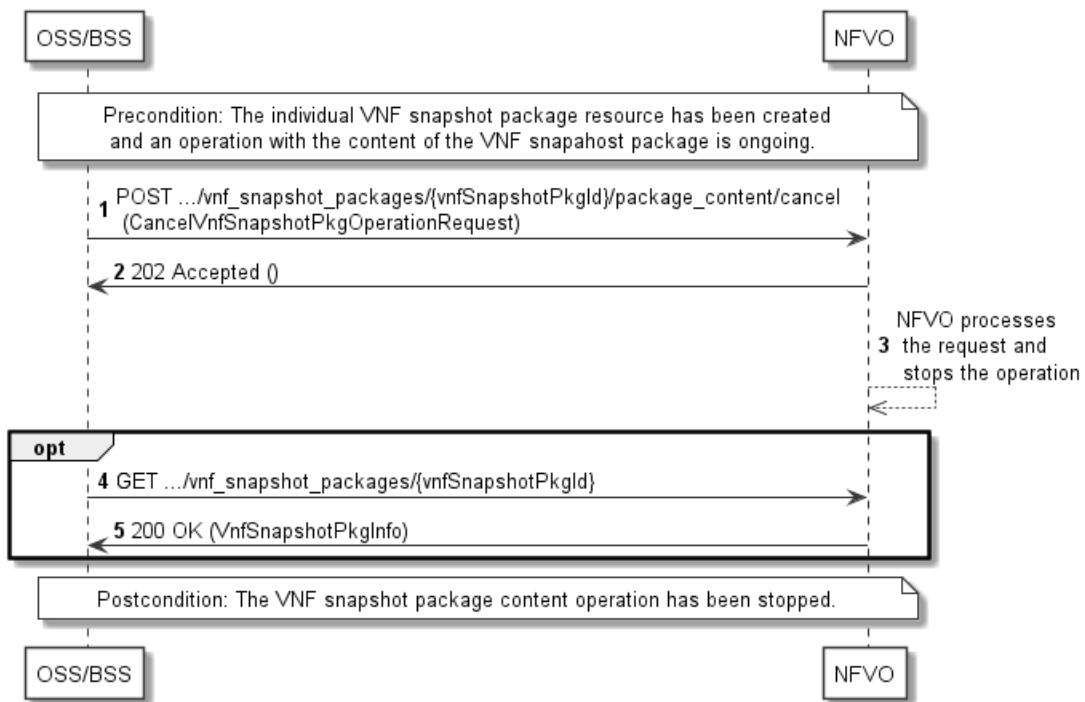
- 1) The OSS/BSS sends a POST request to the "Extract VNF snapshot package content task" resource including in the payload body a data structure of type "ExtractVnfSnapshotPkgRequest".
- 2) The NFVO returns a "202 Accepted" response with an empty payload body.
- 3) The NFVO continues processing the extraction of the VNF snapshot package from the VNF snapshot package content and populates the associated "Individual VNF snapshot" resource on the corresponding VNFM.
- 4) Optionally, the OSS/BSS can send a GET request to the "Individual VNF snapshot package" resource to check the extracting state of the VNF snapshot package resource.
- 5) In case of step 4), the NFVO returns a "200 OK" response containing the information of the VNF snapshot package resource.

**Postcondition:** Upon successful completion, the content of the VNF snapshot package has been extracted.

**Error handling:** In case of failure, appropriate error information is provided in the response.

### 11.3.10 Flow of cancelling the operation with the VNF snapshot package content

This clause describes the procedure of cancelling the operation with the VNF snapshot package content.



**Figure 11.3.10-1: Flow of cancelling the operation with the VNF snapshot package content**

**Precondition:** The "Individual VNF snapshot package" resource has been created and an operation with the content of the VNF snapshot package is ongoing.

Cancelling the operation with the VNF snapshot package content, as illustrated in figure 11.3.10-1, consists of the following steps:

- 1) The OSS/BSS sends a POST request to the "Cancel VNF snapshot package operation task" resource including in the payload body a data structure of type "CancelVnfSnapshotPkgOperationRequest".
- 2) The NFVO returns a "202 Accepted" response with an empty payload body.
- 3) The NFVO continues processing the cancellation request and stops the operation related to the VNF snapshot package content.
- 4) Optionally, the OSS/BSS can send a GET request to the "Individual VNF snapshot package" resource to check the state of the VNF snapshot package resource.
- 5) In case of step 4), the NFVO returns a "200 OK" response containing the information of the VNF snapshot package resource.

**Postcondition:** Upon successful completion, the operation related to the VNF snapshot package content has been stopped.

**Error handling:** In case of failure, appropriate error information is provided in the response.

## 11.4 Resources

### 11.4.1 Introduction

This clause defines all the resources and methods provided by the VNF snapshot package management interface.

#### 11.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [16] are part of the VNF snapshot package management interface.

## 11.4.2 Resource: VNF snapshot packages

### 11.4.2.1 Description

This resource represents VNF snapshot packages. The API consumer can use this resource to create "Individual VNF snapshot package" resources, and to query information of the VNF snapshot packages.

### 11.4.2.2 Resource definition

The resource URI is:

**{apiRoot}/vnfsnapshotpkgm/{apiMajorVersion}/vnf\_snapshot\_packages**

This resource shall support the resource URI variables defined in table 11.4.2.2-1.

**Table 11.4.2.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 11.1a.

### 11.4.2.3 Resource methods

#### 11.4.2.3.1 POST

The POST method creates a new "Individual VNF snapshot package" resource.

As the result of successfully executing this method, a new "Individual VNF snapshot package" resource as defined in clause 11.4.3 shall have been created, and the value of the "state" attribute in the representation of that resource shall be "CREATED".

This method shall follow the provisions specified in the tables 11.4.2.3.1-1 and 11.4.2.3.1-2 for URI query parameters, request and response data structures, and response codes.

**Table 11.4.2.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 11.4.2.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	CreateVnfSnapshotPkgInfoRequest	1	Individual VNF snapshot package resource creation parameters, as defined in clause 11.5.2.2.	
Response body	Data type	Cardinality	Response Codes	Description
	VnfSnapshotPkgInfo	1	201 Created	Shall be returned when an "Individual VNF snapshot package" resource has been created successfully.  The response body shall contain a representation of the new "Individual VNF snapshot package" resource, as defined in clause 11.5.2.3.  The HTTP response shall include a "Location" HTTP header that contains the resource URI of the "Individual VNF snapshot package" resource.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 11.4.2.3.2 GET

The GET method queries the information of the VNF packages matching the filter.

This method shall follow the provisions specified in the tables 11.4.2.3.2-1 and 11.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 11.4.2.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>All attribute names that appear in the VnfSnapshotPkgInfo and in data types referenced from it shall be supported by the NFVO in the filter expression.</p>
all_fields	0..1	<p>Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.</p>
fields	0..1	<p>Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.</p>
exclude_fields	0..1	<p>Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.</p>
exclude_default	0..1	<p>Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details.</p> <p>The NFVO shall support this parameter.</p> <p>The following attributes shall be excluded from the VnfSnapshotPkgInfo structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided:</p> <ul style="list-style-type: none"> <li>- vnfSnapshotImages</li> <li>- additionalArtifacts</li> <li>- userDefinedData</li> <li>- checksum</li> </ul>
nextpage_opaque_marker	0..1	<p>Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.</p>

Table 11.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
n/a				
Response body	Data type	Cardinality	Response Codes	Description
	VnfSnapshotPkgInfo	0..N	200 OK	<p>Shall be returned when information about zero or more VNF snapshot packages has been queried successfully.</p> <p>The response body shall contain in an array the VNF snapshot package info representations that match the attribute filter, i.e. zero or more VNF snapshot package info representations as defined in clause 11.5.2.3.</p> <p>If the "filter" URI parameter or one of the "all_fields", "fields", "exclude_fields" or "exclude_default" URI parameters was supplied in the request and is supported, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 and 5.3.2 of ETSI GS NFV-SOL 013 [16], respectively.</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute selector.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>

### 11.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 11.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 11.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 11.4.3 Resource: Individual VNF snapshot package

### 11.4.3.1 Description

This resource represents an individual VNF snapshot package. The API consumer can use this resource to read information of the VNF snapshot package, update information of the VNF snapshot package, or delete a VNF snapshot package.

### 11.4.3.2 Resource definition

The resource URI is:

**{apiRoot}/vnfsnapshotpkgm/{apiMajorVersion}/vnf\_snapshot\_packages/{vnfSnapshotPkgId}**

This resource shall support the resource URI variables defined in table 11.4.3.2-1.

**Table 11.4.3.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 11.1a.
vnfSnapshotPkgId	Identifier of the VNF snapshot package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "id" attribute of the applicable "VnfSnapshotPkgInfo" in the body of the response to requesting the creation of a new "Individual VNF snapshot package" resource or in a response to a GET request querying the "Individual VNF snapshot package" or the "VNF snapshot packages" resource.

### 11.4.3.3 Resource methods

#### 11.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.3.3.2 GET

The GET method reads the information of an individual VNF snapshot package.

This method shall follow the provisions specified in the tables 11.4.3.3.2-1 and 11.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 11.4.3.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		



Table 11.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	VnfSnapshotPkgInfo	1	200 OK	<p>Shall be returned when information of the VNF snapshot package has been read successfully.</p> <p>The response body shall contain the VNF snapshot package info representation defined in clause 11.5.2.3.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 11.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.3.3.4 PATCH

The PATCH method updates the information of a VNF snapshot package.

This method shall follow the provisions specified in the tables 11.4.3.3.4-1 and 11.4.3.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 11.4.3.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 11.4.3.3.4-2: Details of the PATCH request/response on this resource

Request body	Data type	Cardinality	Description	
	VnfSnapshotPkgInfoModifications	1		Parameters for VNF snapshot package information modifications.
Response body	Data type	Cardinality	Response Codes	Description
	VnfSnapshotPkgInfoModifications	1	200 OK	<p>Shall be returned when the operation has been completed successfully.</p> <p>The response body shall contain attribute modifications for an "Individual VNF snapshot package" resource.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the state of the VNF snapshot package resource is in a state other than CREATED, ERROR_EXTRACTING or AVAILABLE.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>

	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.
--	----------------	------------------------	---------	---

### 11.4.3.3.5 DELETE

The DELETE method deletes an "Individual VNF snapshot package" resource.

This method shall follow the provisions specified in the tables 11.4.3.3.5-1 and 11.4.3.3.5-2 for URI query parameters, request and response data structures, and response codes.

**Table 11.4.3.3.5-1: URI query parameters supported by the DELETE method on this resource**

Name	Cardinality	Description
none supported		

**Table 11.4.3.3.5-2: Details of the DELETE request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the VNF snapshot package has been deleted successfully.  The response body shall be empty.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.  Typically, this is due to the fact that the operational state of the VNF snapshot package resource is other than CREATED, ERROR or AVAILABLE.  The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

## 11.4.4 Resource: VNF snapshot package content

### 11.4.4.1 Description

This resource represents a VNF snapshot package identified by the VNF snapshot package identifier allocated by the NFVO. The API consumer can use this resource to upload and fetch the content of the VNF snapshot package.

### 11.4.4.2 Resource definition

The resource URI is:

**{apiRoot}/vnfsnapshotpkgm/{apiMajorVersion}/vnf\_snapshot\_packages/{vnfSnapshotPkgId}/package\_content**

This resource shall support the resource URI variables defined in table 11.4.4.2-1.

**Table 11.4.4.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 11.1a.
vnfSnapshotPkgId	Identifier of the VNF snapshot package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "id" attribute of the applicable "VnfSnapshotPkgInfo" in the body of the response to requesting the creation of a new "Individual VNF snapshot package" resource or in a response to a GET request querying the "Individual VNF snapshot package" or the "VNF snapshot packages" resource.

### 11.4.4.3 Resource methods

#### 11.4.4.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.4.3.2 GET

The GET method fetches the content of a VNF snapshot package.

The content of the package is provided as onboarded to the NFVO, or as built by the NFVO.

This method shall follow the provisions specified in the tables 11.4.4.3.2-1 and 11.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 11.4.4.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		

**Table 11.4.4.3.2-2: Details of the GET request/response on this resource**

	Data type	Cardinality	Description	
	n/a		The request may contain a "Range" HTTP header to obtain single range of bytes from the VNF snapshot package file. This can be used to continue an aborted transmission.  If the Range header is present in the request and the NFVO does not support range requests with a 206 response, it shall return the whole file with a 200 OK response instead as defined below.	
Request body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	Shall be returned when the whole content of the VNF snapshot package file has been read successfully.  The response body shall include a copy of the VNF snapshot package file.  The "Content-Type" HTTP header shall be set according to the type of the file, i.e. to "application/zip" for a VNF snapshot package. The VNF snapshot package format is defined in ETSI GS NFV-SOL 010 [i.8].
Response body	n/a	1		

	n/a	1	206 Partial Content	<p>If the NFVO supports range requests, this response shall be returned when a single consecutive byte range from the content of the VNF snapshot package file has been read successfully according to the request.</p> <p>The response body shall contain the requested part of the VNF snapshot package file.</p> <p>The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [10].</p> <p>The "Content-Type" HTTP header shall be set as defined above for the "200 OK" response.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact the "state" of the VNF snapshot package has a value different from "AVAILABLE".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	0..1	416 Range Not Satisfiable	<p>Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the VNF snapshot package file (e.g. "access after end of file").</p> <p>The response body may contain a ProblemDetails structure.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>

#### 11.4.4.3.3 PUT

The PUT method uploads the content of a VNF package.

This method shall follow the provisions specified in the tables 11.4.4.3.3-1 and 11.4.4.3.3-2 for URI query parameters, request and response data structures, and response codes.

**Table 11.4.4.3.3-1: URI query parameters supported by the PUT method on this resource**

Name	Cardinality	Description
none supported		

Table 11.4.4.3.3-2: Details of the PUT request/response on this resource

	Data type	Cardinality	Description	
Request body	n/a	1	The payload body contains a ZIP file that represents the VNF snapshot package.  The "Content-Type" HTTP header shall be set according to the type of the file, i.e. to "application/zip" for a VNF snapshot package. The VNF snapshot package format is defined in ETSI GS NFV-SOL 010 [i.8].	
	n/a		202 Accepted	Shall be returned when the VNF snapshot package has been accepted for uploading, but the processing has not been completed. It is expected to take some time for processing.  The response body shall be empty. See note.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.  Typically, this is due to the fact the state of the VNF snapshot package resource is other than "CREATED" or "ERROR".  The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.
NOTE: The API consumer can track the uploading progress by reading the status of the "Individual VNF snapshot package" resource using the GET method.				

#### 11.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 11.4.5 Resource: Upload VNF snapshot package from URI task

#### 11.4.5.1 Description

This task resource represents the "Upload VNF snapshot package from URI" operation. The API consumer can use this resource to request the uploading of a VNF snapshot package by providing address information to the NFVO for retrieving the content of the VNF snapshot package.

#### 11.4.5.2 Resource definition

The resource URI is:

**{apiRoot}/vnfsnapshotpkgm/{apiMajorVersion}/vnf\_snapshot\_packages/{vnfSnapshotPkgId}/package\_content/upload\_from\_uri**

This resource shall support the resource URI variables defined in table 11.4.5.2-1.

Table 11.4.5.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 11.1a.
vnfSnapshotPkgId	Identifier of the VNF snapshot package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "id" attribute of the applicable "VnfSnapshotPkgInfo" in the body of the response to requesting the creation of a new "Individual VNF snapshot package" resource or in a response to a GET request querying the "Individual VNF snapshot package" or the "VNF snapshot packages" resource.

### 11.4.5.3 Resource methods

#### 11.4.5.3.1 POST

The POST method provides the information for the NFVO to get the content of a VNF snapshot package.

This method shall follow the provisions specified in the tables 11.4.5.3.1-1 and 11.4.5.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 11.4.5.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 11.4.5.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	UploadVnfSnapshotPkgFromUriRequest	1	The payload body contains the address information based on which the NFVO can obtain the content of the VNF snapshot package.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	Shall be returned when the information about the VNF snapshot package has been received successfully, but the uploading has not been completed. It is expected to take some time for processing.  The response body shall be empty. See note.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.  Typically, this is due to the fact the state of the VNF snapshot package resource is other than "CREATED" or "ERROR".  The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	
NOTE:	The API consumer can track the uploading progress by reading the status of the "Individual VNF snapshot package" resource using the GET method.			

#### 11.4.5.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 11.4.6 Resource: Build VNF snapshot package content task

#### 11.4.6.1 Description

This task resource represents the "Build VNF snapshot package" operation. The API consumer can use this resource to request building the content of a VNF snapshot package to the NFVO by providing information of the VNF snapshot that is to be packaged.

#### 11.4.6.2 Resource definition

The resource URI is:

**{apiRoot}/vnfsnapshotpkgm/{apiMajorVersion}/vnf\_snapshot\_packages/{vnfSnapshotPkgId}/package\_content/build**

This resource shall support the resource URI variables defined in table 11.4.6.2-1.

**Table 11.4.6.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 11.1a.
vnfSnapshotPkgId	Identifier of the VNF snapshot package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "id" attribute of the applicable "VnfSnapshotPkgInfo" in the body of the response to requesting the creation of a new "Individual VNF snapshot package" resource or in a response to a GET request querying the "Individual VNF snapshot package" or the "VNF snapshot packages" resource.

#### 11.4.6.3 Resource methods

##### 11.4.6.3.1 POST

The POST method provides the information for the NFVO to start building the content of a VNF snapshot package.

If the request to build the content of a VNF snapshot package is accepted, the NFVO shall allocate a globally unique identifier to the VNF snapshot package (i.e. "vnfSnapshotPkgUniqueId"). To proceed with the packaging process, the NFVO shall collect the constituent information and artifacts as indicated by the input information in the request:

- "vnfSnapshotInfoId": references the VNF snapshot to be packaged;

- one or more "vnfcSnapshotInfoId" referencing specific VNFC snapshot constituents in the VNF snapshot.

Both identifiers are assumed to be known by the NFVO (as it holds the information of available VNF snapshot of VNF that are part of the NS instance) and can also be retrieved by the NFVO from the VNF from the "id" attribute of the applicable "VnfSnapshot" in the body of the response to a GET request querying the "Individual VNF snapshot" or the "VNF snapshots" resource of the VNF LCM interface as specified in clause 5.4 of ETSI GS NFV-SOL 003 [4].

The API consumer may indicate whether the VNF snapshotted resources on the NFVI shall be included into the VNF snapshot package file or be left on the NFVI as external image artifacts to the VNF snapshot package file. In case of building the VNF snapshot package with snapshotted resources as external image artifacts, the NFVO shall set the value of the "imageUri" and "imagePath" as specified in clause 11.5.3.2.

The NFVO shall determine whether there is any conflict for building the content of the VNF snapshot package, and in particular, determine if the VNF snapshot is complete by checking the "createdAt" information of the VNF snapshot resource.

In addition, the NFVO shall update the "state" attribute of the "VnfSnapshotPkgInfo" during the build process as specified in clause 11.6.

The NFVO shall build the VNF snapshot package as indicated by the input information in the request. For the creation of the VNF snapshot record (VNFSR) that is compiled within the VNF snapshot package (refer to ETSI GS NFV-SOL 010 [i.8]), the NFVO shall include from the array of "vnfcSnapshots" in the "VnfSnapshot" structure only those entries that correspond to the VNFC snapshots that are requested to be packaged as indicated in the data structure of the "BuildVnfSnapshotPkgRequest" included in the payload body of the POST request. In addition, for each VNFC snapshot to be packaged, the NFVO shall package the image artifact associated to the compute snapshot resource as well as all the image artifacts associated to the storage snapshot resources referred in the corresponding "VnfcSnapshotInfo" representing such a VNFC snapshot.

NOTE: The format and provisions for building a VNF snapshot package are specified in ETSI GS NFV-SOL 010 [i.8].

This method shall follow the provisions specified in the tables 11.4.6.3.1-1 and 11.4.6.3.1-2 for URI query parameters, request and response data structures, and response codes.

**Table 11.4.6.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 11.4.6.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	BuildVnfSnapshotPkg Request		1	The payload body contains the information based on which the NFVO can build the content of the VNF snapshot package.
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	Shall be returned when the information for building the VNF snapshot package has been received successfully, but the build has not been completed. It is expected to take some time for processing.  The response body shall be empty. See note.



	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact the state of the VNF snapshot package resource is not "CREATED", or the state is "AVAILABLE", or the VNF snapshot creation is not complete.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.
<p>NOTE: The API consumer can track the building progress by reading the status of the "Individual VNF snapshot package" resource using the GET method.</p>				

#### 11.4.6.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.6.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.6.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 11.4.7 Resource: Extract VNF snapshot package content task

#### 11.4.7.1 Description

This task resource represents the "Extract VNF snapshot package" operation. The API consumer can use this resource to request extracting the content of a VNF snapshot package to the NFVO by providing information of the "Individual VNF snapshot" resource to which the extracted content has to be associated. During this process, the information/content of the "Individual VNF snapshot" resource is populated from the content of the VNF snapshot package.

#### 11.4.7.2 Resource definition

The resource URI is:

**{apiRoot}/vnfsnapshotpkgm/{apiMajorVersion}/vnf\_snapshot\_packages/{vnfSnapshotPkgId}/package\_content/extract**

This resource shall support the resource URI variables defined in table 11.4.7.2-1.

**Table 11.4.7.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 11.1a.
vnfSnapshotPkgId	Identifier of the VNF snapshot package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "id" attribute of the applicable "VnfSnapshotPkgInfo" in the body of the response to requesting the creation of a new "Individual VNF snapshot package" resource or in a response to a GET request querying the "Individual VNF snapshot package" or the "VNF snapshot packages" resource.

### 11.4.7.3 Resource methods

#### 11.4.7.3.1 POST

The POST method provides the information for the NFVO to start extracting the content of a VNF snapshot package.

To proceed with the extraction process, the NFVO shall have been provided either with:

- a valid "Individual VNF snapshot" identifier, indicated by the input parameter "vnfSnapshotInfoId". This identifier corresponds to a known "Individual VNF snapshot" resource of the VNF LCM interface that the NFVO can obtain from the VNFM; or
- a valid "Individual VNF instance" identifier, indicated by the input parameter "vnfInstanceId". This identifier corresponds to a known "Individual VNF instance" resource of the VNF LCM interface that the NFVO can obtain from the VNFM.

The NFVO shall determine whether there is any conflict for extracting the content of the VNF snapshot package, and in particular:

- determine if the "Individual VNF snapshot" resource can be populated with the extracted content of the VNF snapshot package, i.e. the "vnfSnapshot" attribute of the "Individual VNF snapshot" resource is empty;
- determine if the content of the VNF snapshot package is available, and thus the "state" attribute of the "VNF snapshot package" resource has a value equal to "AVAILABLE".

In addition, the NFVO shall update the "state" attribute of the "VnfSnapshotPkgInfo" during the extraction process and change it to "EXTRACTING".

This method shall follow the provisions specified in the tables 11.4.7.3.1-1 and 11.4.7.3.1-2 for URI query parameters, request and response data structures, and response codes.

**Table 11.4.7.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 11.4.7.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	ExtractVnfSnapshotPkgRequest	1	The payload body contains the information based on which the NFVO can extract the content of the VNF snapshot package.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	Shall be returned when the information for extracting the VNF snapshot package has been received successfully, but the extraction has not been completed. It is expected to take some time for processing.  The response body shall be empty. See note.

	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact the state of the VNF snapshot package resource is not "AVAILABLE", or the "Individual VNF snapshot" resource is not empty, or the "Individual VNF instance" is not known to the NFVO.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.
<p>NOTE: The API consumer can track the extraction progress by reading the status of the "Individual VNF snapshot package" resource using the GET method.</p>				

#### 11.4.7.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.7.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.7.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.7.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 11.4.8 Resource: Cancel VNF snapshot package operation task

#### 11.4.8.1 Description

This task resource represents the "Cancel VNF snapshot package operation" operation. The API consumer can use this resource to request cancelling an ongoing operation with the content of a VNF snapshot package to the NFVO.

#### 11.4.8.2 Resource definition

The resource URI is:

**{apiRoot}/vnfsnapshotpkgm/{apiMajorVersion}/vnf\_snapshot\_packages/{vnfSnapshotPkgId}/package\_content/cancel**

This resource shall support the resource URI variables defined in table 11.4.8.2-1.

**Table 11.4.8.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 11.1a.
vnfSnapshotPkgId	Identifier of the VNF snapshot package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "id" attribute of the applicable "VnfSnapshotPkgInfo" in the body of the response to requesting the creation of a new "Individual VNF snapshot package" resource or in a response to a GET request querying the "Individual VNF snapshot package" or the "VNF snapshot packages" resource.

### 11.4.8.3 Resource methods

#### 11.4.8.3.1 POST

The POST method provides the information for the NFVO to cancel the ongoing operation related to the content of a VNF snapshot package.

In case of success of processing the asynchronous request:

- 1) If the request has been processed in "UPLOADING", "BUILDING" or "PROCESSING" state, the "state" attribute in the representation of the "Individual VNF snapshot package" resource shall be changed to "ERROR".
- 2) If the request has been processed in "EXTRACTING" state, the "state" attribute in the representation of the "Individual VNF snapshot package" resource shall be changed to "ERROR\_EXTRACTING".

In both cases, the NFVO shall update the "isCancelPending" attribute in the representation of the "Individual VNF snapshot package" resource according to the provisions in clause 11.5.2.3 to reflect the new status.

Due to race conditions, the processing of the actual operation that is to be cancelled may eventually still succeed, in which case the "state" attribute in the representation of the "Individual VNF snapshot package" resource shall represent the result of that operation, rather than the result of the cancellation.

This method shall follow the provisions specified in the tables 11.4.8.3.1-1 and 11.4.8.3.1-2 for URI query parameters, request and response data structures, and response codes.

**Table 11.4.8.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 11.4.8.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	CancelVnfSnapshotPkgOperationRequest	1	The payload body contains the information based on which the NFVO can proceed with the cancellation.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	Shall be returned when the information for cancelling the ongoing operation with the content of the VNF snapshot package has been received successfully, but the stopping of the operation has not been completed. It is expected to take some time for processing.  The response body shall be empty. See note.

	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact the state of the VNF snapshot package resource is other than "UPLOADING", "BUILDING", "PROCESSING" or "EXTRACTING".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.
<p>NOTE: The API consumer can track the cancellation progress by reading the status of the "Individual VNF snapshot package" resource using the GET method.</p>				

#### 11.4.8.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.8.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.8.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.8.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 11.4.9 Resource: Access configuration for external artifacts

#### 11.4.9.1 Description

This resource represents access configuration information for the download of external artifacts of the VNF snapshot package. The OSS/BSS can use this resource to provide required credentials and address information for the downloading of external artifacts to the NFVO, enabling to retrieve the artifacts. The OSS/BSS can also read the information and can overwrite wrong information to be used for retrying unsuccessful downloads.

#### 11.4.9.2 Resource definition

The resource URI is:

**{apiRoot}/vnfsnapshotpkgm/{apiMajorVersion}/vnf\_snapshot\_packages/{vnfSnapshotPkgId}/ext\_artifacts\_access**

This resource shall support the resource URI variables defined in table 11.4.9.2-1.

Table 11.4.9.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 11.1a.
vnfSnapshotPkgId	Identifier of the VNF snapshot package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "id" attribute of the applicable "VnfSnapshotPkgInfo" in the body of the response to requesting the creation of a new "Individual VNF snapshot package" resource or in a response to a GET request querying the "Individual VNF snapshot package" or the "VNF snapshot packages" resource.

### 11.4.9.3 Resource methods

#### 11.4.9.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.9.3.2 GET

The GET method reads the access configuration information that is used by the NFVO to get the content of external VNF snapshot package artifacts.

This method shall follow the provisions specified in the tables 11.4.9.3.2-1 and 11.4.9.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 11.4.9.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 11.4.9.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	VnfSnapshotPkgExtArtifactsAccessConfig	1	200 OK	Shall be returned when the access configuration information has been read successfully.  If no information has been provided previously, an empty array shall be returned.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 11.4.9.3.3 PUT

The PUT method provides the access configuration information for the NFVO to download the content of external VNF package artifacts.

As precondition for invoking this method, the "Individual VNF snapshot package" resource shall have been created, and the value of "state" attribute in the representation of the "Individual VNF snapshot package" resource shall equal to "CREATED" or "ERROR".

The resource representation in the payload body of the PUT request shall replace the current state of the resource.

This method shall follow the provisions specified in the tables 11.4.9.3.3-1 and 11.4.9.3.3-2 for URI query parameters, request and response data structures, and response codes.

Table 11.4.9.3.3-1: URI query parameters supported by the PUT method on this resource

Name	Cardinality	Description
none supported		

Table 11.4.9.3.3-2: Details of the PUT request/response on this resource

Request body	Data type	Cardinality	Description	
	VnfSnapshotPkgExtArtifactsAccessConfig	1	The payload body contains the access configuration information based on which the NFVO can obtain the external VNF snapshot package artifact files.	
Response body	Data type	Cardinality	Response Codes	Description
	VnfSnapshotPkgExtArtifactsAccessConfig	1	200 OK	<p>Shall be returned when the attribute "state" equals "CREATED" or when the attribute "state" equals "ERROR" and the VNF snapshot package has not been uploaded successfully previously, to indicate that the access configuration information has been stored successfully by the NFVO.</p> <p>The response body shall contain a representation of the state of the resource.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact the "state" attribute of the "Individual VNF snapshot package" resources contains a value different from "CREATED" or "ERROR".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 11.4.9.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.9.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 11.4.10 Resource: Individual VNF snapshot package artifact

#### 11.4.10.1 Description

This resource represents an individual artifact contained in a VNF snapshot package or external to the VNF snapshot package main file. The API consumer can use this resource to fetch the content of the artifact.

### 11.4.10.2 Resource definition

The resource URI is:

**{apiRoot}/vnfsnapshotpkgm/{apiMajorVersion}/vnf\_snapshot\_packages/{vnfSnapshotPkgId}/artifacts/{artifactPath}**

This resource shall support the resource URI variables defined in table 11.4.10.2-1.

**Table 11.4.10.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 11.1a.
vnfSnapshotPkgId	Identifier of the VNF snapshot package. The identifier is allocated by the NFVO. See note 1.
artifactPath	<p>For an artifact contained as a file in the VNF snapshot package, this variable shall contain a sequence of one or path segments representing the path of the artifact within the VNF snapshot package, relative to the root of the package.</p> <p>EXAMPLE:       foo/bar/m%40ster.sh</p> <p>For an external artifact represented as a URI in the VNF snapshot package manifest, this variable shall contain a sequence of one or more path segments as synthesized by the NFVO (see clause 11.5.3.3) representing this artifact. See note 2 and note 3.</p>
NOTE 1:	This identifier can be retrieved from the "id" attribute of the applicable "VnfSnapshotPkgInfo" in the body of the response to requesting the creation of a new "Individual VNF snapshot package" resource or in a response to a GET request querying the "Individual VNF snapshot package" or the "VNF snapshot packages" resource.
NOTE 2:	This identifier can be retrieved from the "artifactPath" attribute of the applicable "additionalArtifacts" entry in the body of the response to a GET request querying the "Individual VNF snapshot package" or the "VNF snapshot packages" resource.
NOTE 3:	Since multiple path segments are allowed to be contained in this variable, the "/" character that separates these segments is not percent-encoded. Each individual segment is percent-encoded if necessary as defined in clause 4.1 of ETSI GS NFV-SOL 013 [16].

### 11.4.10.3 Resource methods

#### 11.4.10.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 11.4.10.3.2 GET

The GET method fetches the content of an artifact within the VNF snapshot package.

If the VNF snapshot package is a result of a building process (refer to "Build VNF snapshot package content task" resource in clause 11.4.6), and the referred "Individual VNF snapshot package artifact" is external to the main VNF snapshot package file, the NFVO shall return a "302 Found" response code referencing the external artifact resource. To fetch the "Individual VNF snapshot package artifact" from URI indicated by the "Location" header, the OSS/BSS will have to be authorized to access the resource referred by the new URI. Furthermore, as indicated by the "302 Found" response code, the OSS/BSS will have to use the original URI of the present "Individual VNF snapshot package artifact" resource in future requests.

This method shall follow the provisions specified in the tables 11.4.10.3.2-1 and 11.4.10.3.2-2 for URI query parameters, request and response data structures, and response codes.

**Table 11.4.10.3.2-1: URI query parameters supported by the GET method on this resource**

Name	Cardinality	Description
none supported		



Table 11.4.10.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			<p>The request may contain a "Range" HTTP header to obtain single range of bytes from the artifact file. This can be used to continue an aborted transmission.</p> <p>If the Range header is present in the request and the NFVO does not support range requests with a 206 response, it shall return the whole file with a 200 OK response instead as defined below.</p>
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the whole content of the artifact file has been read successfully.</p> <p>The response body shall include a copy of the artifact file from the VNF snapshot package. The VNF snapshot package format is defined in ETSI GS NFV-SOL 010 [i.8].</p> <p>The "Content-Type" HTTP header shall be set according to the content type of the artifact file. If the content type cannot be determined, the header shall be set to the value "application/octet-stream".</p>
	n/a	1	206 Partial Content	<p>If the NFVO supports range requests, this response shall be returned when a single consecutive byte range from the content of the artifact file has been read successfully according to the request.</p> <p>The response body shall contain the requested part of the artifact file from the VNF snapshot package. The VNF snapshot package format is defined in ETSI GS NFV-SOL 010 [i.8].</p> <p>The "Content-Type" HTTP header shall be set according to the content type of the artifact file. If the content type cannot be determined, the header shall be set to the value "application/octet-stream".</p> <p>The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [10].</p>
	n/a		302 Found	<p>Shall be returned if the referred "Individual VNF snapshot package artifact" is external to the main VNF snapshot package file created from a build process.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the artifact.</p> <p>The response body shall be empty.</p>
ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact the "state" of the VNF snapshot package has a value different from "AVAILABLE".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>	

	ProblemDetails	0..1	416 Range Not Satisfiable	<p>Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the artifact file (e.g. "access after end of file").</p> <p>The response body may contain a ProblemDetails structure.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

### 11.4.10.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 11.4.10.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 11.4.10.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 11.5 Data model

### 11.5.1 Introduction

This clause defines the request and response data structures of the VNF snapshot package management interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error, and may choose to ignore them.

### 11.5.2 Resource and notification data types

#### 11.5.2.1 Introduction

This clause defines data structures to be used in resource representations and notifications.

#### 11.5.2.2 Type: CreateVnfSnapshotPkgInfoRequest

This type represents the request parameters for creating a new "Individual VNF snapshot package" resource. It shall comply with the provisions defined in table 11.5.2.2-1.

**Table 11.5.2.2-1: Definition of the CreateVnfSnapshotPkgInfoRequest data type**

Attribute name	Data type	Cardinality	Description
name	String	1	Human-readable name of the VNF snapshot package.
userDefinedData	KeyValuePairs	0..1	User defined data for the VNF snapshot package to be built/uploaded.

### 11.5.2.3 Type: VnfSnapshotPkgInfo

This type represents the information of a VNF snapshot package. It shall comply with the provisions defined in table 11.5.2.3-1.

**Table 11.5.2.3-1: Definition of the VnfSnapshotPkgInfo data type**

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the VNF snapshot package information held by the NFVO. This identifier is allocated by the NFVO.
vnfSnapshotPkgUniqueld	Identifier	0..1	Identifier of the VNF snapshot package, which identifies the VNF snapshot package in a globally unique way. It is created during the "build VNF snapshot package operation". Multiples instances of the same VNF snapshot package share the same vnfSnapshotPkgUniqueld. See note.
name	String	1	Human-readable name of the VNF snapshot package.
checksum	Checksum	0..1	Checksum of the stored VNF snapshot package. Hash algorithms applicable to VNF snapshot package are defined in ETSI GS NFV-SOL 010 [i.8]. See note.
createdAt	DateTime	0..1	Timestamp indicating when the VNF snapshot package creation has been completed. See note.
vnfSnapshotId	Identifier	0..1	Identifier of the specific VNF snapshot in the VNF snapshot package. This identifier is allocated by the VNFM during the VNF snapshot creation. See note.
vnfcSnapshotInfolids	IdentifierLocal	0..N	Identifier of information held by the VNFM about specific VNFC snapshot(s) part of the VNF snapshot and contained in the VNF snapshot package. This identifier is allocated by the VNFM during the VNF snapshot creation. See note.
isFullSnapshot	Boolean	1	Value is TRUE in case of a "full" VNF snapshot package, i.e. containing all snapshotted VNFC instances; otherwise the value is FALSE.
vnfdInfo	VnfdInfo	0..1	VNFD of the snapshotted VNF instance that is contained in the stored VNF snapshot package. See note.
vnfsr	VnfSnapshotRecord	0..1	VNF snapshot record with the information as present in the representation of the "Individual VNF snapshot" resource.
vnfcSnapshotImages	VnfcSnapshotImageInfo	0..N	Information about VNF snapshot artifacts that are VNFC snapshot images. Every local and external snapshot image shall be included. No other artifacts shall be included. See note.
additionalArtifacts	SnapshotPkgArtifactInfo	0..N	Information about VNF snapshot artifacts that are not VNFC snapshot images. See note.

Attribute name	Data type	Cardinality	Description
state	Enum (inlined)	1	State of the VNF snapshot package.  Permitted values: <ul style="list-style-type: none"> <li>- CREATED: the VNF snapshot package information has been created.</li> <li>- BUILDING: the VNF snapshot package is being built.</li> <li>- UPLOADING: the VNF snapshot package is being uploaded.</li> <li>- EXTRACTING: the VNF snapshot package's content is being extracted.</li> <li>- AVAILABLE: the VNF snapshot package is available (i.e. build or upload is completed).</li> <li>- PROCESSING: the VNF snapshot package is being processed.</li> <li>- ERROR: failure during the VNF snapshot package building, uploading or processing.</li> <li>- ERROR_EXTRACTING: failure during the VNF snapshot package extraction task.</li> </ul>
isCancelPending	Boolean	1	Indicates if an ongoing operation with the content of the VNF snapshot package is being cancelled. If the value of the "state" attribute is "BUILDING", "UPLOADING", "PROCESSING" or "EXTRACTING" and the operation is being cancelled, this attribute shall be set to true. Otherwise, it shall be set to false.
failureDetails	Structure (inlined)	0..1	Failure details associated to current error state of the VNF snapshot package state.  If "state" is "ERROR" or "ERROR_EXTRACTING", this attribute shall be present unless it has been requested to be excluded via an attribute selector.
>errorType	Enum (inlined)	1	Type of error, when the failure happened (building, upload, processing, extracting).  Permitted values: <ul style="list-style-type: none"> <li>- BUILD_ERROR</li> <li>- UPLOAD_ERROR</li> <li>- PROCESS_ERROR</li> <li>- CANCELLED</li> <li>- EXTRACTION_ERROR</li> </ul>
>details	ProblemDetails	1	Failure details containing error information (such as failed uploading or processing operation, affected artifact, reason for cancellation, etc.). See clause 6.3 of ETSI GS NFV-SOL 013 [16] for the details of "ProblemDetails" structure.
userDefinedData	KeyValuePairs	0..1	User defined data for the VNF snapshot package to be built/uploaded.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>packageContent	Link	1	Link to the "VNF snapshot package content" resource.
NOTE: The attribute shall not be present before the VNF snapshot package content has been uploaded or built. Otherwise, this attribute shall be present unless it has been requested to be excluded per attribute selector.			

### 11.5.2.4 Type: VnfSnapshotPkgInfoModifications

This type represents modifications to the information of a VNF snapshot package. It shall comply with the provisions defined in table 11.5.2.4-1.

**Table 11.5.2.4-1: Definition of the VnfSnapshotPkgInfoModifications data type**

Attribute name	Data type	Cardinality	Description
name	String	0..1	New value of the human-readable name of the VNF snapshot package. See note.
userDefinedData	KeyValuePairs	0..1	User defined data for the VNF snapshot package to be updated. For existing keys, the value is replaced. See note.
state	Enum (inlined)	0..1	New value of the "state" of the VNF snapshot package.  Permitted values: <ul style="list-style-type: none"> <li>- AVAILABLE: to change the "Individual VNF snapshot package" resource state to "AVAILABLE".</li> </ul> Explicit change of state is only permitted from the following states: <ul style="list-style-type: none"> <li>- ERROR_EXTRACTING.</li> </ul> See note.
NOTE: At least one of the three parameters shall be present. If the VNF snapshot package is not uploaded or built, the operation is used only to update existing or add additional user defined data using the userDefinedData attribute.			

### 11.5.2.5 Type: UploadVnfSnapshotPkgFromUriRequest

This type represents the request parameters for uploading the content of a VNF snapshot package. The NFVO can obtain the VNF snapshot package content through the information provided in the request parameters. It shall comply with the provisions defined in table 11.5.2.5-1.

**Table 11.5.2.5-1: Definition of the UploadVnfSnapshotPkgFromUriRequest data type**

Attribute name	Data type	Cardinality	Description
addressInformation	Uri	1	Address information of the VNF snapshot package content. The NFVO can use this address to obtain the VNF snapshot package.
authType	Enum (inlined)	0..1	Defines the type of authentication/authorization for downloading the VNF snapshot package.  Permitted values: <ul style="list-style-type: none"> <li>- BASIC: Only the "username" and "password" attributes shall be present.</li> <li>- OAUTH2_CLIENT_CREDENTIALS: Only the "paramsOAuth2ClientCredentials" attribute shall be present.</li> </ul> This attribute shall not be present if the credentials were provisioned out of band.
username	String	0..1	Username to be used for authentication. Shall be present if user name is needed and has not been provisioned out of band.
password	String	0..1	Password to be used for authentication. Shall be present if password is needed and has not been provisioned out of band.
paramsOAuth2ClientCredentials	Structure (inlined)	0..1	Parameters for authentication/authorization using OAuth 2.0.

Attribute name	Data type	Cardinality	Description
>clientId	String	1	Client identifier to be used in the access token request of the OAuth 2.0 client credentials grant type.
>clientPassword	String	1	Client password to be used in the access token request of the OAuth 2.0 client credentials grant type.
>tokenEndpoint	Uri	1	The token endpoint from which the access token can be obtained.

### 11.5.2.6 Type: BuildVnfSnapshotPkgRequest

This type represents the request parameters for building the content of a VNF snapshot package. The NFVO can obtain the VNF snapshot data through the information provided in the request parameters. It shall comply with the provisions defined in table 11.5.2.6-1.

**Table 11.5.2.6-1: Definition of the BuildVnfSnapshotPkgRequest data type**

Attribute name	Data type	Cardinality	Description
vnfSnapshotInfoId	Identifier	1	Identifier held by the NFVO about the "Individual VNF snapshot" resource with the information managed by the VNFM about a specific VNF snapshot to be packaged into the VNF snapshot package.
vnfcSnapshotsInfoIds	IdentifierLocal	0..N	List of identifiers of information held by the NFVO about VNFC snapshots to be packaged into the VNF snapshot package. Shall be present when it is requested to build a "partial" VNF snapshot package by packaging specific VNFC snapshots of the VNF snapshot. If not present, a "full" VNF snapshot package is requested to be built including all the constituents of the VNF snapshot.
overrideImportForVnfcSnapshotIds	IdentifierLocal	0..N	If present, it indicates the list of VNFC snapshots to which the VNF snapshot-level import snapshot resource policy indicated by the "importSnapshotResource" attribute does not apply and the opposite value shall be considered. See note.
importSnapshotResource	Boolean	0..1	VNF snapshot-level policy indicating whether the NFVO shall import the snapshotted resources from the NFVI (TRUE) or keep in the NFVI the snapshotted resources as external artifacts during the building of the VNF snapshot package file (FALSE). Default value is TRUE. See note.
NOTE: The "overrideImportForVnfcSnapshotIds" provides the list of VNFC snapshots for which the VNF snapshot-level snapshot resource import policy is overridden. See also examples in the present clause.			

Examples on the usage of the BuildVnfSnapshotPkgRequest follow:

**EXAMPLE 1:** Assume a case of a VNF snapshot, identified by "vnfSnapshot\_A", which is composed of three VNFC snapshotted resources identified by "vnfcSnapshotInfoIds" = { "vnfcSnapshot\_1", "vnfcSnapshot\_2", "vnfcSnapshot\_3" }. The request is to perform a "full VNF snapshot packaging" (i.e. all VNFC snapshotted resources which are part of the VNF snapshot), and the client requests the "vnfcSnapshot\_2" not to be imported. The following BuildVnfSnapshotPkgRequest will convey such a request:

```
{
  "vnfSnapshotInfoId" : "vnfSnapshot_A",
  "overrideImportForVnfcSnapshotIds" : [ "vnfcSnapshot_2" ],
  "importSnapshotResource" : TRUE
}
```

END OF EXAMPLE 1

**EXAMPLE 2:** Assume the same case of a VNF snapshot as in example 1, but now the request is to perform a "partial VNF snapshot packaging" (i.e. selected VNFC snapshotted resources which are part of the VNF snapshot) with "vnfcSnapshot\_1" and "vnfcSnapshot\_3", and the global policy is not to import snapshot resources, but only those of "vnfcSnapshot\_1". The following BuildVnfSnapshotPkgRequest will convey such a request:

```
{
  "vnfSnapshotInfoId" : "vnfSnapshot_A",
  "vnfcSnapshotInfoIds" : [ "vnfcSnapshot_1", "vnfcSnapshot_3" ],
  "overrideImportForVnfcSnapshotIds" : [ "vnfcSnapshot_1" ],
  "importSnapshotResource" : FALSE
}
```

END OF EXAMPLE 2

**EXAMPLE 3:** Assume the same case of a VNF snapshot as in example 1, but now the request is to perform a "full VNF snapshot packaging" and the global policy is not to import any of the snapshot resources. The following BuildVnfSnapshotPkgRequest will convey such a request:

```
{
  "vnfSnapshotInfoId" : "vnfSnapshot_A",
  "importSnapshotResource" : FALSE
}
```

END OF EXAMPLE 3

### 11.5.2.7 Type: ExtractVnfSnapshotPkgRequest

This type represents the request parameters for extracting the content of a VNF snapshot package. It shall comply with the provisions defined in table 11.5.2.7-1.

**Table 11.5.2.7-1: Definition of the ExtractVnfSnapshotPkgRequest data type**

Attribute name	Data type	Cardinality	Description
vnfSnapshotInfoId	Identifier	0..1	Identifier held by the NFVO about an "Individual VNF snapshot" resource managed by the VNFM to which the content of the VNF snapshot package will be extracted to. See note.
vnfInstanceInfoId	Identifier	0..1	Identifier of the VNF instance to which the content and extraction of the VNF snapshot package is to be associated. See note.

NOTE: Either the parameter vnfSnapshotInfoId or vnfInstanceInfoId, but not both, shall be provided.

### 11.5.2.8 Type: CancelVnfSnapshotPkgOperationRequest

This type represents the request parameters for cancelling an ongoing operation related to the content of a VNF snapshot package. It shall comply with the provisions defined in table 11.5.2.8-1.

**Table 11.5.2.8-1: Definition of the CancelVnfSnapshotPkgOperationRequest data type**

Attribute name	Data type	Cardinality	Description
cause	String	0..1	Description about the reason for cancelling the operation. If this attribute is present, the NFVO shall add such information into the "failureDetails" of the representation of the "Individual VNF snapshot package" resource.

### 11.5.2.9 Type: VnfSnapshotPkgExtArtifactsAccessConfig

This type represents the access configuration information for downloading external VNF snapshot package artifacts. The NFVO can obtain the external VNF snapshot package artifact file through the information provided in this structure, together with information provided in the manifest. The data structure shall comply with the provisions defined in table 11.5.2.9-1. If the data structure is part of a response body, security-sensitive attributes shall be excluded as specified in table 11.5.2.9-1.

**Table 11.5.2.9-1: Definition of the VnfSnapshotPkgExtArtifactsAccessConfig data type**

Attribute name	Data type	Cardinality	Description
artifact	Structure (inlined)	0..N	Access configuration information for an external artifact.
>artifactUri	Uri	1	The artifactUri attribute as defined in the VNF snapshot package manifest that identifies the external artifact for which the access configuration information is provided, and that provides the default download location for this artifact.
>overrideUri	Uri	0..1	The URI from which the external artifact can be downloaded. If this attribute is present, the NFVO shall download from this address. If this attribute is not present, the NFVO shall download from the address provided in "artifactUri".
>authType	Enum (inlined)	0..1	Defines the type of authentication/authorization for downloading the VNF package.  Permitted values: <ul style="list-style-type: none"> <li>- BASIC: Only the "username" and "password" attributes shall be present.</li> <li>- OAUTH2_CLIENT_CREDENTIALS: Only the "paramsOauth2ClientCredentials" attribute shall be present.</li> </ul> This attribute shall not be present if no credentials are provided for the artifact.
>username	String	0..1	Username to be used for authentication.
>password	String	0..1	Password to be used for authentication. Shall not be present in response bodies.
>paramsOauth2ClientCredentials	Structure (inlined)	0..1	Parameters for authentication/authorization using OAUTH2_CLIENT_CREDENTIALS.
>>clientId	String	1	Client identifier to be used in the access token request of the OAuth 2.0 client credentials grant type.
>>clientPassword	String	1	Client password to be used in the access token request of the OAuth 2.0 client credentials grant type. Shall not be present in response bodies.
>>tokenEndpoint	Uri	1	The token endpoint from which the access token can be obtained.

## 11.5.3 Referenced structured data types

### 11.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but are not resource representations.



### 11.5.3.2 Type: VnfcSnapshotImageInfo

This type represents an artifact contained in or external to a VNF snapshot package which represents a snapshot image. It shall comply with the provisions defined in table 11.5.3.2-1.

**Table 11.5.3.2-1: Definition of the VnfcSnapshotImageInfo data type**

Attribute name	Data type	Cardinality	Description
id	IdentifierLocal	1	<p>Identifier of the VNFC snapshot image.</p> <p>When building the VNF snapshot package, the NFVO shall set the value of this attribute as follows:</p> <ul style="list-style-type: none"> <li>- for an image artifact corresponding to a compute snapshot resource, the value is copied from the "id" attribute of the "VnfcSnapshotInfo";</li> <li>- for an image artifact corresponding to a storage snapshot resource, the value is copied from the "storageResourceId" attribute in the "VnfcSnapshotInfo" of the corresponding storage snapshot resource.</li> </ul> <p>When onboarding an existing VNF snapshot package, the NFVO shall set the value of this attribute as provided in the manifest file in the VNF snapshot package (refer to ETSI GS NFV-SOL 010 [i.8]).</p>
name	String	1	Name of the VNFC snapshot image.
checksum	Checksum	1	Checksum of the snapshot image file. Hash algorithms applicable to VNF snapshot package artifacts are defined in ETSI GS NFV-SOL 010 [i.8].
isEncrypted	Boolean	1	Reflects whether the artifact is encrypted (true) or not (false).
vnfcInstanceId	IdentifierInVnf	1	Identifier of the snapshotted VNFC instance that this snapshot image belongs to.
containerFormat	Enum (inlined)	1	<p>Container format indicates whether the snapshot image is in a file format that also contains metadata about the actual snapshot.</p> <p>Permitted values:</p> <ul style="list-style-type: none"> <li>- AKI: a kernel image format.</li> <li>- AMI: a machine image format.</li> <li>- ARI: a ramdisk image format.</li> <li>- BARE: the image does not have a container or metadata envelope.</li> <li>- DOCKER: docker container format.</li> <li>- OVA: OVF package in a tarfile.</li> <li>- OVF: OVF container format.</li> </ul> <p>See note 1.</p>
diskFormat	Enum (inlined)	1	<p>Disk format of a snapshot image is the format of the underlying disk image.</p> <p>Permitted values:</p> <ul style="list-style-type: none"> <li>- AKI: a kernel image format.</li> <li>- AMI: a machine image format.</li> <li>- ARI: a ramdisk image format.</li> <li>- ISO: an archive format for the data contents of an optical disc, such as CD-ROM.</li> <li>- QCOW2: a common disk image format, which can expand dynamically and supports copy on write.</li> <li>- RAW: an unstructured disk image format.</li> <li>- VDI: a common disk image format.</li> <li>- VHD: a common disk image format.</li> <li>- VHDX: enhanced version of VHD format.</li> <li>- VMDK: a common disk image format.</li> </ul> <p>See note 2.</p>

Attribute name	Data type	Cardinality	Description
createdAt	DateTime	1	Timestamp indicating when the VNFC snapshot image was created.
minDisk	UnsignedInt	1	The minimal disk for this VNFC snapshot image (in bytes).
minRam	UnsignedInt	1	The minimal RAM for this VNFC snapshot image (in bytes).
size	UnsignedInt	1	Size of this VNFC snapshot image (in bytes).
userMetadata	KeyValuePairs	0..1	User-defined metadata.
imagePath	String	0..1	<p>Path which identifies the image artifact and also allows to access a copy of the image artifact.</p> <p>For an image artifact contained as a file in the VNF snapshot package, this attribute shall be present, and the value of this attribute shall start with the name of the first segment in the path in the package, i.e. it shall not be prefixed by path separator characters such as "." and "/".</p> <p>EXAMPLE:       foo/bar/m%40ster.vhd</p> <p>For an external image artifact represented as a URI in the manifest file, this attribute shall be present if the artifact has been downloaded by the NFVO or the artifact has been processed after building the VNF snapshot package and shall be absent otherwise. If present, it shall contain the artifactPath under which the image artifact can be obtained using the "Individual artifact in a VNF snapshot package" resource defined in clause 11.4.10. It is the responsibility of the NFVO to synthesize this path in a manner that avoids any collision of the synthesized artifact path with the paths and names of artifacts included in the snapshot package.</p>
imageUri	Uri	0..1	<p>URI of the image artifact as defined in the VNF snapshot package manifest. Shall be present if the image artifact is external to the snapshot package and shall be absent otherwise.</p> <p>EXAMPLE:       https://example.com/m%40ster.vhd</p>
NOTE 1: The list of permitted values was taken from "Container formats" in [i.2].			
NOTE 2: The list of permitted values was adapted from "Disk formats" in [i.2].			

### 11.5.3.3 Type: SnapshotPkgArtifactInfo

This type represents an artifact other than a software image which is contained in a VNF snapshot package. It shall comply with the provisions defined in table 11.5.3.3-1.

Table 11.5.3.3-1: Definition of the SnapshotPkgArtifactInfo data type

Attribute name	Data type	Cardinality	Description
artifactPath	String	0..1	<p>Path which identifies the artifact and also allows to access a copy of the artifact.</p> <p>For an artifact contained as a file in the VNF snapshot package, this attribute shall be present, and the value of this attribute shall start with the name of the first segment in the path in the package, i.e. it shall not be prefixed by path separator characters such as "." and "/".</p> <p>EXAMPLE:       foo/bar/m%40ster.sh</p> <p>For an external artifact represented as a URI in the manifest file, this attribute shall be present if the artifact has been downloaded by the NFVO or the artifact has been processed after building the VNF snapshot package and shall be absent otherwise. If present, it shall contain the artifactPath under which the artifact can be obtained using the "Individual artifact in a VNF snapshot package" resource defined in clause 11.4.10. It is the responsibility of the NFVO to synthesize this path in a manner that avoids any collision of the synthesized artifact path with the paths and names of artifacts included in the snapshot package.</p>
artifactUri	Uri	0..1	<p>URI of the artifact as defined in the VNF snapshot package manifest. Shall be present if the artifact is external to the snapshot package and shall be absent otherwise.</p> <p>EXAMPLE:       https://example.com/m%40ster.sh</p>
checksum	Checksum	1	Checksum of the artifact file. Hash algorithms applicable to VNF snapshot package artifacts are defined in ETSI GS NFV-SOL 010 [i.8].
isEncrypted	Boolean	1	Reflects whether the artifact is encrypted (true) or not (false).
metadata	KeyValuePairs	0..1	The metadata of the artifact that are available in the VNF package, such as Content type, size, creation date, etc.

#### 11.5.3.4 Type: VnfdInfo

This type represents the VNFD which is contained in a VNF snapshot package. It shall comply with provisions defined in table 11.5.3.4-1.

Table 11.5.3.4-1: Definition of the VnfdInfo data type

Attribute name	Data type	Cardinality	Description
vnfdId	Identifier	1	VNFD identifier of the snapshot VNF instance.
vnfdPath	String	1	<p>Path which allows to access a copy of the VNFD. The VNFD is implemented as a collection of one or more files, and the path refers to the ZIP archive file embedding these files. The VNF snapshot package format is defined in ETSI GS NFV-SOL 010 [i.8].</p> <p>The value of this attribute shall start with the name of the first segment of the path in the package, i.e. it shall not be prefixed by path separator characters such as "." and "/".</p> <p>EXAMPLE:       foo/bar/m@ster</p>
checksum	Checksum	1	Checksum of the VNFD archive file. Hash algorithms applicable to VNF snapshot package artifacts are defined in ETSI GS NFV-SOL 010 [i.8].
isEncrypted	Boolean	1	Reflects whether the VNFD archive file is encrypted (true) or not (false).

### 11.5.3.5 Type: VnfSnapshotRecord

This type represents the VNF snapshot record which is contained in a VNF snapshot package. It shall comply with provisions defined in table 11.5.3.5-1.

**Table 11.5.3.5-1: Definition of the VnfSnapshotRecord data type**

Attribute name	Data type	Cardinality	Description
recordPath	String	1	Path which identifies the VNF snapshot record and allows to access a copy of the VNF snapshot record for the extraction.  The value of this attribute shall start with the name of the first segment of the path in the package, i.e. it shall not be prefixed by path separator characters such as "." and "/".  EXAMPLE:       foo/bar/m@ster
checksum	Checksum	1	Checksum of the VNF snapshot record file. Hash algorithms applicable to VNF snapshot package artifacts are defined in ETSI GS NFV-SOL 010 [i.8].
isEncrypted	Boolean	1	Reflects whether the VNF snapshot record file is encrypted (true) or not (false).

## 11.5.4 Referenced simple data types and enumerations

No particular simple data types and enumerations are defined for this interface, in addition to those defined in clause 4.4.

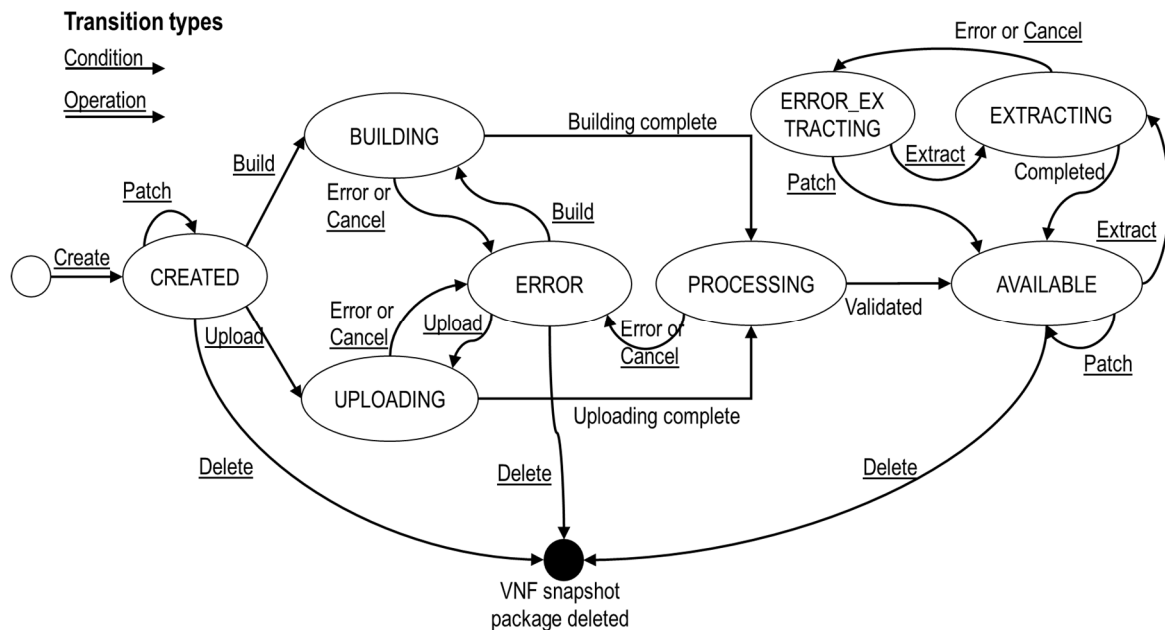
## 11.6 VNF snapshot package state model and error handling

### 11.6.1 Introduction

This clause describes the state model of the VNF snapshot package. In addition, during the VNF snapshot package management procedures, failures can occur. This clause describes also how to handle errors during the VNF snapshot package management operations.

### 11.6.2 States and state transitions

The API producer shall support the VNF snapshot package states and state transition specified below. The states and state transitions are also illustrated in figure 11.6.2-1.



**Figure 11.6.2-1: States and state transition of the VNF snapshot package**

**CREATED:** The "Individual VNF snapshot package" resource has been created. This state has the following characteristics:

- This is the initial state after the successful completion of the POST request to the "VNF snapshot packages" resource.
- The API consumer may send a PATCH request to the "Individual VNF snapshot package" resource to update information of the VNF snapshot package.
- If the API consumer sends a POST request to the "Build VNF snapshot package content task" resource, the state of the "Individual VNF snapshot package" resource shall transit to the "BUILDING" state.
- If the API consumer sends a POST request to the "Upload VNF snapshot package from URI task" resource or a PUT request to the "Individual VNF snapshot package content" resource, the state of the "Individual VNF snapshot package" resource shall transit to the "UPLOADING" state.
- A successful handling of the DELETE request to the "Individual VNF snapshot package" will delete the "Individual VNF snapshot package" resource. The provisions for the deletion of the "Individual VNF snapshot package" are specified in clause 11.4.3.3.5.

**UPLOADING:** The content of the VNF snapshot package is being uploaded, and it is not completed. This state has the following characteristics:

- This state shall block the request to delete the "Individual VNF snapshot package" resource.
- Upon successful completion of the uploading of the VNF snapshot package, the state of the "Individual VNF snapshot package" resource shall transit to the "PROCESSING" state.
- If a failure occurs during the uploading state from which the API producer knows that it cannot be fixed, the state of the "Individual VNF snapshot package" resource shall transit to the "ERROR" state.
- If the API consumer sends a POST request to the "Cancel VNF snapshot package operation task" resource, upon completion of the cancellation the state of the "Individual VNF snapshot package" resource shall transit to the "ERROR" state.

**BUILDING:** The content of the VNF snapshot package is being built from a VNF snapshot. This state has the following characteristics:

- This state shall block the request to delete the "Individual VNF snapshot package" resource.

- Upon successful building of the VNF snapshot package, the state of the "Individual VNF snapshot package" resource shall transit to the "PROCESSING" state.
- If a failure occurs during the building process from which the API producer knows that it cannot be fixed, the state of the "Individual VNF snapshot package" resource shall transit to the "ERROR\_BUILDING" state.
- If the API consumer sends a POST request to the "Cancel VNF snapshot package operation task" resource, upon completion of the cancellation the state of the "Individual VNF snapshot package" resource shall transit to the "ERROR" state.

**PROCESSING:** The content of the VNF snapshot package is being processed and validated after its uploading or building. This state has the following characteristics:

- This state shall block the request to delete the "Individual VNF snapshot package" resource.
- Upon successful processing and validation of the VNF snapshot package, the state of the "Individual VNF snapshot package" resource shall transit to the "AVAILABLE" state.
- If a failure occurs during the processing of the VNF snapshot package, or errors are found during the validation of the VNF snapshot package, the state of the "Individual VNF snapshot package" resource shall transit to the "ERROR" state.
- If the API consumer sends a POST request to the "Cancel VNF snapshot package operation task" resource, upon completion of the cancellation the state of the "Individual VNF snapshot package" resource shall transit to the "ERROR" state.

**ERROR:** Some failure happened during the uploading, building or processing of the VNF snapshot package, or underlying operation has been cancelled. This state has the following characteristics:

- If the previous state before transiting to "ERROR" was "UPLOADING" and the API consumer sends a POST request to the "Upload VNF snapshot package from URI task" resource or a PUT request to the "VNF snapshot package content" resource, the state of the "Individual VNF snapshot package" resource shall transit to the "UPLOADING" state.
- If the previous state before transiting to "ERROR" was "BUILDING" and the API consumer sends a POST request to the "Build VNF snapshot package content task" resource, the state of the "Individual VNF snapshot package" resource shall transit to the "BUILDING" state.
- A successful handling of the DELETE request to the "Individual VNF snapshot package" will delete the "Individual VNF snapshot package" resource. The provisions for the deletion of the "Individual VNF snapshot package" are specified in clause 11.4.3.3.5.

**AVAILABLE:** The content of the VNF snapshot package is valid and is available for its use. This state has the following characteristics:

- The API consumer may send a PATCH request to the "Individual VNF snapshot package" resource to update information of the VNF snapshot package.
- If the API consumer sends a POST request to the "Extract VNF snapshot package content task" resource, the state of the "Individual VNF snapshot package" resource shall transit to the "EXTRACTING" state.
- A successful handling of the DELETE request to the "Individual VNF snapshot package" will delete the "Individual VNF snapshot package" resource. The provisions for the deletion of the "Individual VNF snapshot package" are specified in clause 11.4.3.3.5.

**EXTRACTING:** The content of the VNF snapshot package is being extracted into a VNF snapshot resource. This state has the following characteristics:

- This state shall block other extraction requests from being executed on the same VNF snapshot package.
- This state shall block the request to delete the "Individual VNF snapshot package" resource.
- Upon successful completion of the extraction of the VNF snapshot package content, the state of the "Individual VNF snapshot package" resource shall transit to the "AVAILABLE" state.

- If a failure occurs during the extraction process from which the API producer knows that it cannot be fixed, the state of the "Individual VNF snapshot package" resource shall transit to the "ERROR\_EXTRACTING" state.
- If the API consumer sends a POST request to the "Cancel VNF snapshot package operation task" resource, upon completion of the cancellation the state of the "Individual VNF snapshot package" resource shall transit to the "ERROR\_EXTRACTING" state.

**ERROR\_EXTRACTING:** Error happened during the extraction process of the VNF snapshot package content or the extracting operation has been cancelled. This state has the following characteristics:

- This state shall block the request to delete the "Individual VNF snapshot package" resource.
- The API consumer may send a PATCH request to the "Individual VNF snapshot package" resource to update information or status of the VNF snapshot package.
- If the API consumer sends a POST request to the "Extract VNF snapshot package content task" resource, the state of the "Individual VNF snapshot package" resource shall transit to the "EXTRACTING" state.
- If the API consumer sends a PATCH request to the "Individual VNF snapshot package" resource with a change of the state to "AVAILABLE" in the payload body, the state of the "Individual VNF snapshot package" resource shall transit to the "AVAILABLE" state.

### 11.6.3 Handling of errors

#### 11.6.3.1 Introduction

To handle the error, the API consumer has the following options:

- Delete the VNF snapshot package resource and create a new one.
- Retry the operation that has failed with or without modifying input information.

The following clauses specify the error handling applicable to different states of the VNF snapshot package.

#### 11.6.3.2 Failure during uploading of the VNF snapshot package

Upon a failure during the uploading, the state of the "Individual VNF snapshot package" resource will transition to the "ERROR" state. After entering the error state:

- the API consumer may discard the uploading process by issuing a DELETE request to the "Individual VNF snapshot package" resource; or
- by reading failure details, the API consumer can determine that the failure might have originated from wrong access/URI information and a retry might be successful, the API consumer may send a POST request to the "Upload VNF snapshot package from URI task" resource with new access/URI information or a PUT request to the "VNF snapshot package content" resource with a new payload containing the VNF snapshot package.

In addition, the API consumer may at any time during the "UPLOADING" state send a POST request to the "Cancel VNF snapshot package operation task" resource in order to cancel the uploading process. For instance, due to abnormal long running execution of the upload, the API consumer might suspect that some silent error is occurring and decides to abort the process.

#### 11.6.3.3 Failure during building of the VNF snapshot package

Upon a failure during the building of the VNF snapshot package, the state of the "Individual VNF snapshot package" resource will transition to the "ERROR" state. After entering the error state:

- the API consumer may discard the building process by issuing a DELETE request to the "Individual VNF snapshot package" resource;

- by reading failure details, the API consumer can determine that the failure might have originated due to errors in the source VNF snapshot and a retry might be successful, the API consumer may send a POST request to the "Build VNF snapshot package content task" resource with new VNF snapshot or VNFC snapshot information.

In addition, the API consumer may at any time during the "BUILDING" state send a POST request to the "Cancel VNF snapshot package operation task" resource in order to cancel the building process. For instance, due to abnormal long running execution of the building, the API consumer might suspect that some silent error is occurring and decides to abort the process.

#### 11.6.3.4 Failure during processing of the VNF snapshot package

Upon a failure during the processing of the VNF snapshot package, the state of the "Individual VNF snapshot package" resource will transition to the "ERROR" state. After entering the error state:

- the API consumer may discard the processing of the VNF snapshot package by issuing a DELETE request to the "Individual VNF snapshot package" resource; or
- in addition, the API consumer may at any time during the "PROCESSING" state send a POST request to the "Cancel VNF snapshot package operation task" resource in order to cancel the processing process. For instance, due to abnormal long running execution of the processing, the API consumer might suspect that some silent error is occurring and decides to abort the process.

#### 11.6.3.5 Failure during extraction of the VNF snapshot package

Upon a failure during the extraction of the VNF snapshot package content, the state of the "Individual VNF snapshot package" resource will transition to the "ERROR\_EXTRACTING" state. After entering the error state:

- by reading failure details, the API consumer can determine that the specific extraction error is unresolvable, but the VNF snapshot package is still valid, the API consumer may send a PATCH request to the "Individual VNF snapshot package" resource to change the state to "AVAILABLE";
- by reading failure details, the API consumer can determine that the failure might have originated due to errors in the target VNF snapshot information resource and a retry might be successful, the API consumer may send a POST request to the "Extract VNF snapshot package content task" resource with new VNF snapshot resource information.

---

## 12 NS LCM Coordination interface

### 12.1 Description

This interface allows an NFVO to request the OSS/BSS to perform coordinative actions during LCM operations. Further, this interface allows API version information retrieval.

The operation provided through this interface is as follows:

- Coordinate LCM Operation.
- The support of the NS LCM Coordination interface by the OSS/BSS is optional and is declared in the NSD.



NOTE: This interface follows the information model and requirements defined in ETSI GS NFV-IFA 013 [3] and supports a synchronous and an asynchronous mode as described in clause 12.3.1 "Flow of LCM coordination". The action "CONTINUE\_AFTER\_DELAY" is not signalled explicitly on the interface, as it can be realized by the API producer delaying returning the coordination result when using the asynchronous mode. When using the synchronous mode, it maps to "RETRY\_AFTER\_DELAY" and a subsequent repetition of the coordination. The action "RETRY\_AFTER\_DELAY" is also not signalled explicitly on the interface, as it can be realized by the API producer by waiting for the delay time to pass and then retrying the operation directly when using the asynchronous mode. When using the synchronous mode, the action "RETRY\_AFTER\_DELAY" is realized in a RESTful way by the API producer responding with a 503 error response, and the API consumer resending the request after the delay that was signalled in the 503 response has passed.

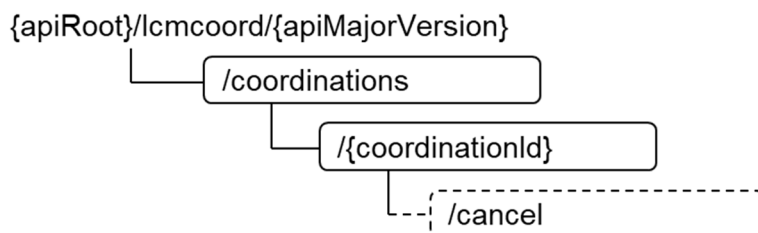
## 12.1a API version

For the NS LCM coordination interface version as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 10, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [16] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v1".

## 12.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [16]. The string "lcmcoord" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 12.2-1 shows the overall resource URI structure defined for the NS LCM coordination interface.



**Figure 12.2-1: Resource URI structure of the NS LCM Coordination Interface**

Table 12.2-1 lists the individual resources defined, and the applicable HTTP methods.

If the OSS/BSS supports the NS LCM coordination interface, the OSS/BSS shall support responding to requests for all HTTP methods on the resources in table 12.2-1 that are marked as "M" (mandatory) in the "Cat" column. The OSS/BSS shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [16].

**Table 12.2-1: Resources and methods overview of the NS LCM Coordination interface**

Resource name	Resource URI	HTTP Method	Cat	Meaning
Coordinations	/ coordinations	POST	M	Request a coordination action.
Individual Coordination action	/ coordinations/{coordinationId}	GET	M	Read the result of a coordination action.
Cancel coordination action task	/ coordinations/{coordinationId}/cancel	POST	M	Cancel an ongoing coordination action.

## 12.3 Sequence diagrams (informative)

### 12.3.1 Flow of LCM coordination

This clause describes the sequence for an LCM coordination between the NFVO and OSS/BSS.

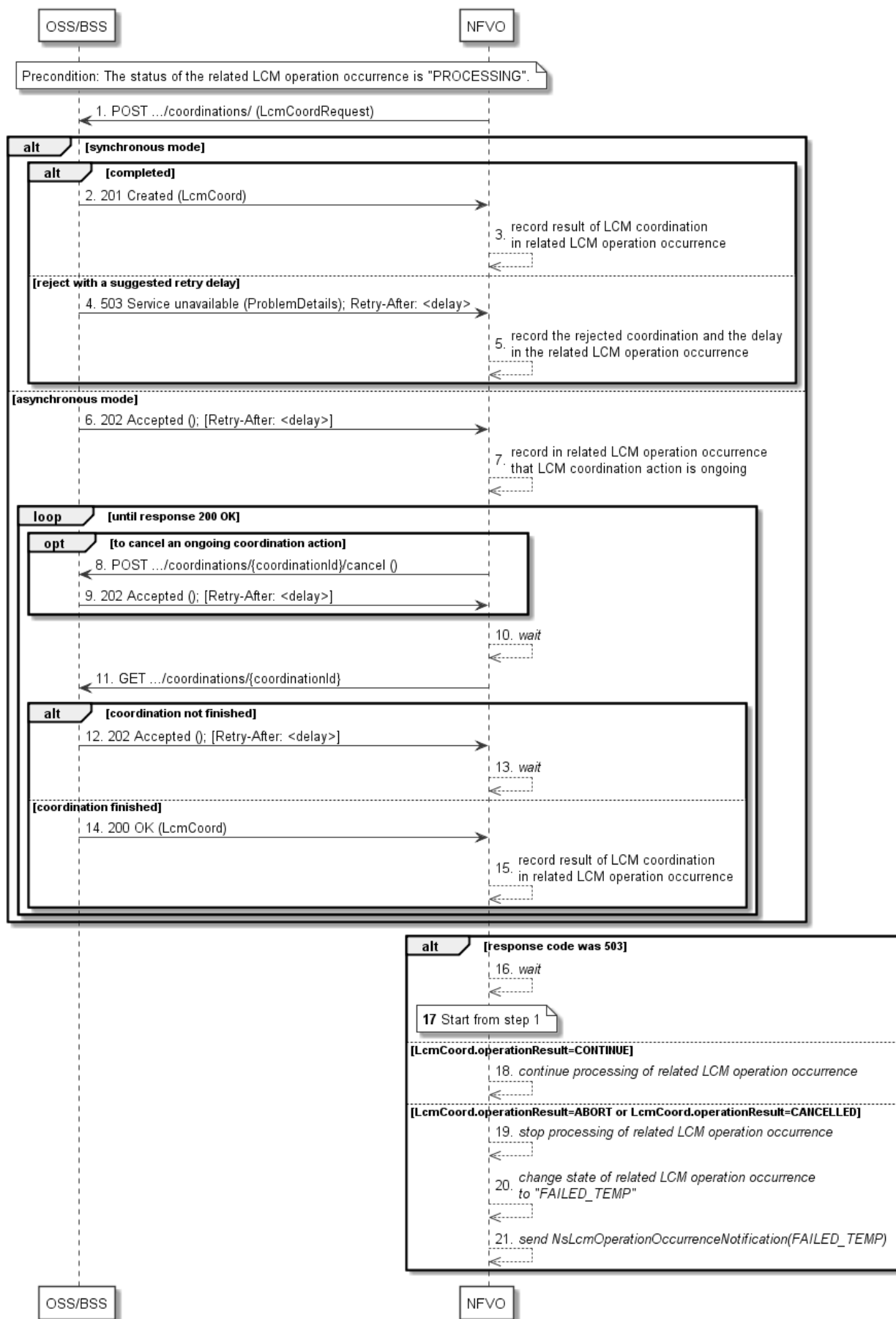


Figure 12.3.1-1: Flow of LCM coordination

An LCM coordination occurs always in the context of an LCM operation occurrence which is represented as a resource in the NS lifecycle management API (see clause 6.4.10). The time intervals mentioned in the steps below are determined by means outside the scope of the present document, e.g. policy, unless they are signalled by the API producer in the "Retry-After" header. The API producer can choose whether to use the synchronous or asynchronous mode.

**Precondition:** The status of the related LCM operation occurrence resource is "PROCESSING".

An LCM coordination, as illustrated in figure 12.3.1-1, consists of the following steps:

- 1) The NFVO sends a POST request to the "Coordinations" resource with a "LcmCoordRequest" data structure in the body.

Synchronous mode:

In case the coordination was completed, the following steps are executed:

- 2) The OSS/BSS returns to the NFVO a "201 Created" response with an "LcmCoord" data structure in the body and a "Location" HTTP header that indicates the URI of the "Individual coordination action" resource that has been created as the result of the finished coordination procedure.
- 3) The NFVO records in the related LCM operation occurrence resource that the result of the LCM coordination action.

In case the coordination request is rejected with a suggestion to retry the request after a delay:

- 4) The OSS/BSS returns to the NFVO a "503 Service unavailable" response with a "ProblemDetails" data structure in the body and a "Retry-After" HTTP header that indicates the length of a delay after which a retry of the coordination is suggested.
- 5) The NFVO records in the related LCM operation occurrence resource that the LCM coordination action was rejected with a suggested delay for a retry. After the delay interval has passed, the NFVO starts again at step 1).

Asynchronous mode:

- 6) The OSS/BSS returns to the NFVO a "202 Accepted" response with an empty body and a "Location" HTTP header that indicates the URI of the "Individual coordination action" resource that will be created once the coordination procedure at the EM/VNF is finished and an optional "Retry-After" header that indicates a delay after which the resource is suggested to be read with a GET request.
- 7) The NFVO records in the related LCM operation occurrence resource that an LCM coordination action is ongoing and stores related information.

The following steps 8) to 15) are executed in a loop until the coordination has produced a response or an implementation-specific timeout has occurred.

- 8) Optionally, to cancel an ongoing LCM coordination action, the NFVO sends a POST request to the "Cancel coordination action task" resource with an empty request body.
- 9) In that case, the OSS/BSS starts the cancellation and returns to the NFVO a "202 Accepted" response with an empty payload body and an optional "Retry-After" header that indicates a delay after which the resource is suggested to be read with a GET request.
- 10) The NFVO waits for a certain time interval (as indicated in the Retry-After header of the previous 202 Response if signalled, or determined by other means otherwise) before the next iteration of the loop.
- 11) The NFVO polls the status of the coordination by sending a GET request to the OSS/BSS, using the URI that was returned in step 4) in the "Location" header.
- 12) If the coordination action is ongoing at the OSS/BSS and consequently the "Individual coordination action" resource is still in the process of being created, the OSS/BSS returns a "202 Accepted" response with an empty body and an optional "Retry-After" header that indicates a delay after which the resource is suggested to be read with a GET request.

- 13) In that case, the NFVO waits for a certain time interval (as indicated in the Retry-After header of the previous 202 response if signalled, or determined by other means otherwise) before the next iteration of the loop.
- 14) If the result of the coordination is available, the OSS/BSS returns a "200 OK" response with an "LcmCoord" data structure in the body.
- 15) In this case, the NFVO records in the related LCM operation occurrence resource that the result of the LCM coordination action.

After finish of the loop in case of asynchronous mode or after completion of step 3) or 5) in case of synchronous mode:

- 16) If the previous response was 503 Service unavailable, the NFVO waits for the time interval indicated in the Retry-After header.
- 17) In that case, after waiting, the NFVO starts the procedure again from step 1), passing the same parameters as in the previous invocation.
- 18) If the coordinationResult resulting from the coordination was "CONTINUE", the NFVO continues the processing of the LCM operation.
- 19) If the coordinationResult resulting from the coordination was "ABORT" or CANCELLED, the NFVO stops the processing of the LCM operation.
- 20) In that case, the NFVO sets the state of the related LCM operation occurrence resource to "FAILED\_TEMP".
- 21) Further in that case, the NFVO notifies the subscribers of the state change.

**Postcondition:** The state of the related LCM operation occurrence resource is either "PROCESSING" or "FAILED\_TEMP" depending on the coordination result. If the state has changed, NsLcmOperationOccurrenceNotifications have been sent to subscribed entities.

## 12.4 Resources

### 12.4.1 Introduction

This clause defines all the resources and methods provided by the NS LCM coordination interface.

#### 12.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [16] are part of the NS LCM coordination operation interface.

### 12.4.2 Resource: Coordinations

#### 12.4.2.1 Description

This resource represents LCM coordination actions. The NFVO can use this resource to request the coordination of an LCM operation occurrence with a management operation executed in the API producer. The coordination can be required at various stages of the LCM operation.

#### 12.4.2.2 Resource definition

The resource URI is:

**{apiRoot}/lcmcoord/{apiMajorVersion}/coordinations**

This resource shall support the resource URI variables defined in table 12.4.2.2-1.

Table 12.4.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 12.1a.

### 12.4.2.3 Resource methods

#### 12.4.2.3.1 POST

The POST method requests the coordination of an LCM operation occurrence with a management operation executed in the API producer.

This method shall follow the provisions specified in the tables 12.4.2.3.1-1 and 12.4.2.3.1-2 for URI query parameters, request and response data structures, and response codes.

The API producer chooses whether the request is processed asynchronously which shall be indicated by responding with "201 Created" or "503 Service Unavailable", or synchronously which shall be indicated by responding with "202 Accepted". As the result of successfully finalizing the operation, a new "Individual coordination action" resource shall be created.

If a "Retry-After" delay value is signalled in a 503 response, the NFVO shall send the coordination request again with the same parameters after the signalled time interval has passed, unless the NFVO is no longer willing to retry the coordination in which case the LCM operation occurrence state shall be changed to "FAILED\_TEMP".

If a "Retry-After" delay value is signalled in a 202 response, the NFVO should not send the subsequent GET request before the signalled time interval has passed.

Table 12.4.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Remarks
none supported		

Table 12.4.2.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	LcmCoordRequest	1	Parameters for the coordination action as defined in clause 12.5.2.2.	
Response body	Data type	Cardinality	Response Codes	Remarks
	LcmCoord	1	201 Created	<p>Shall be returned to indicate a finished coordination action when the API producer has chosen the synchronous mode, which may be selected for coordination actions that finish within the time frame in which an HTTP response is expected.</p> <p>The response body shall contain an LcmCoord data structure that represents the result of the coordination action.</p> <p>The HTTP response shall include a "Location" HTTP header that indicates the URI of the "Individual coordination action" resource that has been created as the result of the finished coordination procedure.</p>
	n/a		202 Accepted	<p>Shall be returned when the API producer has chosen the asynchronous mode and the request has been accepted for processing.</p> <p>The response body shall be empty.</p>

				The HTTP response shall include a "Location" HTTP header that indicates the URI of the "Individual coordination action" resource that will be created once the coordination operation has finished successfully. Further, the HTTP response may include a "Retry-After" HTTP header that indicates the time to wait before sending the next GET request to the "individual coordination" resource indicated in the "Location" header. If the header is provided, the NFVO shall record the signalled delay value in the "delay" attribute of the applicable entry in the "lcmCoordinations" array in the "NsLcmOpOcc" structure.
	ProblemDetails	1	403 Forbidden	<p>Shall be returned upon the following error: The starting of the coordination operation has been rejected.</p> <p>No "Individual coordination action" resource shall be created.</p> <p>A ProblemDetails structure shall be included in the response to provide more details about the rejection in the "details" attribute.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the "Coordinations" resource.</p> <p>Typically, this is due to the fact that no more coordination actions can be executed currently e.g. because too many of them, or conflicting ones, are in progress.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	503 Service unavailable	<p>Shall be returned upon the following error: The API producer has chosen the synchronous mode and cannot perform the requested coordination currently, but expects to be able to perform it sometime in the future.</p> <p>No "individual coordination action" resource shall be created.</p> <p>A ProblemDetails structure shall be included in the response to provide more details about the rejection in the "details" attribute.</p> <p>The HTTP response shall include a "Retry-After" HTTP header that indicates the delay after which it is suggested to repeat the coordination request with the same set of parameters. The NFVO shall record the signalled delay value in the "delay" attribute of the applicable entry in the "rejectedLcmCoordinations" array in the "NsLcmOpOcc" structure.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 12.4.2.3.2 GET

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 12.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 12.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

### 12.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 12.4.3 Resource: Individual coordination action

### 12.4.3.1 Description

This resource represents an individual coordination action. The NFVO can use this resource to determine whether the coordination action is ongoing or finished, and to read the result of the coordination.

The coordination result includes an indication whether to continue the execution of the related LCM operation occurrence and may include additional information. By delaying the response, the API producer can delay the further execution of the LCM operation, as the NFVO will wait for the API producer to provide a response.

### 12.4.3.2 Resource definition

The resource URI is:

**{apiRoot}/lcmcoord/{apiMajorVersion}/ coordinations/{coordinationId}**

The base resource URI variables for this resource are defined in table 12.4.3.2-1.

**Table 12.4.3.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 12.1a.
coordinationId	Identifier of the LCM coordination. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request to the "Coordinations" resource.

### 12.4.3.3 Resource methods

#### 12.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 12.4.3.3.2 GET

The GET method reads a coordination result.

This method shall follow the provisions specified in the tables 12.4.3.3.2-1 and 12.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.



Table 12.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Remarks
none supported		

Table 12.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response codes	Remarks
	LcmCoord	1	200 OK	<p>Shall be returned when the coordination is finished and the coordination result has been read successfully.</p> <p>A representation of the "Individual coordination action" resource shall be returned in the response body.</p>
	n/a		202 Accepted	<p>Shall be returned when the management operation with which coordination is requested is still ongoing or in the process of being cancelled, i.e. no coordination result is available yet.</p> <p>The response body shall be empty.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 12.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 12.4.3.3.4 PATCH

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 12.4.3.3.5 DELETE

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

It is determined by means outside the scope of the present document, such as configuration or policy, how long an "Individual coordination action" resource is available after the coordination action has finished or was cancelled.

### 12.4.4 Resource: Cancel coordination action task

#### 12.4.4.1 Description

This task resource represents the "cancel" operation related to an individual coordination action. The NFVO can use this resource to request the cancellation of an ongoing individual coordination action.

#### 12.4.4.2 Resource definition

The resource URI is:

**{apiRoot}/lcmcoord/{apiMajorVersion}/ coordinations/cancel**

This resource shall support the resource URI variables defined in table 12.4.4.2-1.

**Table 12.4.4.2-1: Resource URI variables for this resource**

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 12.1a.
coordinationId	Identifier of the LCM coordination. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request to the "Coordinations" resource.

### 12.4.4.3 Resource methods

#### 12.4.4.3.1 POST

The POST method initiates the cancellation of an ongoing coordination action.

This method shall follow the provisions specified in the tables 12.4.4.3.1-1 and 12.4.4.3.1-2 for URI query parameters, request and response data structures, and response codes.

**Table 12.4.4.3.1-1: URI query parameters supported by the POST method on this resource**

Name	Cardinality	Description
none supported		

**Table 12.4.4.3.1-2: Details of the POST request/response on this resource**

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the cancellation request has been accepted for processing.</p> <p>The response shall have an empty payload body.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the "Individual coordination action" resource.</p> <p>Typically, this is due to the fact that the coordination action has finished processing.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]		4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

#### 12.4.4.3.2 GET

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 12.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 12.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

#### 12.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

## 12.5 Data model

### 12.5.1 Introduction

This clause defines the request and response data structures of the NS LCM coordination interface.

If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error and may choose to ignore them.

Coordination actions are specific to the NS and the LCM operation.

### 12.5.2 Resource and notification data types

#### 12.5.2.1 Introduction

This clause defines the data structures to be used in resource representations and notifications.

#### 12.5.2.2 Type: LcmCoordRequest

This type represents an LCM coordination request. It shall comply with the provisions defined in table 12.5.2.2-1.

**Table 12.5.2.2-1: Definition of the LcmCoordRequest data type**

Attribute name	Data type	Cardinality	Description
nsInstancelId	Identifier	1	Identifier of the NS instance which this coordination request is related to.
nsLcmOpOccId	Identifier	1	The identifier of the NS lifecycle management operation occurrence related to the coordination.
lcmOperationType	LcmOperationForCoordType	1	Indicates the type of the LCM operation with which coordination is requested. Shall be the same as the value of the "lcmOperationType" attribute in the NsLcmOpOcc structure that is referenced by the "nsLcmOpOccId".
coordinationActionName	IdentifierInNsd	1	Indicates the LCM coordination action.  See note.
inputParams	KeyValuePairs	0..1	Additional input parameters passed as input to the coordination action.
_links	Structure (inlined)	1	Links to resources related to this request.
>nsLcmOpOcc	Link	1	Related lifecycle management operation occurrence.
>nsInstance	Link	1	Related NS instance.
NOTE: How to determine the supported coordination actions is outside the scope of the present version of the present document.			

#### 12.5.2.3 Type: LcmCoord

This type represents an LCM coordination result. It shall comply with the provisions defined in table 12.5.2.3-1.

Table 12.5.2.3-1: Definition of the LcmCoord data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this coordination result.
coordinationResult	LcmCoordResultType	1	The result of executing the coordination action which also implies the action to be performed by the NFVO as the result of this coordination.
nsInstancelId	Identifier	1	Identifier of the NS instance which this coordination request is related to.
nsLcmOpOcclId	Identifier	1	The identifier of the NS lifecycle management operation occurrence related to the coordination.
lcmOperationType	LcmOperationForCoordType	1	Indicates the type of the LCM operation with which coordination is requested. Shall be the same as the value of the "lcmOperationType" attribute in the NsLcmOpOcc structure that is referenced by the "nsLcmOpOcclId".
coordinationActionName	String	1	Indicates the actual LCM coordination action. See note.
outputParams	KeyValuePairs	0..1	Additional parameters returned by the coordination action, e.g. on the reason for the indicated coordinationResult.
warnings	String	0..N	Warning messages that were generated while the operation was executing.
error	ProblemDetails	0..1	Error information related to the coordination.  This attribute shall be present if "coordinationResult" is "ABORT" and may be present if "coordinationResult" is "CANCELLED".  If provided, the error information should be represented in the "error" attribute of the related NsLcmOpOcc data structure.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>nsLcmOpOcc	Link	1	Related lifecycle management operation occurrence.
>nsInstance	Link	1	Related NS instance.
NOTE: How to determine the supported coordination actions is outside the scope of the present version of the present document.			

## 12.5.3 Referenced structured data types

### 12.5.3.1 Introduction

No particular referenced structured data types are defined for this interface.

## 12.5.4 Referenced simple data types and enumerations

### 12.5.4.1 Introduction

This clause defines simple data types that can be referenced from data structures defined in the previous clauses.

### 12.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.2.

### 12.5.4.3 Enumeration: LcmOperationForCoordType

The enumeration LcmOperationForCoordType defines the permitted values to represent NS lifecycle operation types in NS LCM operation coordination actions. It shall comply with the provisions defined in table 12.5.4.3-1.

**Table 12.5.4.3-1: Enumeration LcmOperationForCoordType**

Enumeration value	Description
INSTANTIATE	Represents the "Instantiate NS" LCM operation.
SCALE	Represents the "Scale NS" LCM operation.
UPDATE	Represents the "Update NS" LCM operation.
TERMINATE	Represents the "Terminate NS" LCM operation.
HEAL	Represents the "Heal NS" LCM operation.

### 12.5.4.4 Enumeration: LcmCoordResultType

The enumeration LcmCoordResultType defines the permitted values to represent the result of executing an LCM coordination action. The coordination result also implies the action to be performed by the NFVO as the follow-up to this coordination. The LcmCoordResultType shall comply with the provisions defined in table 12.5.4.4-1.

**Table 12.5.4.4-1: Enumeration LcmCoordResultType**

Enumeration value	Description
CONTINUE	The related LCM operation shall be continued, staying in the state "PROCESSING".
ABORT	The related LCM operation shall be aborted by transitioning into the state "FAILED_TEMP".
CANCELLED	The coordination action has been cancelled upon request of the API consumer, i.e. the NFVO. The related LCM operation shall be aborted by transitioning into the state "FAILED_TEMP".

# Annex A (informative): Mapping operations to protocol elements

## A.1 Overview

This annex provides the mapping between operations as defined in ETSI GS NFV-IFA 013 [3] and the corresponding resources and HTTP methods defined in the present document.

## A.2 NSD Management interface

The mapping of NSD management interface operations, defined in ETSI GS NFV-IFA 013 [3], to the resources and HTTP methods defined in the present document can be found in table A.2-1.

**Table A.2-1: Mapping of ETSI GS NFV-IFA 013 [3] NSD Management interface operations with resources and HTTP methods**

ETSI GS NFV-IFA 013 [3] NSD Management interface operation	HTTP method	Resource	Direction
Create NSD Info	POST	ns_descriptors	OSS/BSS → NFVO
Upload NSD	PUT	ns_descriptors/{nsdInfold}/nsd_content	OSS/BSS → NFVO
Update NSD Info	PATCH	ns_descriptors/{nsdInfold}	OSS/BSS → NFVO
Delete NSD	DELETE	ns_descriptors/{nsdInfold}	OSS/BSS → NFVO
Query NSD	GET	ns_descriptors	OSS/BSS → NFVO
	GET	ns_descriptors/{nsdInfold}	OSS/BSS → NFVO
Fetch NSD	GET	ns_descriptors/{nsdInfold}/nsd_content	OSS/BSS → NFVO
Create PNFD Info	POST	pnf_descriptors	OSS/BSS → NFVO
Upload PNFD	PUT	pnf_descriptors/{pnfdInfold}/pnfd_content	OSS/BSS → NFVO
Update PNFD Info	POST	pnf_descriptors/{pnfdInfold}	OSS/BSS → NFVO
Delete PNFD	DELETE	pnf_descriptors/{pnfdInfold}	OSS/BSS → NFVO
Query PNFD Info	GET	pnf_descriptors	OSS/BSS → NFVO
	GET	pnf_descriptors/{pnfdInfold}	OSS/BSS → NFVO
Fetch PNFD	GET	pnf_descriptors/{pnfdInfold}/pnfd_content	OSS/BSS → NFVO
Subscribe	POST	subscriptions	OSS/BSS → NFVO
Query Subscription Information	GET	subscriptions	OSS/BSS → NFVO
	GET	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Terminate Subscription	DELETE	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Notify	POST	(provided by API consumer)	NFVO → OSS/BSS

## A.3 NS Lifecycle Management interface

The mapping of NS lifecycle management operations, defined in ETSI GS NFV-IFA 013 [3], to the resources and HTTP methods defined in the present document can be found in table A.3-1.

**Table A.3-1: Mapping of ETSI GS NFV-IFA 013 [3] operations with NS Lifecycle Management interface resources and methods**

ETSI GS NFV-IFA 013 [3] NS Lifecycle Management interface operation	HTTP method	Resource	Direction
Create NS Identifier	POST	ns_instances	OSS/BSS → NFVO
Instantiate NS	POST	ns_instances/{nsInstanceld}/instantiate	OSS/BSS → NFVO
Scale NS	POST	ns_instances/{nsInstanceld}/scale	OSS/BSS → NFVO
Update NS	POST	ns_instances/{nsInstanceld}/update	OSS/BSS → NFVO
Terminate NS	POST	ns_instances/{nsInstanceld}/terminate	OSS/BSS → NFVO
Delete NS Identifier	DELETE	ns_instances/{nsInstanceld}	OSS/BSS → NFVO
Query NS	GET	ns_instances/{nsInstanceld}	OSS/BSS → NFVO
	GET	ns_instances	OSS/BSS → NFVO
Heal NS	POST	ns_instances/{nsInstanceld}/heal	OSS/BSS → NFVO
Get Operation Status	GET	ns_lcm_op_occs	OSS/BSS → NFVO
	GET	ns_lcm_op_occs/{nsLcmOpOcclId}	OSS/BSS → NFVO
Subscribe	POST	subscriptions	OSS/BSS → NFVO
Query Subscription Information	GET	subscriptions	OSS/BSS → NFVO
	GET	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Terminate Subscription	DELETE	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Notify	POST	(provided by API consumer)	NFVO → OSS/BSS

## A.4 NS Performance Management interface

The mapping of NS performance management operations, defined in ETSI GS NFV-IFA 013 [3], to the resources and HTTP methods defined in the present document can be found in table A.4-1.

**Table A.4-1: Mapping of ETSI GS NFV-IFA 013 [3] NS Performance Management interface operations with resources and HTTP methods**

ETSI GS NFV-IFA 013 [3] NS Performance Management operation	HTTP method	Resource	Direction
Create PM Job	POST	pm_jobs	OSS/BSS → NFVO
Delete PM Job	DELETE	pm_jobs/{pmJobId}	OSS/BSS → NFVO
Query PM Job	GET	pm_jobs	OSS/BSS → NFVO
	GET	pm_jobs/{pmJobId}	OSS/BSS → NFVO
Create Threshold	POST	thresholds	OSS/BSS → NFVO
Delete Threshold	DELETE	thresholds/{thresholdId}	OSS/BSS → NFVO
Query Threshold	GET	thresholds	OSS/BSS → NFVO
	GET	thresholds/{thresholdId}	OSS/BSS → NFVO
Subscribe	n/a	see note	n/a
Query Subscription Information	n/a	see note	n/a
	n/a	see note	n/a
Terminate Subscription	n/a	see note	n/a
Notify	POST	(provided by API consumer)	NFVO → OSS/BSS
NOTE:	In the NS Performance Management interface, support for subscriptions has been dropped in version 2.7.1 of the present document in favour of controlling the delivery of notifications directly by the "Thresholds" and "PM jobs" resources.		

## A.5 NS Fault Management interface

The mapping of NS fault management operations, defined in ETSI GS NFV-IFA 013 [3], to the resources and HTTP methods defined in the present document can be found in table A.5-1.

**Table A.5-1: Mapping of ETSI GS NFV-IFA 013 [3] NS Fault Management interface operations with resources and HTTP methods**

ETSI GS NFV-IFA 013 [3] NS Fault Management interface operation	HTTP method	Resource	Direction
Get Alarm List	GET	alarms	OSS/BSS → NFVO
	GET	alarms/{alarmId}	OSS/BSS → NFVO
Acknowledge Alarm	PATCH	alarms/{alarmId}	OSS/BSS → NFVO
Subscribe	POST	subscriptions	OSS/BSS → NFVO
Query Subscription Information	GET	subscriptions	OSS/BSS → NFVO
	GET	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Terminate Subscription	DELETE	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Notify	POST	(provided by API consumer)	NFVO → OSS/BSS

## A.6 VNF Package Management interface

The mapping of VNF package management operations, defined in ETSI GS NFV-IFA 013 [3], to the resources and HTTP methods defined in the present document can be found in table A.6-1.

**Table A.6-1: Mapping of ETSI GS NFV-IFA 013 [3] operations with VNF Package Management interface resources and methods**

ETSI GS NFV-IFA 013 [3] VNF Package Management interface operation	HTTP method	Resource	Direction
Create VNF Package Info	POST	vnf_packages	OSS/BSS → NFVO
Update VNF Package Info	PATCH	vnf_packages/{vnfPkgId}	OSS/BSS → NFVO
Delete VNF Package	DELETE	vnf_packages/{vnfPkgId}	OSS/BSS → NFVO
Query VNF Package Info	GET	vnf_packages	OSS/BSS → NFVO
	GET	vnf_packages/{vnfPkgId}	OSS/BSS → NFVO
	GET	vnf_packages/{vnfPkgId}/vnfd	OSS/BSS → NFVO
	GET	/vnf_packages/{vnfPkgId}/manifest	OSS/BSS → NFVO
Upload VNF Package	PUT	vnf_packages/{vnfPkgId}/package_content	OSS/BSS → NFVO
	POST	vnf_packages/{vnfPkgId}/package_content/upload_from_uri	OSS/BSS → NFVO
Fetch VNF Package	GET	vnf_packages/{vnfPkgId}/package_content	OSS/BSS → NFVO
Fetch VNF Package Artifacts	GET	/vnf_packages/{vnfPkgId}/artifacts	OSS/BSS → NFVO
	GET	vnf_packages/{vnfPkgId}/artifacts/{artifactPath}	OSS/BSS → NFVO
Subscribe	POST	subscriptions	OSS/BSS → NFVO
Query Subscription Information	GET	subscriptions	OSS/BSS → NFVO
	GET	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Terminate subscription	DELETE	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Notify	POST	(provided by API consumer)	NFVO → OSS/BSS

## A.7 NFVI Capacity Information interface

The mapping of NFVI capacity information operations, defined in ETSI GS NFV-IFA 013 [3], to the resources and HTTP methods defined in the present document can be found in table A.7-1.



**Table A.7-1: Mapping of ETSI GS NFV-IFA 013 [3] operations with NFVI Capacity Information interface resources and methods**

ETSI GS NFV-IFA 013 [3] NFVI Capacity Information interface operation	HTTP method	Resource	Direction
Query NFVI Capacity	GET	nfvi_capacity_infos	OSS/BSS → NFVO
Create Capacity Threshold	POST	capacity_thresholds	OSS/BSS → NFVO
Delete Capacity Thresholds	DELETE	capacity_thresholds/{capacityThresholdId}	OSS/BSS → NFVO
Query Capacity Threshold	GET	capacity_thresholds/{capacityThresholdId}	OSS/BSS → NFVO
Notify	POST	(provided by API consumer)	NFVO → OSS/BSS

## A.8 VNF Snapshot Package Management interface

The mapping of VNF snapshot package management operations, defined in ETSI GS NFV-IFA 013 [3], to the resources and HTTP methods defined in the present document can be found in table A.8-1.

**Table A.8-1: Mapping of ETSI GS NFV-IFA 013 [3] operations with VNF Snapshot Package Management interface resources and methods**

ETSI GS NFV-IFA 013 [3] NFVI Capacity Information interface operation	HTTP method	Resource	Direction
Create VNF Snapshot Package Info	POST	vnf_snapshot_packages	OSS/BSS → NFVO
Build VNF Snapshot Package	POST	vnf_snapshot_packages/{vnfSnapshotPkgId}/package_content/build	OSS/BSS → NFVO
Upload VNF Snapshot Package	POST	vnf_snapshot_packages/{vnfSnapshotPkgId}/package_content/upload_from_uri	OSS/BSS → NFVO
Extract VNF Snapshot Package	POST	vnf_snapshot_packages/{vnfSnapshotPkgId}/package_content/extract	OSS/BSS → NFVO
Fetch VNF Snapshot Package	GET	vnf_snapshot_packages/{vnfSnapshotPkgId}/package_content	OSS/BSS → NFVO
Fetch VNF Snapshot Package Artifacts	GET	vnf_snapshot_packages/{vnfSnapshotPkgId}/artifacts/{artifactPath}	OSS/BSS → NFVO
Query VNF Snapshot Package Info	GET	vnf_snapshot_packages/{vnfSnapshotPkgId}	OSS/BSS → NFVO
	GET	vnf_snapshot_packages	OSS/BSS → NFVO
Delete VNF Snapshot Package	DELETE	vnf_snapshot_packages/{vnfSnapshotPkgId}	OSS/BSS → NFVO
Update VNF Snapshot Package	PATCH	vnf_snapshot_packages/{vnfSnapshotPkgId}	OSS/BSS → NFVO

## A.9 NS LCM Coordination interface

The mapping of NS lifecycle management operations, defined in ETSI GS NFV-IFA 013 [3], to the resources and HTTP methods defined in the present document can be found in table A.9-1.

**Table A.9-1: Mapping of ETSI GS NFV-IFA 013 [3] operations with NS Coordination interface resources and methods**

ETSI GS NFV-IFA 013 [3] NS Lifecycle Management interface operation	HTTP method	Resource	Direction
Coordinate LCM operation	POST	coordinations	NFVO → OSS/BSS
	GET	coordinations/{coordinationId}	NFVO → OSS/BSS
	POST	coordinations/{coordinationId}	NFVO → OSS/BSS

---

## Annex B (informative): State models

### B.1 NSD state model

#### B.1.1 Introduction

This clause describes the state model of NSD in the NFVO. It includes the state models for two phases, i.e. onboarding phase of NSD and operational phase of NSD.

#### B.1.2 State model

A given NSD has three states, i.e. on-boarding state, operational state and usage state.

The on-boarding state is represented by the "nsdOnboardingState" attribute in the "NsdInfo" data type with below values:

- **CREATED:** The NSD information object has been created.
- **UPLOADING:** The NSD is being uploaded.
- **PROCESSING:** The NSD is being processed, e.g. validation.
- **ONBOARDED:** The NSD has been on-boarded successfully.
- **ERROR:** There was an error during NSD uploading or processing.

The operational state is represented by the "nsdOperationalState" attribute in the "NsdInfo" data type with below values:

- **ENABLED:** The NSD is enabled.
- **DISABLED:** The NSD is disabled.

The usage state is represented by the "nsdUsageState" attribute in the "NsdInfo" data type with below values:

- **IN\_USE:** The NSD is in use.
- **NOT\_IN\_USE:** The NSD is not in use.

The state model of on-boarding phase in figure B.1.2-1 applies to a given NSD being on-boarded. Besides the operations and conditions specified in the figure, below operations are also considered as available during the on-boarding phase:

- Query NSD Info
- Update NSD Info (with user defined data only)

The state model of operational phase in figure B.1.2-1 applies to an on-boarded NSD. Besides the operations and conditions specified in the figure, below operations are also considered as available during the operational phase:

- Query NSD Info
- Update NSD Info (with user defined data only)
- Fetch NSD

At the end of the on-boarding phase, the "nsdOnboardingState" value transitions to "ONBOARDED" and the "nsdOperationalState" value transitions from "DISABLED" to "ENABLED", and the operational phase is entered.

The "nsdOperationalState" and "nsdUsageState" detail the state changes during the NSD operational phase. During the NSD on-boarding phase, the value of the "nsdOperationalState" is "DISABLED" and the value of the "nsdUsageState" is "NOT\_IN\_USE". Right after the NSD becomes on-boarded, the value of the "nsdOperationalState" is changed to "ENABLED" and the value of the "nsdUsageState" is kept as "NOT\_IN\_USE".

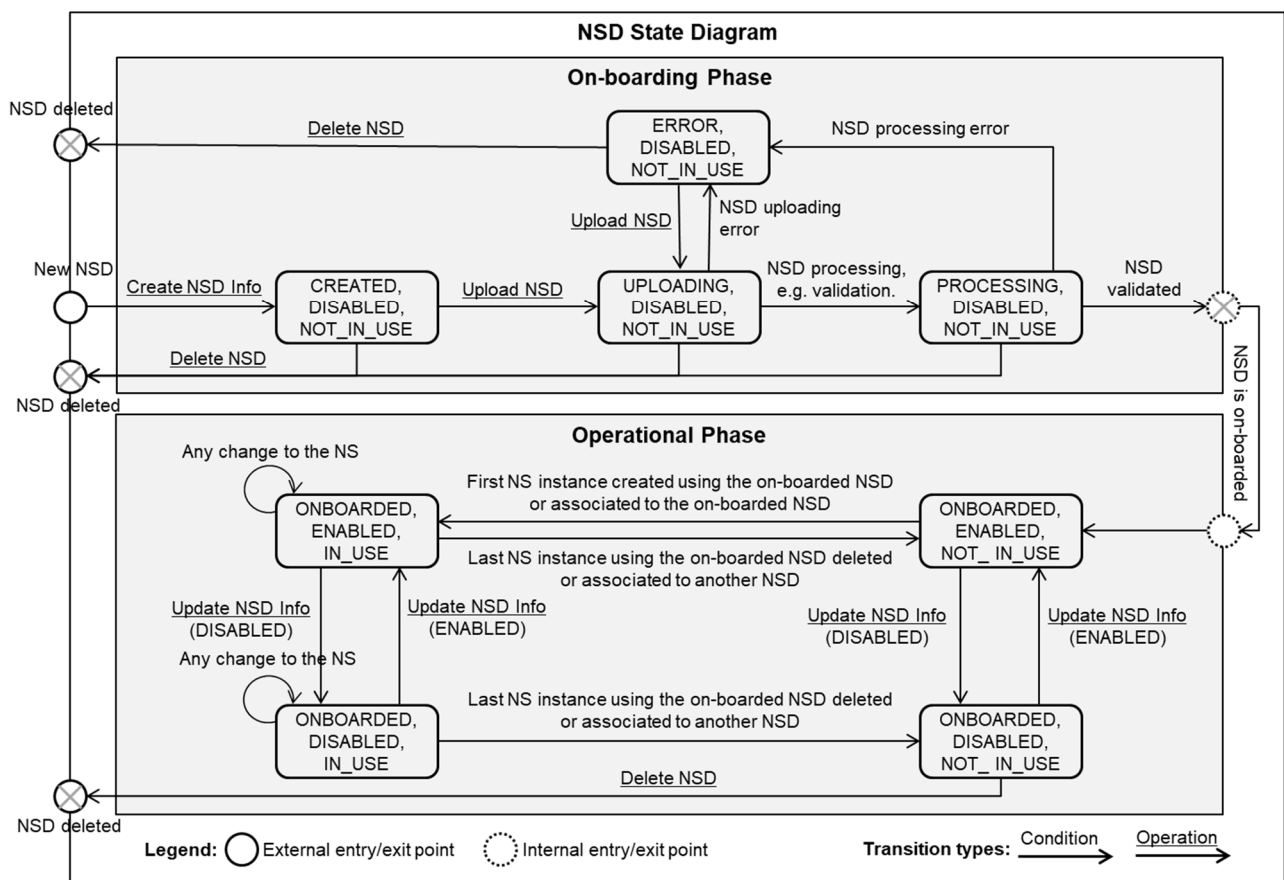


Figure B.1.2-1: NSD state model

## B.2 VNF package state model

### B.2.1 Introduction

This clause describes the state model of VNF Package in the NFVO. It includes the state models for two phases, i.e. on-boarding phase and operational phase.

### B.2.2 State model

A given VNF Package has three states, i.e. on-boarding state, operational state and usage state.

The on-boarding state is represented by the "onboardingState" attribute in the "VnfPkgInfo" information element with below values:

- **CREATED:** The VNF Package information object has been created.
- **UPLOADING:** The VNF Package or the external artifacts are being uploaded.
- **PROCESSING:** The VNF Package is being processed, e.g. validation.
- **ONBOARDED:** The VNF Package has been on-boarded successfully.

- **ERROR:** There was an error during uploading of the VNF package or the external artifacts, or during VNF package processing.

The operational state is represented by the "operationalState" attribute in the "VnfPkgInfo" information element with below values:

- **ENABLED:** The VNF Package is enabled.
- **DISABLED:** The VNF Package is disabled.

The usage state is represented by the "usageState" attribute in the "VnfPkgInfo" information element with below values:

- **IN\_USE:** The VNF Package is in use.
- **NOT\_IN\_USE:** The VNF Package is not in use.

The state model of on-boarding phase in figure B.2.2-1 applies to a given VNF Package being on-boarded. Besides the operations and conditions specified in the figure, below operations are also considered as available during the on-boarding phase:

- Query VNF Package Info
- Update VNF Package Info (with user defined data only)

The state model of operational phase in figure B.2.2-1 applies to an on-boarded VNF Package. Besides the operations and conditions specified in the figure, below operations are also considered as available during the operational phase:

- Query VNF Package Info
- Update VNF Package Info (with user defined data only)
- Fetch VNF Package
- Fetch VNF Package Artifacts

The "onboardingState" details the state changes during the VNF Package on-boarding phase. The value of this attribute during the VNF Package operational phase is "ONBOARDED".

The "operationalState" and "usageState" detail the state changes during the VNF Package operational phase. During the VNF Package on-boarding phase, the value of the "operationalState" is "DISABLED" and the value of the "usageState" is "NOT\_IN\_USE". Right after the VNF Package becomes on-boarded, the value of the "operationalState" is changed to "ENABLED" and the value of the "usageState" is kept as "NOT\_IN\_USE", as shown in figure B.2.2-1.

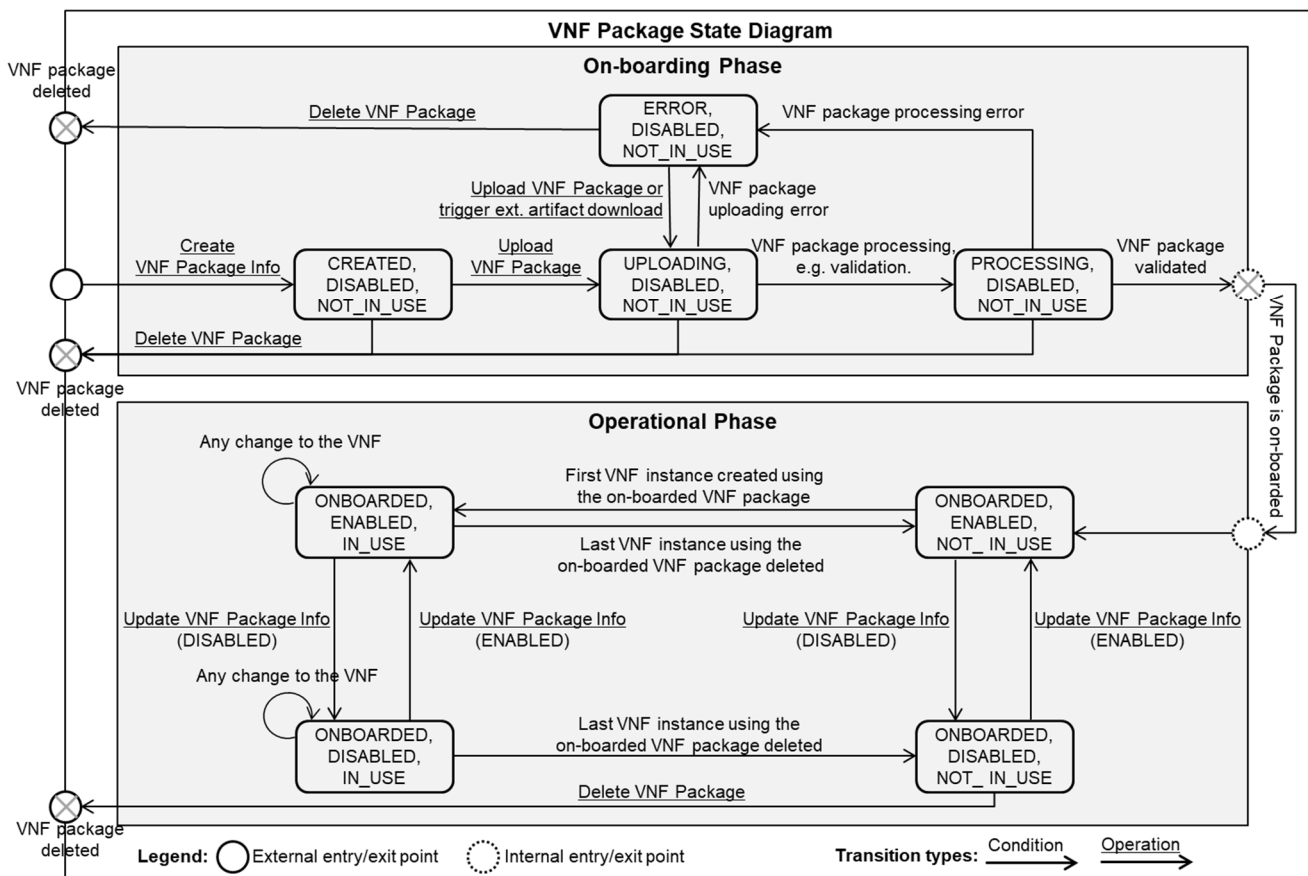


Figure B.2.2-1: VNF Package state model

---

## Annex C (informative): Complementary material for API utilization

To complement the definitions of each method, resource, and data type defined in the main body of the present document, the ETSI NFV ISG is providing supplementary description files, compliant to the OpenAPI Specification [i.3], for the Os-Ma-nfvo reference point. These supplementary description files, containing the OpenAPI Specification for each API defined in the present document, are located at <https://forge.etsi.org/rep/nfv/NFV-SOL005>.

In case of discrepancies between the supplementary files and the related data structure definitions in the main body of the present document, the data structure definitions take precedence.

The OpenAPI representations referenced above:

- 1) use the MAJOR.MINOR.PATCH version fields to signal the version of the API as defined in the present document; and
- 2) use the "impl" version parameter (see clause 9.1.2 of ETSI GS NFV-SOL 013 [16]) to represent changes to the OpenAPI representation without changing the present document.

It is specified in clause 6 of ETSI GS NFV-SOL 015 [i.7] how the OpenAPI specification references the present document and signals the version information.

## Annex D (informative): History of features added to the present document

### D.1 Overview

The present document has been first released as part of ETSI NFV Release 2 and went through multiple cycles of maintenance.

In ETSI NFV Release 3, features were added. The branching has occurred after version 2.7.1 of the present document.

This annex lists the features that were added on top of Release 2 in Release 3. To help implementers to determine which changes make up together a particular feature, these are documented below per feature.

### D.2 Features added in Release 3

#### D.2.1 FEAT02: VNF Software modification

This feature addresses the initiation and the coordination of the software modification process related to VNFs. Goal is to minimize the impact of software modification on service availability.

**Table D.2.1-1: Changes that make up the feature**

Clause	Interface	Content of the change	Type of change
6.5.2.12	ns lcm	UpdateNsRequest: added value CHANGE_VNFPKG	New enum value
6.5.2.12	ns lcm	UpdateNsRequest.changeVnfPackageData	New attribute
6.5.3.2	ns lcm	AffectedVnf: added value CHANGE_VNFPKG	New enum value
6.5.3.2	ns lcm	AffectedVnf.modificationsTriggeredByVnfPkgChange	New attribute
6.5.3.51	ns lcm	VnfScaleInfo.vnfId	New attribute
6.5.3.59	ns lcm	ExtManagedVirtualLinkInfo.vnfId	New attribute
6.5.3.60	ns lcm	VnfResourceInfo.vnfId	New attribute
6.5.3.61	ns lcm	VnfVirtualLinkResourceInfo.vnfId	New attribute
6.5.3.69	ns lcm	VnfMonitoringParameter.vnfId	New attribute
6.5.3.70	ns lcm	VnfExtCplInfo.vnfId	New attribute

#### D.2.2 FEAT04: Host reservation

The present enhancement proposes adding the capability to the NFV-MANO architectural framework to support the reservation of compute hosts.

**Table D.2.2-1: Changes that make up the feature**

Clause	Interface	Content of the change	Type of change
10	nfvi	NFVI capacity information interface	New API

#### D.2.3 FEAT10: Multi-site connectivity services

This feature specifies management requirements, interfaces and information models to support connectivity for multi-site services.

**Table D.2.3-1: Changes that make up the feature**

Clause	Interface	Content of the change	Type of change
6.5.3.27	nslcm	ExtManagedVirtualLinkData.vnfLinkPort	New attribute
6.5.3.27	nslcm	ExtManagedVirtualLinkData.extManagedMultisiteVirtualLinkId	New attribute
6.5.3.59	nslcm	ExtManagedVirtualLinkInfo.extManagedMultisiteVirtualLinkId	New attribute
6.5.2.11	nslcm	InstantiateNsRequest.wanConnectionData	New attribute
6.5.2.10	nslcm	NsInstance.wanConnectionInfo	New attribute
6.5.2.12	nslcm	UpdateNsRequest: added value MODIFY_WAN_CONNECTION_INFO	New enum value
6.5.2.12	nslcm	UpdateNsRequest.modifyWanConnectionInfoData	New Attribute
6.5.3.6	nslcm	AffectedNs: new structure changeInfo.wanConnectionInfoModifications	Modified permitted attribute values

## D.2.4 FEAT15: VNF snapshotting

VNF Snapshot is a replication of a VNF instance at a specific point in time with a corresponding VNF Snapshot Package which is collection of files representing a VNF Snapshot. The feature implementation enables operations on and management of VNF Snapshots and their corresponding packages.

**Table D.2.4-1: Changes that make up the feature**

Clause	Interface	Content of the change	Type of change
6.2	nslcm	/vnf_snapshots	New resource
6.2	nslcm	/vnf_snapshots/{vnfSnapshotInfoId}	New resource
6.4.19	nslcm	VNF snapshots resource: GET method	New method
6.4.20	nslcm	Individual VNF snapshot resource: GET/DELETE methods	New method
6.5.2.10	nslcm	NsInstance.vnfSnapshotInfoIds	New attribute
6.5.2.12	nslcm	UpdateNsRequest: added value CREATE_VNF_SNAPSHOT	New enum value
6.5.2.12	nslcm	UpdateNsRequest: added value REVERT_VNF_TO_SNAPSHOT	New enum value
6.5.2.12	nslcm	UpdateNsRequest: added value DELETE_VNF_SNAPSHOT_INFO	New enum value
6.5.2.12	nslcm	UpdateNsRequest.createSnapshotVnfInstanceId	New attribute
6.5.2.12	nslcm	UpdateNsRequest.revertVnfToSnapshotData	New attribute
6.5.2.12	nslcm	UpdateNsRequest.deleteVnfSnapshotData	New attribute
6.5.2.17	nslcm	VnfSnapshotInfo	New resource data type
6.5.2.18	nslcm	VnfSnapshot	New resource data type
11	vnfsnapsh otpkgm	VNF snapshot package management interface	New API

## D.2.5 Additional new functionality outside the "NFV features" scheme

### D.2.5.1 Trunking support

The parameters that provide external CP data have been modified to support trunking and to allow easier modification.



**Table D.2.5.1-1: Changes that make up the feature**

Clause	Interface	Content of the change	Type of change
6.5.2.12	nslcm	UpdateNsRequest.added value CREATE_NS_VIRTUAL_LINK	New enum value
6.5.2.12	nslcm	UpdateNsRequest.added value DELETE_NS_VIRTUAL_LINK	New enum value
6.5.2.12	nslcm	UpdateNsRequest.addNsVirtualLinkData	New attribute
6.5.2.12	nslcm	UpdateNsRequest.deleteNsVirtualLinkId	New attribute
6.5.3.12	nslcm	IpOverEthernetAddressData.segmentationId	New attribute
6.5.3.12	nslcm	IpOverEthernetAddressData.segmentationType	New attribute
6.5.3.18	nslcm	IpOverEthernetAddressInfo.segmentationId	New attribute
6.5.3.28	nslcm	ExtLinkPortData.trunkResourceId	New attribute
6.5.3.29	nslcm	VnfExtCpData.cpConfig	Other change: turn this attribute into a map, Modified attribute semantics
6.5.3.33	nslcm	ChangeExtVnfConnectivityData	Other change: added normative provision
6.5.3.60	nslcm	VnfcResourceInfo.vnfcCpInfo.parentCpld	New attribute
6.5.3.63	nslcm	ExtLinkPortInfo.trunkResourceId	New attribute
6.5.3.64	nslcm	VnfLinkPortInfo.trunkResourceId	New attribute

## D.2.5.2 Verbosity of VNF LCM operation occurrence notifications

This change enables to control the verbosity of NS LCM operation occurrence notifications.

**Table D.2.5.2-1: Changes that make up the feature**

Clause	Interface	Content of the change	Type of change
6.5.2.2	nslcm	LccnSubscriptionRequest.verbosity	New attribute
6.5.2.4	nslcm	LccnSubscription.verbosity	New attribute
6.5.2.5	nslcm	NsLcmOperationOccurrenceNotification.verbosity	New attribute
6.5.2.5	nslcm	NsLcmOperationOccurrenceNotification	Other change: Allow to omit certain attributes depending on the value of the "verbosity" attribute

## D.2.5.3 Support for virtual IP connection points

The VNF connectivity model has been updated to support virtual IP connection points (VIP CPs). Refer to clause A.4 in ETSI GS NFV-IFA 007 [19] for the supported use cases.

**Table D.2.5.3-1: Changes that make up the feature**

Clause	Interface	Content of the change	Type of change
6.5.3.26	nslcm	ExtVirtualLinkData.extLinkPorts	Modified attribute semantics (added support for ports related to a VIP CP)
6.5.3.30	nslcm	VnfExtCpConfig.createExtLinkPort	New attribute
6.5.3.57	nslcm	VnfInstance.instantiatedVnfInfo.vipCpInfo	New attribute
6.5.3.63	nslcm	ExtLinkPortInfo.secondaryCpInstanceId	New attribute
6.5.3.64	nslcm	VnfLinkPortInfo.vipCpInstanceId	New attribute
6.5.3.70	nslcm	VnfExtCpInfo.associatedVipCpld	New attribute

## D.2.5.4 NS LCM coordination

LCM coordination allows an ongoing LCM operation occurrence to trigger a related management operation by the OSS/BSS, to wait for its result, and to coordinate that management operation with the LCM operation occurrence. This functionality is used e.g. by FEAT05 if such coordination is needed.

**Table D.2.5.4-1: Changes that make up the feature**

Clause	Interface	Content of the change	Type of change
12	lcmcoord	NS LCM coordination interface	New API
6.5.2.3	ns lcm	NsLcmOpOcc.lcmCoordinations	New attribute
6.5.2.3	ns lcm	NsLcmOpOcc.rejectedLcmCoordinations	New attribute

## D.2.6 FEAT05: Network slicing

This feature specifies the support for Network Slicing within NFV-MANO architectural framework.

**Table D.2.6-1: Changes that make up the feature**

Clause	Interface	Content of the change	Type of change
6.3.3a	ns lcm	Flows of NS lifecycle management operations triggered by task resources with pre-emption	Other change - new flows were created
6.4.18.3	ns lcm	<<Callback URI>>(NsLcmCapacityShortageNotification)	New notification
6.5.2.19	ns lcm	NsLcmCapacityShortageNotification	New notification
6.5.3.8	ns lcm	LifecycleChangeNotificationsFilter.notificationTypes	New attribute
6.5.3.8	ns lcm	LifecycleChangeNotificationsFilter.affectedNsInstanceIds	New attribute

---

## D.3 Features added in Release 4

### D.3.1 FEAT17: Cloud-Native VNFs and Container Infrastructure management

This feature addresses the following capabilities:

- NFV Architecture support for VNFs which follow "cloud-native" design principles.
- Enhance NFV-MANO capabilities to support container technologies.
- Enhance NFV-MANO capabilities for container management and orchestration.
- Enhance information model for containerized VNFs both using bare metal or nested virtualisation technologies.

Table D.3.1-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
4.4.1.6	multiple	ResourceHandle.vimId	Modified attribute semantics
4.4.1.6	multiple	ResourceHandle.vimLevelAdditionalResourceInfo, ResourceHandle.containerNamespace	New attribute
4.4.1.6	multiple	ResourceHandle.resourceId, ResourceHandle.vimLevelResourceType	Modified attribute semantics
4.4.1.6	multiple	ResourceHandle: updated note 1, added note 2	Other change: added normative provision
4.4.2.2	multiple	IdentifierInVim description	Modified attribute semantics
6.5.3.11	nslcm	CpProtocolData.layerProtocol: added IP_FOR_VIRTUAL_CP	Additional enumeration value
6.5.3.11	nslcm	CpProtocolData.virtualCpAddress	New attribute
6.5.3.12a	nslcm	VirtualCpAddressData	New resource data type
6.5.3.18a	nslcm	VirtualCpAddressInfo	New resource data type
6.5.3.26	nslcm	ExtVirtualLinkData.extNetAttDefResourceData	New attribute
6.5.3.26	nslcm	ExtVirtualLinkData: added note 2	Other change: added normative provision
6.5.3.27	nslcm	ExtManagedVirtualLinkData.netAttDefResourceData, ExtManagedVirtualLinkData.intCp	New attributes
6.5.3.27	nslcm	ExtManagedVirtualLinkData: added notes 1 and 2	Other change: added normative provision
6.5.3.30	nslcm	VnfExtCpConfig description	Other change: updated normative provision
6.5.3.30	nslcm	VnfExtCpConfig.netAttDefResourceId	New attribute
6.5.3.30	nslcm	VnfExtCpConfig: updated note 1, added notes 3, 4 and 5	Other change: added normative provision
6.5.3.30	nslcm	VnfExtCpConfig.cpProtocolData	Modified attribute semantics
6.5.3.57	nslcm	VnfInstance.instantiatedVnfInfo.virtualCpInfo	New attribute
6.5.3.57	nslcm	VnfInstance.instantiatedVnfInfo.mciInfo	New attribute
6.5.3.58	nslcm	CpProtocolInfo.virtualCpAddress	New attribute
6.5.3.58	nslcm	CpProtocolInfo.layerProtocol: added IP_FOR_VIRTUAL_CP	Additional enumeration value
6.5.3.59	nslcm	ExtManagedVirtualLinkInfo.vnfNetAttDefResource	New attribute
6.5.3.60	nslcm	VnfcResourceInfo.vnfcCpInfo.netAttDefResourceId	New attribute
6.5.3.60	nslcm	VnfcResourceInfo: added notes 4 and 5	Other change: added normative provision
6.5.3.60	nslcm	VnfcResourceInfo description	Other change: updated normative provision
6.5.3.60	nslcm	VnfcResourceInfo.computeResource	Modified attribute semantics
6.5.3.60	nslcm	VnfcResourceInfo.storageResourceIds	Modified attribute semantics
6.5.3.61	nslcm	VnfVirtualLinkResourceInfo.networkResource	Modified attribute semantics
6.5.3.62	nslcm	ExtVirtualLinkInfo.extNetAttDefResource	New attribute
6.5.3.70	nslcm	VnfExtCpInfo.associatedVirtualCpId	New attribute
6.5.3.70	nslcm	VnfExtCpInfo.associatedVipCpId	Modified attribute semantics
6.5.3.70	nslcm	VnfExtCpInfo: updated note 1, added notes 3 and 4	Other change: added normative provision
6.5.3.70	nslcm	VnfExtCpInfo.netAttDefResourceId	New attribute
6.5.3.94	nslcm	VirtualStorageResourceInfo.storageResource	Modified attribute semantics
6.5.4.13	nslcm	McioTypeName	New string type

## D.3.2 FEAT21: NFV support for 5G

The scope of the feature covers the following areas:

- investigating the NFV support for deploying 5G networks, capabilities and associated requirements;
- determine the 5G network capabilities and features interworking and their relationship with NFV; and
- determine and profile how NFV can support 5G deployments.

Table D.3.2-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
5.5.2.2	nsd	NsdInfo.nsdInvariantId	Modified attribute semantics
5.5.2.2	nsd	NsdInfo: added notes 5, 6 and 7	Other change: added normative provision
5.5.2.2	nsd	NsdInfo.vnfPkgIds, NsdInfo.pnfInfolds, NsdInfo.nestedNsdInfolds	Modified attribute semantics
5.4.3.3.5	nsd	Resource: Individual NS Descriptor - DELETE method description	Other change: updated the normative provision
5.4.4.3.3	nsd	Resource: NSD archive content - PUT method description	Other change: updated the normative provision
5.4.6.3.5	nsd	Resource: Individual PNF Descriptor - DELETE method description	Other change: updated the normative provision
5.4.7.3.3	nsd	Resource: PNFD archive content - PUT method description	Other change: updated the normative provision
6.4.4.3.1	nslcm	Resource: Instantiate NS Task - POST method description	Other change: updated the normative provision
6.4.6.3.1	nslcm	Resource: Update NS task - POST method description	Other change: updated the normative provision
6.5.2.10	nslcm	NsInstance.versionDependency	New attribute
6.5.3.13	nslcm	PnfInfo.versionDependency	New attribute
6.5.3.14	nslcm	AddPnfData.overridingVersionDependency	New attribute
6.5.3.14	nslcm	AddPnfData: added note 2	Other change: added normative provision
6.5.3.19	nslcm	VnfInstanceData.overridingVersionDependency	New attribute
6.5.3.19	nslcm	VnfInstanceData: added notes 3 and 4	Other change: added normative provision
6.5.3.19	nslcm	VnfInstanceData.overridingVnfId	Modified attribute semantics
6.5.319a	nslcm	NestedNsInstanceData.overridingVersionDependency	New attribute
6.5.3.19a	nslcm	NestedNsInstanceData: added notes 3 and 4	Other change: added normative provision
6.5.3.19a	nslcm	NestedNsInstanceData.overridingNsdId	Modified attribute semantics
6.5.3.21a	nslcm	ParamsForNestedNs.overridingVersionDependency	New attribute
6.5.3.21a	nslcm	ParamsForNestedNs: added notes 2 and 3	Other change: added normative provision
6.5.3.21a	nslcm	ParamsForNestedNs.overridingNsdId	Modified attribute semantics
6.5.3.22	nslcm	ParamsForVnf.overridingVersionDependency	New attribute
6.5.3.22	nslcm	ParamsForVnf: added notes 2 and 3	Other change: added normative provision
6.5.3.22	nslcm	ParamsForVnf.overridingVnfId	Modified attribute semantics
6.5.3.24	nslcm	InstantiateVnfData.overridingVersionDependency	New attribute
6.5.3.24	nslcm	InstantiateVnfData: added notes 8 and 9	Other change: added normative provision
6.5.3.24	nslcm	InstantiateVnfData.overridingVnfId	Modified attribute semantics
6.5.3.54	nslcm	ChangeVnfPackageData.overridingVersionDependency	New attribute
6.5.3.54	nslcm	ChangeVnfPackageData: added note	Other change: added normative provision
6.5.3.57	nslcm	VnfInstance.versionDependency	New attribute
6.5.3.113	nslcm	VersionDependencyStatement	New resource data type
9.4.3.3.5	vnfpkgm	Resource: Individual VNF package - DELETE method description	Other change: updated the normative provision
9.4.5.3.3	vnfpkgm	Resource: VNF package content - PUT method description	Other change: updated the normative provision

## D.3.3 ENH02: Special technical enhancements

### D.3.3.1 ENH02.02: NFV-MANO enhancement for NS feasibility check

This enhancement proposes to add the capability of feasibility check of Network Service to the lifecycle management.

It suggests enhancing the Os-Ma-nfvo reference point to support the feasibility check of Network Service that can be used, for example, to check the feasibility of Network Slicing subnet by sending enquiry requests to NFV-MANO to determine the availability of the network constituents.

**Table D.3.3.1-1: Changes that make up the feature**

Clause	Interface	Content of the change	Type of change
6.4.10.3.4	nslcm	PATCH method	New method
6.5.2.5	nslcm	NsLcmOperationOccurrenceNotification description	Other change: updated the normative provision
6.5.2.5	nslcm	NsLcmOperationOccurrenceNotification: added note 2	Other change: added normative provision
6.5.2.5	nslcm	NsLcmOperationOccurrenceNotification.feasibilityCheckResult	New attribute
6.5.2.11	nslcm	InstantiateNsRequest.startTime	Modified attribute semantics
6.5.2.11	nslcm	InstantiateNsRequest.feasibilityCheckReserve	New attribute
6.5.2.11	nslcm	InstantiateNsRequest: added notes 7, 8, 9, 10 and 11	Other change: added normative provision
6.5.2.12	nslcm	UpdateNsRequest.updateTime	Modified attribute semantics
6.5.2.12	nslcm	UpdateNsRequest.feasibilityCheckReserve	New attribute
6.5.2.12	nslcm	UpdateNsRequest: added notes 6, 7, 8, 9 and 10	Other change: added normative provision
6.5.2.14	nslcm	ScaleNsRequest.scaleTime	Modified attribute semantics
6.5.2.15	nslcm	TerminateNsRequest.terminationTime	Modified attribute semantics
6.5.2.20	nslcm	NsLcmInfoModificationRequest	New resource data type
6.5.4.14	nslcm	FeasibilityCheckReserveType	New enumeration type
6.5.4.15	nslcm	FeasibilityCheckResultType	New enumeration type

### D.3.3.2 ENH02.03: NFV-MANO enhancement with Data flow mirroring

This enhancement proposes the NFV-MANO functionality to enable data flow mirroring management. It was specified the intra NFVI-PoP data flow mirroring requirements in the NSD model in order to provide the flow information need to be mirrored at the design time. The Or-Vi reference point was also enhanced in order to manage the intra NFVI-PoP data flow mirroring during the lifecycle of an NS.

**Table D.3.3.2-1: Changes that make up the feature**

Clause	Interface	Content of the change	Type of change
6.5.2.10	nslcm	NsInstance.dataFlowMirroringInfo	New attribute
6.5.2.11	nslcm	InstantiateNsRequest.dataFlowMirroringData	New attribute
6.5.2.12	nslcm	UpdateNsRequest.updateType: added CREATE_DATA_FLOW_MIRRORING, DELETE_DATA_FLOW_MIRRORING and UPDATE_DATA_FLOW_MIRRORING	Additional enumeration values
6.5.2.12	nslcm	UpdateNsRequest.createDataFlowMirroring	New attribute
6.5.2.12	nslcm	UpdateNsRequest.deleteDataFlowMirroringId	New attribute
6.5.2.12	nslcm	UpdateNsRequest.updateDataFlowMirroring	New attribute

### D.3.3.3 ENH02.04: Invariant identification of NSD constituents

This enhancement feature proposes to add the capability to identify the VNFDs, nested NSDs and PNFDs of an NSD by invariant identities, as an alternative option to the descriptor Ids. The aim is to avoid having to change and create a new NSD when its components (VNFDs, PNFDs or nested NSDs) are replaced by another version and this replacement does not require changes in the rest of the NSD.

**Table D.3.3.3-1: Changes that make up the feature**

Clause	Interface	Content of the change	Type of change
6.5.2.11	nslcm	InstantiateNsRequest.targetNsScaleLevelInfo	New attribute Other change: added normative provision: notes 5 and 6
6.5.3.24	nslcm	InstantiateVnfData.targetScaleLevelInfo	New attribute Other change: added normative provision: notes 4 and 5
6.5.3.25	nslcm	ChangeVnfFlavourData.targetScaleLevelInfo	New attribute Other change: added normative provision: notes 3 and 4
5.5.2.2	nsd	NsdInfo.nsdInvariantId and NsdInfo.nsdExtInvariantId	New attribute
5.5.2.5	nsd	PnfdInfo.pnfdExtInvariantId	New attribute
9.5.2.5	vnfpgm	VnfPkgInfo.vnfdExtInvariantId	New attribute

### D.3.3.4 ENH02.05: Flexibility with scalable VNF/NS instantiation

This enhancement feature proposes to add scaleLevels as input during instantiation to support flexibility with scalable VNF/NS instantiation. The VNFs/NSs supporting flexible instantiations are identified with VNFD/NSD level attribute(s). This enhancement provides flexibility for the service providers to adjust instantiation level when instantiating a VNF and supports instantiate a VNF with required size in one single operation.

**Table D.3.3.4-1: Changes that make up the feature**

Clause	Interface	Content of the change	Type of change
6.5.3.11	nslcm	InstantiateNsRequest	Other change: updated normative provision: notes 1, 2, 3 and 4
6.5.2.12	nslcm	UpdatedNsRequest	Other change: added normative provision: note 5
6.5.3.14	nslcm	AddPnfData.pnfdId	Other change: added normative provision: pnfdId description and note
6.5.3.19	nslcm	VnfInstanceData.overridingVnfdId	New attribute Other change: added normative provision: notes 1 and 2
6.5.3.19a	nslcm	NestedNsInstanceData.overridingNsdId	New attribute Other change: added normative provision: notes 1 and 2
6.5.3.21a	nslcm	ParamsForNestedNs.overridingNsdId	New attribute Other change: added normative provision: note
6.5.3.22	nslcm	ParamsForVnf.overridingVnfdId	New attribute Other change: added normative provision: note
6.5.3.24	nslcm	InstantiateVnfData.overridingVnfdId	New attribute Other change: added normative provision: notes 4 and 5
6.5.3.34	nslcm	AssocNewNsdVersionData	Other change: added normative provision: note

---

# Annex E (informative): Guidelines and examples for the use of ETSI GS NFV SOL 005 APIs for multi-site service connectivity

## E.1 Overview and concepts description

### E.1.1 Introduction

NS VL and VNF VL can span network resources across multiple sites (NFVI-PoPs) and WAN, as specified in clause 5.4 of ETSI GS NFV-IFA 010 [2]. In terms of integration of WAN infrastructure management, two cases are supported:

- The WIM is part of the NFV-MANO framework.
- The WIM is external to the NFV-MANO framework (e.g. under control of other OSS/BSS systems).

When the WIM becomes part of the NFV-MANO framework, overall NFV orchestration, including orchestration and management of WAN resources is performed within NFV-MANO. The case that the WIM is external to the NFV-MANO framework is envisioned for cases where WAN resources are pre-provisioned or controlled under the responsibility of other OSS/BSS systems. More information with regards to the WIM integration is present in annex G of ETSI GS NFV-IFA 010 [2].

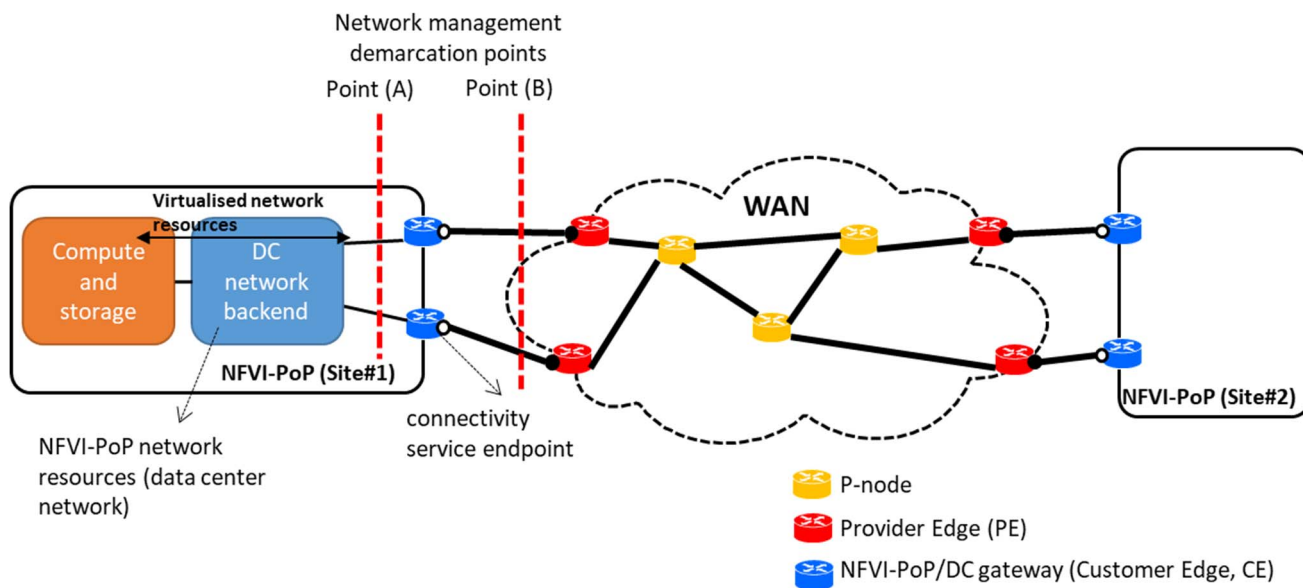
The inter-connection ("stitching") of virtualised network resources in the NFVI-PoP with WAN resources is performed at the boundary between the site (NFVI-PoP) and the network links to/from the WAN at the NFVI-PoP network gateway (also referred as Data Centre Gateway (DC-GW) or Customer Edge (CE) node), as described in clause 4.3 of ETSI GS NFV-IFA 032 [i.10].

### E.1.2 NFVI-PoP network gateway management models

With regards to the NFVI-PoP network gateway, different management models are possible from the perspective of the WAN service provider. In all cases, the WAN service provider is responsible for the management of the Multi-Site Connectivity Services (MSCS) as supported by the WIM and as defined in the ETSI GS NFV-IFA 032 [i.10]:

- WAN provider managed: the NFVI-PoP network gateway is managed by the WAN service provider. In this case, the responsibility boundary is between the gateway and the virtualised network resources in the NFVI-PoP. This case is illustrated as demarcation point A in figure E.1.2-1.
- NFVI-PoP network provider managed: the NFVI-PoP network gateway is managed by the NFVI-PoP network provider. In this case, the responsibility boundary is between the gateway and the Provider Edge (PE) router on the WAN. This case is illustrated as demarcation point B in figure E.1.2-1.
- Co-managed: the NFVI-PoP network gateway is managed by the WAN service provider, but the NFVI-PoP network provider can access (either directly or via the NFV-MANO) the network gateway for configuration and/or monitoring purposes. The responsibility boundary is the same as in the provider managed case (demarcation point A).

Figure E.1.2-1 illustrates the network management demarcation points.



**Figure E.1.2-1: Generic multi-site network topology and network management demarcation points**

Based on the WIM to NFV-MANO integration options (introduced in clause E.1.1) and the NFVI-PoP network gateway management, table E.1.2-1 describes the responsibilities and roles of NFV-MANO framework.

NOTE: The analysis of responsibilities and roles of NFV-MANO framework in table E.1.2-1 leverages as much as possible the orchestration and automation capabilities of the NFV-MANO framework. Cases in which other operating support systems are involved in the management of certain resources are also possible depending on how the WAN and NFVI network resources are integrated into the operator's network.

**Table E.1.2-1: Roles and responsibilities of NFV-MANO in multi-site network connectivity**

		Type of resources	WIM integration	
			Part of the NFV-MANO framework (case wim.1)	External to the NFV-MANO framework (case wim.2)
NFVI-PoP network gateway management	WAN provider managed	MSCS connectivity on the WAN	NFVO interfaces with the WIM for the management of MSCS connectivity on the WAN.	OSS/BSS interfaces with the WIM for the management of MSCS connectivity on the WAN.
	Demarcation point (A) (Case gw.1)	WAN resources	WIM interfaces with the WAN management functions to control connectivity over the WAN resources.	WIM interfaces with the WAN management functions to control connectivity over the WAN resources.
		NFVI-PoP GW resources	WIM interfaces with the NFVI-PoP network gateway management functions to manage the GW resources.	WIM interfaces with the NFVI-PoP network gateway management functions to manage the GW resources.
		NFVI-PoP network resources	WIM interfaces with the NFVI-PoP network management functions to manage the NFVI-PoP network resources.	WIM interfaces with the NFVI-PoP network management functions to manage the NFVI-PoP network resources.



	Type of resources	WIM integration		
		Part of the NFV-MANO framework (case wim.1)	External to the NFV-MANO framework (case wim.2)	
	<b>NFVI-PoP network provider managed</b>  <b>Demarcation point (B)</b>  <b>(Case gw.2)</b>	MSCS connectivity on the WAN	NFVO interfaces with the WIM for the management of WAN connectivity for MSCS.	OSS/BSS interfaces with the WIM for the management of WAN connectivity for MSCS.
		WAN resources	WIM interfaces with the WAN management functions to control connectivity over the WAN resources.	WIM interfaces with the WAN management functions to control connectivity over the WAN resources.
		NFVI-PoP GW resources	NFVO interfaces either directly, or via the VIM, with the NFVI-PoP network gateway management functions to manage the GW resources. See note.	NFVO interfaces either directly, or via the VIM, with the NFVI-PoP network gateway management functions to manage the GW resources. See note.
		NFVI-PoP network resources	VIM interfaces with the NFVI-PoP network management functions to manage the NFVI-PoP network resources.	VIM interfaces with the NFVI-PoP network management functions to manage the NFVI-PoP network resources.
	<b>Co-managed</b>  <b>Demarcation point (A)</b>  <b>(Case gw.3)</b>	MSCS connectivity on the WAN	NFVO interfaces with the WIM for the management of WAN connectivity for MSCS.	OSS/BSS interfaces with the WIM for the management of WAN connectivity for MSCS.
		WAN resources	WIM interfaces with the WAN management functions to control connectivity over the WAN resources.	WIM interfaces with the WAN management functions to control connectivity over the WAN resources.
		NFVI-PoP GW resources	WIM interfaces with the NFVI-PoP network gateway management functions to manage the GW resources.  NFVO can interface either directly, or via the VIM, with the NFVI-PoP network gateway management functions to perform some configuration and/or monitoring related to the GW resources.  See note.	WIM interfaces with the NFVI-PoP network gateway management functions to manage the GW resources.  NFVO can interface either directly, or via the VIM, with the NFVI-PoP network gateway management functions to perform some configuration and/or monitoring related to the GW resources.  See note.
		NFVI-PoP network resources	VIM interfaces with the NFVI-PoP network management functions to manage the NFVI-PoP network resources.	VIM interfaces with the NFVI-PoP network management functions to manage the NFVI-PoP network resources.
NOTE: It is assumed that in both cases, the NFVI-PoP network gateway is logically part of the NFVI-PoP and exposed as "managed resources" to the NFV-MANO. Whether the NFVO interfaces directly with the NFVI-PoP network gateway management functions or via the VIM can depend on the capabilities of the deployed VIM and its network backend solution and it is not in the scope of the present document to specify which mechanism to be supported.				

The set of cases described in table E.1.2-1 influence the types of information to be signalled in between the OSS/BSS and NFVO over the Os-Ma-nfvo reference point, as supported by the NFV-MANO APIs specified in the present document.

### E.1.3 Types of connectivity service endpoint

A Connectivity Service Endpoint (CSE) represents bearer connectivity established in between the NFVI-PoP and the WAN network devices. Such a bearer connectivity may be realized by the network devices in different ways, as described below. In addition, the connectivity service endpoint represents also the "shared context information" regarding the network links available in between the NFVI-PoP and the WAN network devices. Having such a "shared context information" allows that different network providers and their operating support systems understand on what links it is requested to establish the MSCS and multi-site connectivity. As a result:

- The WIM maintains information about the connectivity service endpoints that are enabled on the edge links of the WAN, as specified in clause 8.3.12 of ETSI GS NFV-IFA 032 [i.10].
- The VIM maintains information about the connectivity service endpoints that are enabled allowing to connect in between the NFVI-PoP and the WAN, as specified in clause 8.10.3.2 of ETSI GS NFV-IFA 005 [i.9].

NOTE: This assumes NFVI-PoP network gateway is a managed resource from the VIM's point of view.

The types of connectivity service endpoints for the establishment of the connectivity in between the NFVI-PoP network gateway and the WAN can be mapped to the following forms of enabled connectivity:

- Case cse.A: physical interface, e.g. an Ethernet interface port at a link card in an Ethernet switch chassis (e.g. eth0/0/1).
- Case cse.B: aggregated interface using Link Aggregation Group (LAG) capabilities. It can also be referred as "bundled interface".

Typically, on both physical and aggregated interfaces, logical interfaces can be created which allow segregating the traffic into separate logical channels on a single interface.

Figure E.1.3-1 illustrates the types of connectivity service endpoints and the mapping of these endpoints to the physical or aggregated interfaces that are configured on the respective NFVI-PoP gateway and WAN network devices (i.e. the PE in this case). As the figure shows, the management systems of the respective network devices can map specific interfaces of the network device with the corresponding service connectivity endpoint.

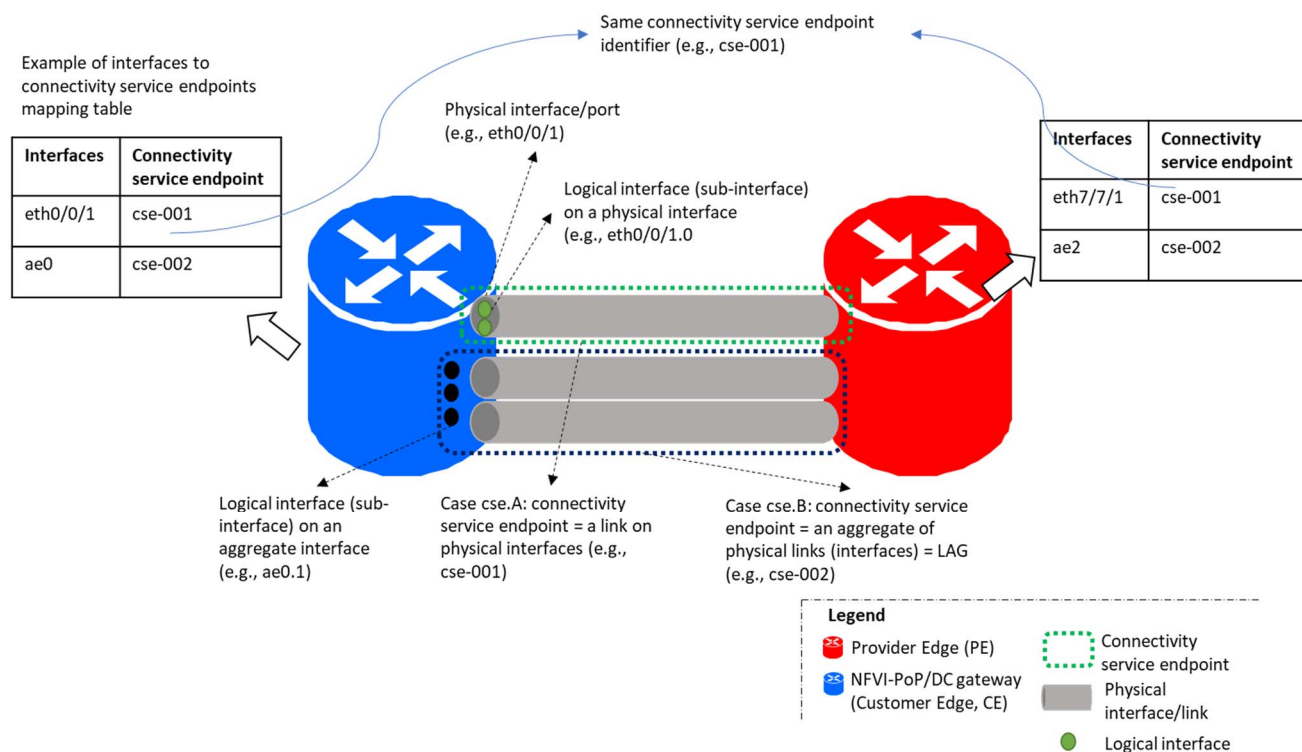


Figure E.1.3-1: Types of connectivity service endpoints

In addition, an NFVI-PoP network gateway:

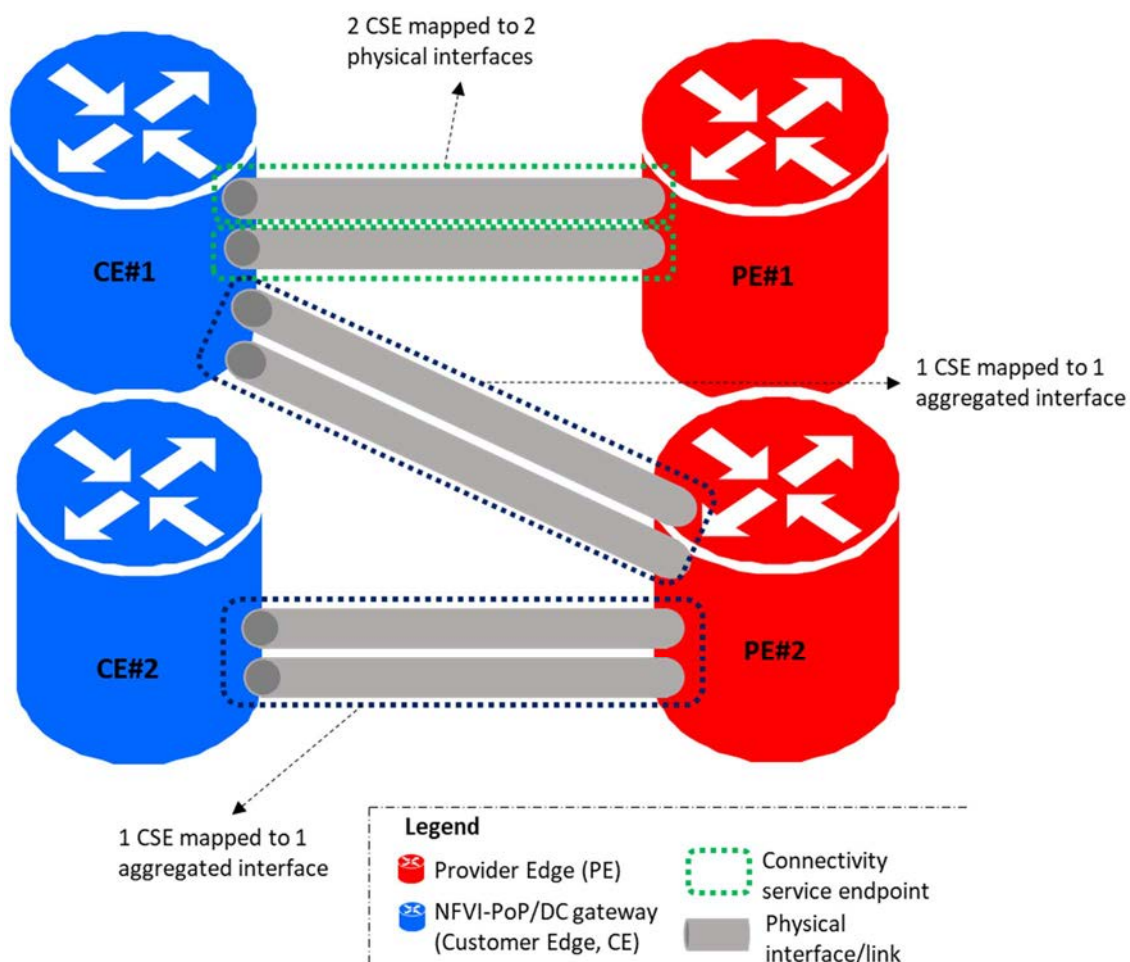
- Can have one or more connectivity service endpoints established towards a single WAN provider edge node.
- Can have connectivity service endpoints towards more than one WAN provider edge node.

Figure E.1.3-2 illustrates the cases related to the multiplicity of connectivity in between the NFVI-PoP network gateways and the WAN network devices (i.e. the PE). In this example:

- The CE#1 has four physical interfaces:
  - two interfaces connect to respective interfaces on the PE#1 and both interfaces are mapped 1:1 to respective connectivity service endpoints; and
  - two other interfaces that connect to PE#2 which are aggregated and there is a 1:1 relationship in between the aggregated interface and the connectivity service endpoint.

In this case, the CE#1 is in "dual-homed" configuration.

- The CE#2 has two physical interfaces, and these interfaces are bundled into an aggregate interface which represent (is mapped to) the connectivity service endpoint.
- The PE#2 is connected to two CEs. These CEs could be part of the same NFVI-PoP (site) or be part of two different sites.



**Figure E.1.3-2: Cases of multiple connections/links in between NFVI-PoP network gateway and WAN network devices**

When a CE is multi-homed, as shown in the example in figure E.1.3-2, it is also possible to establish a single connectivity service endpoint mapped to a LAG involving links connecting to different PE routers. This case assumes that the PE routers are able to discover that they are connected to the same CE and are able to advertise that they are part of the same LAG.

In the case that network equipment in the DC network fabric (e.g. B-leaf switches) acts as CE, similar multi-homing and connectivity service endpoint cases are possible as long as the underlying connectivity in between these devices and the PE follow same patterns as shown in figure E.1.3-2. In this regard, the concept of connectivity service endpoint is not restricted to a specific case of network equipment, as it simply represents the connectivity endpoints in between the NFVI-PoP and the WAN, noting that many different types of equipment in the NFVI-PoP can be qualified as a CE.

---

## E.2 Description and usage guidelines of API attributes

### E.2.1 Description of API attributes

#### E.2.1.1 WanConnectionData

The following clauses provide usage guidelines of the information provided via the "WanConnectionData" data type.

As specified in clause 6.5.2.11, there can be more than one "wanConnectionData" provided in the "InstantiateNsRequest". This supports the use case of providing relevant network information applicable to multiple VL that are expected to span multiple sites. Both NS VL and internal VNF VL are considered as supported by the "nsVirtualLink" and "vnfVirtualLink" attributes. For each of the "wanConnectionData", protocol specific information for connecting to the WAN is provided by the "protocolData", of type "WanConnectionProtocolData".

The attributes present in the "WanConnectionProtocolData" cover different sets of network information:

- "mcsInfo": this attribute and its corresponding referred data type is used to refer and provide to the NFVO information about a pre-provisioned MSCS, if such MSCS already exists.
- "connectivityServiceEndpointConfigDatas": the attribute enables the API consumer to provide information relevant for the configuration of the connectivity service endpoints in order to establish the MSCS endpoint from the NFVI-PoP network gateway perspective. Such an attribute is particularly envisioned to be used in the case that WIM is external to the NFV-MANO and a management model according to demarcation point (B) is applicable as described in clause E.1.2.
- "mcsConfigData": the attribute provides data that enables the NFV-MANO to specialize the MSCS to be created based on specific API consumer requirements.

With regards to the "connectivityServiceEndpointConfigDatas" the information is particularized per VIM (as manager of its corresponding NFVI-PoP) and "connectivity service endpoint". Each "connectivityServiceEndpointConfigData" represents the specific configuration information according to the established MSCS that is required for the connectivity service endpoint. It is assumed that there is link layer reachability in between the NFVI-PoP network gateway and the WAN provider edge node on the WAN side (also referred in some cases as "attachment circuit") which is identifiable via the "connectivityServiceEndpointId" (refer to clause 4.3 of ETSI GS NFV-IFA 032 [i.10] and clause E.1.3 of the present document). A connectivity service endpoint can participate in more than one MSCS, from which a logical MSCS endpoint is then established. Therefore, the full set of configuration parameters to be configured on the NFVI-PoP network gateway for a particular connectivity service endpoint is determined by the whole set of individual "connectivityServiceEndpointConfigData".

When the MSCS is already pre-provisioned, the "mcsInfo" attribute provides information about the identifier, type of MSCS, and existing MSCS endpoints. The NFVO can use such an information to update the runtime information associated to the NS and VNF VLs, such as the resource handles (refer to clauses 6.5.3.53 and 6.5.3.59).

## E.2.1.2 SiteToWanLayer2ProtocolData

The "SiteToWanLayer2ProtocolData" provides Layer 2 protocol specific information needed for the configuration of the NFVI-PoP network gateway to enable the stitching of the intra-site VNs to the MSCS over the WAN. The data type can be used in case that the MSCS has been established to support only Layer 2 multi-site connectivity, or also to define the Layer 2 configuration relevant for enabling Layer 3 multi-site connectivity.

The data type is comprised of four main parts:

- "layer2ConnectionInfo": the attribute provides layer 2 protocol information which is used to determine the type of interface that is to be enabled on the connectivity service endpoint. The following three cases are considered:
  - Case 1: existing layer 2 connectivity of the connectivity service endpoint to be used as is. There are two sub-cases:
    - 1.1) the pre-provisioned connectivity service endpoint is foreseen to be assigned to one single MSCS without multiple VLAN or VXLAN support;
    - 1.2) the pre-provisioned service endpoint is foreseen to be assigned to multiple MSCS reusing VLAN trunking or VXLAN support.
  - Case 2: create logical interfaces on the existing mapped interface of the connectivity service endpoint.
  - Case 3: create an aggregation of connectivity service endpoints and enable logical interfaces on such an aggregated connectivity service endpoint.
- "mtuL2": the attribute is used to convey information about the MTU of the frames that can be forwarded at layer 2.
- "virtualRoutingAndForwarding": depending on the technology used to provision the MSCS, certain L2VPN protocols can be used, which specifically define a MAC-VRF. The attribute allows the information to be provided to the NFVI-PoP network gateway, so the created or existing interface can be assigned as a member to the VPN.
- "forwardingConfig": information indicating what VN(s) in the NFVI-PoP will be forwarded to the (logical) interface(s) on the connectivity service endpoint.

Table E.2.1.2-1 illustrates the use and combination of parameters values to fulfil the different cases of Layer 2 connectivity (see description above of "layer2ConnectionInfo").

**Table E.2.1.2-1: Types of interfaces enabled on the connectivity service endpoint**

Case	Specific sub-attribute setting of "layer2ConnectionInfo"
Case 1.1	connectionType="CSE" interfaceType="PARENT" interfaceTagging="UNTAGGED" encapsulationType="ETH" lagInterfaceData= [leave empty]
Case 1.2	connectionType="CSE" interfaceType="PARENT" interfaceTagging="TAGGED" encapsulationType="VLAN" or "VXLAN" lagInterfaceData= [leave empty]
Case 2	connectionType="CSE" interfaceType="LOGICAL" interfaceTagging="TAGGED" encapsulationType="VLAN" or "VXLAN" lagInterfaceData= [leave empty]
Case 3	connectionType="AGGREGATE_CSE" interfaceType="LOGICAL" interfaceTagging="TAGGED" encapsulationType="VLAN" or "VXLAN" lagInterfaceData= [provide the list of other connectivity service endpoints to aggregate]

### E.2.1.3 SiteToWanLayer3ProtocolData

The "SiteToWanLayer3ProtocolData" provides Layer 3 protocol specific information needed for the configuration of the NFVI-PoP network gateway to enable the stitching of the intra-site VNs to the MSCS over the WAN. In particular, the data type is used for enabling Layer 3 multi-site connectivity.

The data type is comprised of four main parts:

- "routingInfo": the attribute is used to convey the configuration related to the routing activated on the MSCS endpoint. It supports the definition of either static routing entries or use of dynamic routing protocols. Depending on the type of dynamic routing protocol, certain attributes are used and others not.
- "mtuL3": the attribute is used to convey information about the MTU of the frames that can be forwarded at layer 3.
- "virtualRoutingAndForwarding": depending on the technology used to provision the MSCS, certain L3VPN protocols can be used, which specifically define a VRF. The attribute allows the information to be provided to the NFVI-PoP network gateway, so the created or existing interface can be assigned as a member.
- "bfdConfig": information to configure the Bidirectional Forwarding Detection (BFD).

## E.2.2 Use of API attributes based on different management models

### E.2.2.1 Introduction

The following clauses describe the use of the API attributes described in clause E.2.1 in relation to the different NFVI-PoP network gateway management models described in clause E.1.2.

### E.2.2.2 Specific usage for externally managed WAN and NFVI-PoP network provider-managed NFVI-PoP network gateway (case wim.2+gw.2)

In this model, the demarcation of management responsibility is established in between the NFVI-PoP network gateway and the WAN provide edge node, i.e. the PE. In addition, the MSCS is setup by the WIM while being external to the NFV-MANO framework, and the NFV-MANO is responsible through the VIM (or directly with the NFVI-POP network gateway management functions) to stitch the virtualised networks in the NFVI-PoP to the MSCS.

Table E.2.2.2-1 lists and describes the sets of information, as described in clause E.2.1, that can be used to realize certain network configuration.

**Table E.2.2.2-1: Sets of information applicable to externally managed WAN and NFVI-PoP network provider-managed NFVI-PoP network gateway (case wim.2+gw.2)**

Purpose	Data type	Attribute	Additional remarks
Information about the MSCS	WanConnectionProtocolData	mscsInfo	The "mscsInfo" provides information to NFV-MANO about the pre-provisioned MSCS, so that relevant NS/VNF runtime information can be updated.
Configuration for NFVI-PoP network gateway to enable the network connectivity stitching	WanConnectionProtocolData	connectivityServiceEndpointConfigDatas	Used to identify the NFVI-PoP network gateway providing the referred connectivity service endpoints, and the relevant L2 and L3 configuration (see also other attributes in table E.2.2.2-1).
Layer 2 connectivity stitching	SiteToWanLayer2ProtocolData	layer2ConnectionInfo	Used to determine the type of connection to enable on the connectivity service endpoint according to the connectivity towards the MSCS.
		mtuL2	See description of the attribute.
		virtualRoutingAndForwarding	To configure the MAC-VRF information, in case the multi-site connectivity is established at layer 2.
		forwardingConfig	To describe the forwarding configuration at layer 2, in case the multi-site connectivity is established at layer 2.
Layer 3 connectivity stitching	SiteToWanLayer3ProtocolData	logicalInterfaceIpAddress	See description of the attribute.
		routingInfo	See description of the attribute.
		mtuL3	See description of the attribute.
		virtualRoutingAndForwarding	To configure the VRF information, in case the multi-site connectivity is established at layer 3.

### E.2.2.3 Specific usage for externally managed WAN and WAN provider-managed NFVI-PoP network gateway (case wim.2+gw.1)

In this model, the demarcation of management responsibility is established in between the NFVI-PoP internal networks and the NFVI-PoP network gateway, i.e. the CE or DC-GW. In addition, the MSCS is setup by the WIM while being external to the NFV-MANO framework, and the WIM is also responsible to stitch the virtualised networks in the NFVI-PoP to the MSCS.

Table E.2.2.3-1 lists and describes the sets of information, as described in clause E.2.1, that can be used to realize certain network configuration.

**Table E.2.2.3-1: Sets of information applicable to externally managed WAN and WAN provider-managed NFVI-PoP network gateway (case wim.2+gw.1)**

Purpose	Data type	Attribute	Additional remarks
Information about the MSCS	WanConnectionProtocolData	mscsInfo	The "mscsInfo" provides information to NFV-MANO about the pre-provisioned MSCS, so that relevant NS/VNF runtime information can be updated.
Configuration for NFVI-PoP network gateway to enable the network connectivity stitching	WanConnectionProtocolData	connectivityServiceEndpointConfigDatas	To identify the NFVI-PoP network gateway providing the referred connectivity service endpoints, and the relevant L2 and L3 configuration (see also other attributes in table E.2.2.3-1).
Layer 2 connectivity stitching	SiteToWanLayer2ProtocolData	forwardingConfig	To describe the forwarding configuration at layer 2, in case the multi-site connectivity is established at layer 2.
Layer 3 connectivity stitching	SiteToWanLayer3ProtocolData	routingInfo ->routeMapsDistribution	To describe the routes that can be distributed, in case the multi-site connectivity is established at layer 3.

#### E.2.2.4 Specific usage for NFV-MANO managed WAN and NFVI-PoP network provider-managed NFVI-PoP network gateway (case wim.1+gw.2)

In this model, the demarcation of management responsibility is established in between the NFVI-PoP network gateway and the WAN provider edge node, i.e. the PE. In addition, the NFV-MANO is responsible through the WIM to setup the MSCS and through the VIM (or directly with the NFVI-POP network gateway management functions) to stitch the virtualised networks in the NFVI-PoP to the MSCS.

Table E.2.2.4-1 lists and describes the sets of information, as described in clause E.2.1, that can be used to realize certain network configuration.

**Table E.2.2.4-1: Sets of information applicable to NFV-MANO managed WAN and NFVI-PoP network provider-managed NFVI-PoP network gateway (case wim.1+gw.2)**

Purpose	Data type	Attribute	Additional remarks
Requirements and configuration data for the MSCS	WanConnectionProtocolData	mscsConfigData	The "mscsConfigData" provides information to the NFVO to determine some of the characteristics of the MSCS to be provisioned and additional resiliency requirements.

In this model, the NFV-MANO derives most of the information for managing the MSCS and the stitching of the virtualised networks in the NFVI-PoP with the MSCS from other information provided in the API as well as the VNFD and NSD:

- Affinity and anti-affinity information in the descriptors and location constraints information available on the API requests, in addition to resource capacity information managed by the NFVO, can be used by the NFVO to derive the target NFVI-PoPs and their associated connectivity service endpoints over which to enable multi-site connectivity.
- The set of virtualised networks in the NFVI-PoP that need to be stitched/associated to multi-site network connectivity can be determined by the NFVO when coordinating with the VIM, VNFM and the WIM with the actual requests to create the virtualised resources for the NS/VNF VLs and the MSCS.



- In case of layer 3 connectivity, the routing map distribution information can be determined by the NFVO by inspecting the IP information associated to the external connection points of the NS constituents deployed across the different NFVI-PoPs.

Other network configuration information that is still needed for enabling the multi-site connectivity can be determined according to network operator policies and is only used within the NFV-MANO framework (including the WIM), e.g. virtual routing and forwarding configuration.

### E.2.2.5 Specific usage for NFV-MANO managed WAN and WAN provider-managed NFVI-PoP network gateway (case wim.1+gw.1)

In this model, the demarcation of management responsibility is established in between the NFVI-PoP internal networks and the NFVI-PoP network gateway, i.e. the CE or DC-GW. In addition, in this case, the NFV-MANO is responsible through the WIM to setup the multi-site connectivity and the stitching of the virtualised networks in the NFVI-PoP to the MSCS.

Table E.2.2.5-1 lists the sets of information that can be used to realize certain network configuration.

**Table E.2.2.5-1: Sets of information applicable to NFV-MANO managed WAN and WAN provider-managed NFVI-PoP network gateway (case wim.1+gw.1)**

Purpose	Data type	Attribute	Additional remarks
Requirements and configuration data for the MSCS	WanConnectionProtocolData	mscsConfigData	The "mscsConfigData" provides information to the NFVO to determine some of the characteristics of the MSCS to be provisioned and additional resiliency requirements.

In this model, the NFV-MANO derives most of the information for managing the MSCS and the stitching of the virtualised networks in the NFVI-PoP with the MSCS from other information provided in the API as well as the VNFD and NSD:

- Affinity and anti-affinity information in the descriptors and location constraints information available on the API requests, in addition to resource capacity information managed by the NFVO, can be used by the NFVO to derive the target NFVI-PoPs and their associated connectivity service endpoints over which to enable multi-site connectivity.
- The set of virtualised networks in the NFVI-PoP that need to be stitched/associated to multi-site network connectivity can be determined by the NFVO when coordinating with the VIM, VNFM and the WIM with the actual requests to create the virtualised resources for the NS/VNF VLs and the MSCS.
- In case of layer 3 connectivity, the routing map distribution information can be determined by the NFVO by the IP information associated to the external connection points of the NS constituents deployed across the different NFVI-PoPs.

Other network configuration information that is still needed for enabling the multi-site connectivity can be determined according to network operator policies and is only used within the NFV-MANO framework (including the WIM), e.g. virtual routing and forwarding configuration.

### E.2.2.6 Specific usage for externally managed WAN and co-managed NFVI-PoP network gateway (case wim.2+gw.3)

In this model, the demarcation of management responsibility is established in between the NFVI-PoP internal networks and the NFVI-PoP network gateway, i.e. the CE or DC-GW, but there is co-management of NFVI-PoP network gateway resources in between the NFVO/VIM and the WIM. In addition, the MSCS is setup by the WIM while being external to the NFV-MANO framework, and the NFVO/VIM and WIM co-manage the stitching of the virtualised networks in the NFVI-PoP to the MSCS.

Table E.2.2.6-1 lists and describes the sets of information, as described in clause E.2.1, that can be used to realize certain network configuration.

**Table E.2.2.6-1: Sets of information applicable to externally managed WAN and co-managed NFVI-PoP network gateway (case wim.2+gw.3)**

Purpose	Data type	Attribute	Additional remarks
Information about the MSCS	WanConnectionProtocolData	mcsInfo	The "mcsInfo" provides information to NFV-MANO about the pre-provisioned MSCS, so that relevant NS/VNF runtime information can be updated.
Layer 2 connectivity stitching	SiteToWanLayer2ProtocolData	forwardingConfig	To describe the forwarding configuration at layer 2, in case the multi-site connectivity is established at layer 2.
Layer 3 connectivity stitching	SiteToWanLayer3ProtocolData	routingInfo ->routeMapsDistribution	To describe the routes that can be distributed, in case the multi-site connectivity is established at layer 3.

### E.2.2.7 Specific usage for NFV-MANO managed WAN and co-managed NFVI-PoP network gateway (case wim.1+gw.3)

In this model, the demarcation of management responsibility is established in between the NFVI-PoP internal networks and the NFVI-PoP network gateway, i.e. the CE or DC-GW, but there is co-management of NFVI-PoP network gateway resources in between the NFVO/VIM and the WIM. In addition, the NFV-MANO is responsible through the WIM to setup the MSCS, and the NFVO/VIM and WIM co-manage the stitching of the virtualised networks in the NFVI-PoP to the MSCS.

Table E.2.2.7-1 lists and describes the sets of information, as described in clause E.2.1, that can be used to realize certain network configuration.

**Table E.2.2.7-1: Sets of information applicable to NFV-MANO managed WAN and co-managed NFVI-PoP network gateway (case wim.1+gw.3)**

Purpose	Data type	Attribute	Additional remarks
Requirements and configuration data for the MSCS	WanConnectionProtocolData	mcsConfigData	The "mcsConfigData" provides information to the NFVO to determine some of the characteristics of the MSCS to be provisioned and additional resiliency requirements.

In this model, the NFV-MANO derives most of the information for managing the MSCS and the stitching of the virtualised networks in the NFVI-PoP with the MSCS from other information provided in the API as well as the VNFD and NSD:

- Affinity and anti-affinity information in the descriptors and location constraints information available on the API requests, in addition to resource capacity information managed by the NFVO, can be used by the NFVO to derive the target NFVI-PoPs and their associated connectivity service endpoints over which to enable multi-site connectivity.
- The set of virtualised networks in the NFVI-PoP that need to be stitched/associated to multi-site network connectivity can be determined by the NFVO when coordinating with the VIM, VNFM and the WIM with the actual requests to create the virtualised resources for the NS/VNF VLS and the MSCS.
- In case of layer 3 connectivity, the routing map distribution information can be determined by the NFVO by the IP information associated to the external connection points of the NS constituents deployed across the different NFVI-PoPs.

Other network configuration information that is still needed for enabling the multi-site connectivity can be determined according to network operator policies and is only used within the NFV-MANO framework (including the WIM), e.g. virtual routing and forwarding configuration.

---

## Annex F (informative): Bibliography

- IETF RFC 2818: "HTTP Over TLS".
- IETF RFC 3339: "Date and Time on the Internet: Timestamps".
- IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".
- IETF RFC 4291: "IP Version 6 Addressing Architecture".
- IETF RFC 4632: "Classless Inter-Domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan".
- IETF RFC 4918: "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)".
- IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".
- IETF RFC 6585: "Additional HTTP Status Codes".
- IETF RFC 6749: "The OAuth 2.0 Authorization Framework".
- IETF RFC 6750: "The OAuth 2.0 Authorization Framework: Bearer Token Usage".
- IETF RFC 8259: "The JavaScript Object Notation (JSON) Data Interchange Format".
- IETF RFC 7231: "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content".
- IETF RFC 7235: "Hypertext Transfer Protocol (HTTP/1.1): Authentication".
- IETF RFC 7617: "The 'Basic' HTTP Authentication Scheme".
- IETF RFC 7807: "Problem Details for HTTP APIs".
- IETF RFC 6901: "JavaScript Object Notation (JSON) Pointer".
- IETF RFC 4229: "HTTP Header Field Registrations".
- IETF RFC 8288: "Web Linking".
- Semantic Versioning 2.0.0.
- ETSI TS 133 310: "Universal Mobile Telecommunications System (UMTS); LTE; Network Domain Security (NDS); Authentication Framework (AF) (3GPP TS 33.310)".
- Hypertext Transfer Protocol (HTTP) Status Code Registry at IANA.

## Annex G (informative): Change History

Date	Version	Information about changes
December 2016	0.0.1	Initial version based on contributions that were agreed at the NFVSOL#15 meeting <ul style="list-style-type: none"> <li>- NFVSOL(16)000169_SOL005_Scope_Statement</li> <li>- NFVSOL(16)000171_SOL005_Proposed_Table_of_Contents</li> </ul>
February 2017	0.1.0	Version 0.1.0 based on contributions that were agreed at the NFVSOL#20 meeting <ul style="list-style-type: none"> <li>- NFVSOL(17)000061_SOL005_URI_structure_and_supported_content_formats</li> <li>- NFVSOL(17)000064R1_SOL002_SOL003_SOL005_Labeling_of_API_names</li> <li>- NFVSOL(17)000095R1_SOL005_Error_reporting</li> <li>- NFVSOL(17)000106R1_Conventions_document_NFVSOL(17)000050_swagger_representation_of_the_API</li> <li>- NFVSOL(17)000107R1_SOL005_Common_procedures</li> <li>- NFVSOL(17)000111_SOL003_Conventions_move_Resource_structure_up_in_the_TOC</li> </ul>
March 2017	0.2.0	Version 0.2.0 based on contributions that were agreed at the NFVSOL#22 meeting <ul style="list-style-type: none"> <li>- NFVSOL(17)000196R1_SOL005_Add_SOL003_Normative_Reference</li> <li>- NFVSOL(17)000198_SOL005-Clause_4.1_Overview</li> <li>- NFVSOL(17)000199_SOL002_SOL003_Renaming_attribute_selectors</li> <li>- NFVSOL(17)000200_SOL002_SOL003_Attribute_filter_equality</li> <li>- NFVSOL(17)000179_SOL005_Simple_Data_Types</li> <li>- NFVSOL(17)000138_SOL005_NS_LCM_Description_Clause_6.1</li> <li>- NFVSOL(17)000142R1_SOL005_NS_LCM_Resource_Structure_and_Methods_Clause_6.2</li> <li>- NFVSOL(17)000123R2_SOL005_Flow_of_the_Creation_of_a_NS_Instance_Resource</li> <li>- NFVSOL(17)000129R1_SOL005_Flow_of_the_Deletion_of_a_NS_Instance_Resource</li> <li>- NFVSOL(17)000130_SOL005_Flow_of_the_Get_Operations_Status_Operation</li> <li>- NFVSOL(17)000137_SOL005_Flow_of_Managing_Subscriptions_to_Notifications_Related_to_NS_Lifecycle_Management</li> <li>- NFVSOL(17)000136_SOL005_Flow_of_Sending_Notifications_Related_to_NS_Lifecycle_management</li> <li>- NFVSOL(17)000156R2_SOL005_NS_Lifecycle_Change_Resource_Definitions_Methods_and_Data_Types</li> <li>- NFVSOL(17)000160_SOL005_NS_Lifecycle_Management_Resource_Definitions_Methods_and_Data_Types</li> </ul>
April 2017	0.3.0	Version 0.3.0 based on contributions that were agreed at the NFVSOL#26 meeting <ul style="list-style-type: none"> <li>- NFVSOL(17)000263_SOL005_Instantiate_NS_Lifecycle_Management_Resource_Definition_Methods_and_Data_Types</li> <li>- NFVSOL(17)000265_SOL005_Heal_NS_Lifecycle_Management_Resource_Definition_Methods_and_Data_Types</li> </ul>
May 2017	0.4.0	Version 0.4.0 based on contributions that were agreed at the NFVSOL#28 meeting <ul style="list-style-type: none"> <li>- NFVSOL(17)000264R1_SOL005_Update_NS_Lifecycle_Management_Resource_Methods_and_Data_Types</li> <li>- NFVSOL(17)000266_SOL005_Scale_NS_Operation_Resource_Definition_Methods_and_Data_Types</li> <li>- NFVSOL(17)000267_SOL005_Terminate_NS_Operation_Resource_Definition_Methods_and_Data_Type</li> <li>- NFVSOL(17)000268_SOL005_Merge_NS_LCCN_interface_into_the_NS_LCM_interface</li> <li>- NFVSOL(17)000348R4_SOL005_NSD_Management_Interface</li> <li>- NFVSOL(17)000378_SOL005_NS_Lifecycle_Change_Occurrence_Resource_Definitions_Methods_and_Data_Types</li> <li>- NFVSOL(17)000383_SOL005_Flow_of_NS_Lifecycle_Management_Operations_Triggred_by_Task_Resources</li> <li>- NFVSOL(17)000385R1_SOL005_Individual_NS_Descriptor_Resource_Descriptor_Methods_and_Data_Types</li> <li>- NFVSOL(17)000390R1_SOL005_Handling_of_Errors_During_NS_Lifecycle_Management_Operations</li> </ul>

Date	Version	Information about changes
June 2017	0.5.0	<p>Version 0.5.0 based on contributions that were agreed at the NFVSOL#30 and NFVSOL#31 meetings</p> <ul style="list-style-type: none"> <li>- NFVSOL(17)000371R1_SOL005:_Data_Type_NsInstance</li> <li>- NFVSOL(17)000411R1_SOL005:_Clause_6.6_Update,_"Handling_of_Errors_During_NS_Lifecycle_Management"</li> <li>- NFVSOL(17)000420R1_SOL005:_Update_ScaleVnfData_and_ScaleByStepData_Data_Types_consistent_with_NFVIFA(17)000382R1</li> <li>- NFVSOL(17)000421R1_SOL005:_Update_"additionalParam..."_attribute_names, KeyValuePair data type_and_its_cardinality_consistent_with_SOL003</li> <li>- NFVSOL(17)000422_SOL005:_Update_AffinityOrAntiAffinityRule_Data_Type_Conconsistent_with_NFVIFA(17)000534</li> <li>- NFVSOL(17)000424_SOL005:_Modify_VNF_Configuration_in_Update_NS_operation_consistent_with_NFVIFA(17)000527</li> <li>- NFVSOL(17)000427R2_SOL005:_Update_NSD_Interface_Resource_Tree,_Resources,_and_Methods_(Clause_5.2)</li> <li>- NFVSOL(17)000428_SOL005:_ParamsForVnf_and_Resolution_of_the_Associated_Editor's_Note</li> <li>- NFVSOL(17)000431R3_SOL005:_Error_Handling_for_NS_Lifecycle_operation,_Resource_Definition_and_Methods</li> <li>- NFVSOL(17)000432R2_SOL005:_Update_Flow_of_Error_Handling_for_NS_LCM_Operations</li> <li>- NFVSOL(17)000433R3_SOL005:_Update_"address"_attribute's_data_type_in_PnfExtCplInfo_data_type</li> <li>- NFVSOL(17)000434R3_SOL005:_Update_LocationConstraints_data_type_in_Instantiate_NS_operation</li> <li>- NFVSOL(17)000436R2_SOL005:_Filter_design_for_NS_Instances</li> <li>- NFVSOL(17)000438R2_SOL005_VNF_Package_management_interface_-_resource_structure_and_methods</li> <li>- NFVSOL(17)000439R3_SOL005_VNF_package_management_interface_-_resources</li> <li>- NFVSOL(17)000440R3_SOL005:_Updates_to_the_Lifecycle_Change_Notifications_Filter</li> <li>- NFVSOL(17)000441R1_SOL005:_Authorizations_of_API_requests_and_notifications</li> <li>- NFVSOL(17)000446R1_SOL005:_Update_Clause_4.3_"Common_Procedures"_Consistent_with_SOL003</li> <li>- NFVSOL(17)000451_SOL005:_Correct_various_references_to_NS_Lifecycle_operation_occurrences</li> <li>- NFVSOL(17)000464_SOL005_-_Modifications_on_sequence_diagrams_in_clauses_5.3.2_and_5.3.3</li> <li>- NFVSOL(17)000466_SOL005:_Add_IdentifierInVim_simple_data_type</li> </ul>
August 2017	0.6.0	<p>Version 0.6.0 based on contributions that were agreed at the NFVSOL#34 and NFVSOL#35,meetings</p> <ul style="list-style-type: none"> <li>- NFVSOL(17)00426R6_SOL005:_Changes_to_NSD_Resources,_Methods,_and_Data_Types</li> <li>- NFVSOL(17)00493_SOL005:_Update_Clause_3,_Abbreviations</li> <li>- NFVSOL(17)000494R1_SOL005:_NS_PM_Interface_description,_resource_structure,_and_methods</li> <li>- NFVSOL(17)000508_SOL005:_Remove_Editor's_Note_in_Clause_6.5.3.8</li> <li>- NFVSOL(17)000509_SOL005:_Remove_Editor's_Note_in_Clauses_5.5.3.1, 5.5.4.1, and 5.5.4.2</li> <li>- NFVSOL(17)000510R1_SOL005:_NS_Performance_Management_Interface_resources_and_data_model</li> <li>- NFVSOL(17)000511R1_SOL005:_VNF_package_management_interface_flows</li> <li>- NFVSOL(17)000512R1_SOL005:_VNF_package_management_interface_data_types</li> <li>- NFVSOL(17)000513R2_SOL005:_NSD_Management_Interface_operations,_resource_structure,_and_methods</li> <li>- NFVSOL(17)000514R1_SOL005:_NS_LCM_Updates</li> </ul>

Date	Version	Information about changes
September 2017	0.7.0	Version 0.7.0 based on contributions that were agreed at the NFVSOL#36 and NFVSOL#37 meetings <ul style="list-style-type: none"> <li>- NFVSOL(17)000386R3_SOL005:_PNFD_resources,_methods,_and_data_types</li> <li>- NFVSOL(17)00478R1_SOL005:_VL_and_CP_consistency_(mirror_of_403r3_and_423)</li> <li>- NFVSOL(17)000519R4_SOL005:_NS_fault_management_interface</li> <li>- NFVSOL(17)000520R1_SOL005:_New_Annex_B_Mapping_operations_to_protocol_elements</li> <li>- NFVSOL(17)000524R1_SOL005:_Refactoring_of_NSD_management_interface</li> <li>- NFVSOL(17)000526R1_SOL005:_Definition_of_the_HealNsData_data_type (clause_6.5.3.33)</li> <li>- NFVSOL(17)000529_SOL005:_Apply_the_agreed_design_for_VNF_package_onboarding_operation</li> <li>- NFVSOL(17)000530R1_SOL005:_Remaining_VNF_package_management_interface_flows</li> <li>- NFVSOL(17)000531R1_SOL005:_Remaining_resource_design_for_VNF_package_management_interface</li> <li>- NFVSOL(17)000532R1_SOL005:_Filters_and_selectors_for_VNF_package_management_interface</li> <li>- NFVSOL(17)000533R2_SOL005:_Additional_data_model_for_VNF_package_management_interface</li> <li>- NFVSOL(17)000534_SOL005:_VNF_package_management_interface_data_type_align_with_SOL003</li> <li>- NFVSOL(17)000554_SOL005:_Align_with_IFA013_on_vimId</li> </ul>
October 2017	0.8.0	Version 0.8.0 based on contributions that were agreed at the NFVSOL#38 meeting <ul style="list-style-type: none"> <li>- NFVSOL(17)000521_SOL005:_NS_performance_management_interface_sequence_diagrams</li> <li>- NFVSOL(17)000558R1_SOL005:_NSD_management_interface_-_notification_and_state_diagram</li> <li>- NFVSOL(17)000573_SOL005:_NSD_management_interface_subscription_resources</li> </ul>
November 2017	0.9.0	Version 0.9.0 based on contributions that were agreed at the NFVSOL#40 and NFVSOL#41 meetings <ul style="list-style-type: none"> <li>- NFVSOL(17)000535r2_SOL005:_Modification_of_data_types_due_to_the_separation_of_on-boarding_VNF_package_operation</li> <li>- NFVSOL(17)000546R3_SOL005:_Define_the_nfpRule_attribute</li> <li>- NFVSOL(17)000578R1_SOL005:_Refactor_PNFD_management</li> <li>- NFVSOL(17)000579_SOL005:_Annex_B.5_-_Add_NS_fault_management_mapping_operations_to_protocol_elements</li> <li>- NFVSOL(17)000604r1_SOL005:_NS_LCM_interface_-_edit_note_fix</li> <li>- NFVSOL(17)000607r1_SOL005:_NSD_management_interface_-_consistency_fix</li> <li>- NFVSOL(17)000608r1_SOL005:_NSD_management_interface_-_partial_download</li> <li>- NFVSOL(17)000610r1_SOL005:_NS_LCM_interface_-_edit_note_fix2</li> <li>- NFVSOL(17)000613_SOL005:_VNF_package_management_interface_delete_VNF_package_flow</li> <li>- NFVSOL(17)000623R1_SOL005:_Resolve_editor's_note_in_clause_6.5.3.33_(HealNsData)</li> <li>- NFVSOL(17)000625_SOL005:_Remove_"pnfdName"-related_editor's_note_in_clause_5.5.2.5</li> <li>- NFVSOL(17)000627r1_SOL005:_Use_of_verbal_forms_for_the_expression_of_provisions</li> <li>- NFVSOL(17)000628R1_SOL005:_Miscellaneous technical improvements</li> <li>- NFVSOL(17)000640_SOL005:_Update_sequence_diagram_for_the_get_operations_status_operations</li> </ul>
November 2017	0.9.1	Clean-up done by <b>editHelp!</b> E-mail: <a href="mailto:edithelp@etsi.org">mailto:edithelp@etsi.org</a>

Date	Version	Information about changes
November 2017	0.10.0	<p>Version 0.10.0 based on contributions that were agreed at the NFVSOL#42 and NFVSOL#43 meetings and during Email Approval (EA) resulting from the NFVSOL#41 meeting</p> <ul style="list-style-type: none"> <li>- NFVSOL(17)000612R2_SOL005:_VNF_package_management_interface_refactoring_operations_and_data_types</li> <li>- NFVSOL(17)000632R2_SOL005_-_Editorial_changes</li> <li>- NFVSOL(17)000633R3_SOL005:_Sequence_diagram_for_the_deletion_of_an_individual_PNF_descriptor_resource</li> <li>- NFVSOL(17)000639R2_SOL005:_Sequence_diagram_for_the_deletion_of_an_individual_NS_descriptor_resource</li> <li>- NFVSOL(17)000645R1_SOL005:_Adding_405_response</li> <li>- NFVSOL(17)000648_SOL005:_Addition_of_the_notes_for_identifier_in_the_resource_URI</li> <li>- NFVSOL(17)000657_SOL005:_Flow_of_the_creation_of_an_individual_NS_descriptor_resource</li> <li>- NFVSOL(17)000658R1_SOL005:_Flow_of_the_creation_of_an_individual_PNF_descriptor_resource</li> <li>- NFVSOL(17)000659_SOL005:_Resolution_of_Rapporteur's_Notes_in_Annex_A</li> <li>- NFVSOL(17)000660_SOL005_Flow_of_the_querying_reading_of_NS_descriptor_resources</li> <li>- NFVSOL(17)000662_SOL005_Flow_of_the_querying_reading_of_PNF_descriptor_resources</li> <li>- NFVSOL(17)000664_SOL005_Proposed_resolution_of_clause_4_editor's_note</li> <li>- NFVSOL(17)000665_SOL005_Proposed_resolution_of_clause_4.2_editor's_note</li> <li>- NFVSOL(17)000678R1_SOL005_Update_NFP_related_data_types</li> <li>- NFVSOL(17)000679_SOL005_Add_VNF_package_state_model_to_annex_C</li> <li>- NFVSOL(17)000682_SOL005_Flow_of_the_update_of_an_individual_NS_descriptor_resource</li> </ul> <p>Version 0.10.0 also reflects additional clean-up done by <b>editHelp!</b></p> <ul style="list-style-type: none"> <li>- E-mail: <a href="mailto:edithelp@etsi.org">mailto:edithelp@etsi.org</a></li> </ul>
December 2017	0.11.0	<p>Version 0.11.0 based on contributions that were agreed at the NFVSOL#45 meeting</p> <ul style="list-style-type: none"> <li>- NFVSOL(17)000606R3_SOL005:_Update_data_type_NsLcmOpOcc_and_NsLcmOperationOccurrenceNotification</li> <li>- NFVSOL(17)000609R3_SOL005:_NS_LCM_interface_-_Error_handling_enhancement</li> <li>- NFVSOL(17)000614R3_SOL005:_VNF_package_management_interface_upload_VNF_content_through_external_link</li> <li>- NFVSOL(17)000649R2:_SOL005:_6.5.3.2_OperateVnfData</li> <li>- NFVSOL(17)000672R2_SOL005:_Resolution_of_editor's_note_on_structure_of_theNSD_zip_file_in_clause_5.4.4.3.2</li> <li>- NFVSOL(17)000680R2:_SOL005:_Update_ResourceHandle_datatype</li> <li>- NFVSOL(17)000683R2_SOL005:_Remove_pnfdInfold_and_related_editor's_note_from_clause_5.5.2.6</li> <li>- NFVSOL(17)000701R1_SOL005:_NS_LCM_interface_-_Sequence_diagram_for_continue_operation</li> <li>- NFVSOL(17)000702R1_SOL005:_Update_the_NSd_state_diagram_in_annex</li> <li>- NFVSOL(17)000703R1_SOL005:_NS_LCL_interface_-_resolve_the_editor's_note_about_resource_changes_in_NsLcmOperationOccurrenceNotification</li> <li>- NFVSOL(17)000704R1_SOL005:_Resolve_the_editor's_notes_on_NS_PM_interface</li> <li>- NFVSOL(17)000705_SOL005:_Resolve_the_editor's_notes_on_ExtVirtualLinkData_and_ExtManagedVirtualLinkData</li> <li>- NFVSOL(17)000707R2_SOL005:_Sequence_diagram_for_the_uploading_of_NSd_content</li> <li>- NFVSOL(17)000708R1_SOL005:_Sequence_diagram_for_the_uploading_of_PNFd_content</li> <li>- NFVSOL(17)000709R1_SOL005:_Resolution_of_editor's_note_on_the_NsLcmOpOcc_data_type_in_clause_6.4.9.3.2</li> <li>- NFVSOL(17)000710_SOL005:_Resolution_of_editor's_note_in_clause_5.4.4.3.3_on_partial/chunking/resumable_upload</li> <li>- NFVSOL(17)000712R1:_SOL005_-_Additional_fields_and_values_for_NfpRules</li> <li>- NFVSOL(17)000713_SOL005_4.2_Consistency_of_URI_and_OAuth</li> <li>- NFVSOL(17)000714R1:_SOL005:_Resolve_the_editor's_notes_on_ExtLinkPort_and_NsLinkPort</li> <li>- NFVSOL(17)000716R1_SOL005:_Double_subscriptions_for_notifications</li> </ul>



Date	Version	Information about changes
		<ul style="list-style-type: none"> <li>- NFVSOL(17)000720_SOL005:_Resolution_of_two_editor's_notes_on_pnfdInvarian tId</li> <li>- NFVSOL(17)000721R1_SOL005:_Sequences_of_responses_and_notifications</li> <li>- NFVSOL(17)000727_SOL005:_Flow_of_the_fetching_of_the_content_of_a_NSD</li> <li>- NFVSOL(17)000728_SOL005:_Flow_of_the_fetching_of_the_content_of_a_PNFD</li> <li>- NFVSOL(17)000729_SOL005:_Update_to_the_flow_of_the_creation_of_the_indivi dual_NS_descriptor_resource</li> <li>- NFVSOL(17)000731R1_SOL005:_Fix_description_of_unsupported_method_for_n otification_endpoint</li> <li>- NFVSOL(17)000732R1_SOL005:_Resolve_editor's_note_on_checksum</li> <li>- NFVSOL(17)000738_SOL005_Add_Update_PNFD Info_operation_to_the_list_of_NSD_management_interface_operations_in_clause 5.1</li> <li>- NFVSOL(17)000741_SOL005:_Fix_the_inconsistency_related_to_NS_LCM_ope ration_state</li> <li>- NFVSOL(17)000742_SOL005:_Add_PNFD_Notifications</li> <li>- NFVSOL(17)000743R1_SOL005:_Adding_normative_category_to_resource_and_ methods_Tables</li> <li>- NFVSOL(17)000744_SOL005:_Align_PkgmNotificationsFilter_with_VnflInstanceSu bscriptionFilter</li> <li>- NFVSOL(17)000748R1_SOL005:_Add_NS_change_notification_on_NS_LCM_inte rface</li> <li>- NFVSOL(17)000750:_SOL005:_Add_error_code_for_fetching_package_content,_ vnfd_and_artifact_operations</li> <li>- NFVSOL(17)000755:_SOL005:_Remove_redundant_description_of_vnfConfigurab leProperties</li> <li>- NFVSOL(17)000759R2_SOL005_Add_ChangeVnfFlavourData</li> <li>- NFVSOL(17)000760_SOL005_6.5.3.2_Add_changedInfo_to AffectedVnf_data_type</li> <li>- NFVSOL(17)000762R1_SOL005:_Authorization_of_API_requests_and_notifications</li> </ul>
December 2017	0.12.0	<p>Version 0.12.0 based on contributions that were agreed at the NFVSOL#46 meeting and during Email Approval (EA) following the NFVSOL#45 meeting</p> <ul style="list-style-type: none"> <li>- NFVSOL(17)000603R1_SOL005:_NS_LCM_interface_-_network_address</li> <li>- NFVSOL(17)000745_SOL005:_Change_the_name_of_the "ScaleInfo" data type to "VnfScaleInfo"</li> <li>- NFVSOL(17)000747R2_SOL005:_Add_and_modify_PNF_on_NS_LCM_interface</li> <li>- NFVSOL(17)000752_SOL005:_Change_"pnfdInfoStateType"_to_"pnfdOnboarding StateType"_to_resolve_editor's_note_in_clause_5.5.4.6</li> <li>- NFVSOL(17)000761_SOL005:_Resolve_NsInstanceSubscriptionFilter_editor's_not e_(Issue_Gen.3)</li> <li>- NFVSOL(17)000763_SOL005:_Align_normative_statements_in_trigger_conditions _mirror_734</li> <li>- NFVSOL(17)000767_SOL005:_Authorization_method_negotiation</li> <li>- NFVSOL(17)000769_SOL005:_Annex_A.6_operation_name_and_resource_URI alignment</li> <li>- NFVSOL(17)000770_SOL005:_Add/remove_notes_about_race_conditions</li> <li>- NFVSOL(17)000772_SOL005:_Add_get_method_support_on_the_notification_end point_resource_in_the_NSD_and_NS_LCM_interfaces</li> <li>- NFVSOL(17)000776_SOL005:_Alignment_of_timeStamp_attribute_name</li> <li>- NFVSOL(17)000777_SOL005:_Change "NsLinkPort" to "NsLinkPortInfo"</li> <li>- NFVSOL(17)000779_SOL005:_Precondition_for_VNF_package_deletion</li> <li>- NFVSOL(17)000784_SOL005:_Add_a_pointer_to_annex_B_for_the_NSD_state_ model_in_NsdInfo</li> </ul>
February 2018	2.4.1	Publication
March 2018	2.4.2	<p>Version 2.4.2 based on contributions that were agreed at the NFVSOL#55 meeting</p> <ul style="list-style-type: none"> <li>- NFVSOL(18)000047_SOL005ed251_API_authorization_clarification</li> <li>- NFVSOL(18)000084_SOL005ed251_Making_authorisation_negotiation_more_flexi ble</li> </ul>
April 2018	2.4.3	<p>Version 2.4.3 based on contributions that were agreed at the NFVSOL#59, NFVSOL#60, and NFVSOL#61 meetings (including EA)</p> <ul style="list-style-type: none"> <li>- NFVSOL(18)000098_SOL005ed251_Empty_collections_clarification_addressing_ Plugtest™ issue</li> <li>- NFVSOL(18)000137_SOL005ed251_Disambiguating_artifactPath</li> <li>- NFVSOL(18)000155R1_SOL005ed251_Fix_cardinality_of_the_operationParams attribute_in_the_NsLcmOpOcc_data_type</li> </ul>

Date	Version	Information about changes
May 2018	2.4.4	Version 2.4.4 based on contributions that were agreed at the NFVSOL#62, NFVSOL#64, and NFVSOL#65 meetings (including EA) <ul style="list-style-type: none"> <li>- NFVSOL(18)000167_SOL005ed251:_fixing_tracker_issue_007748</li> <li>- NFVSOL(18)000177_SOL005ed251:_Correct_description_of_POST_method_on_NS_descriptors_resourc</li> <li>- NFVSOL(18)000184_SOL005ed251:_Change_the_cardinality_of_the_subscriptionId_attribute_in_the_NSD_management,_NS_LCM,_and_VNF_package_management_notifications</li> <li>- NFVSOL(18)000233_SOL005ed251:_Define_userDefinedData_attribute_consistently</li> <li>- NFVSOL(18)000235_SOL005ed251:_Remove_reference_to_the_note_in_the_description_of_the_pnfdInvariantId_attribute_in_clause_5.5.2.5</li> <li>- NFVSOL(18)000238_SOL005ed251:_Updating_JSON_RFC_reference</li> <li>- NFVSOL(18)000243_SOL005ed251:_VnfPkgm_small_fix</li> <li>- NFVSOL(18)000248_SOL005ed251:_Version_Management</li> <li>- NFVSOL(18)000251R1_SOL005ed251:_Move_ResourceHandle_to_common_data_types_in_clause_4.4.1.6</li> </ul>
July 2018	2.4.5	Version 2.4.5 based on contributions that were agreed at the NFVSOL#66 and NFVSOL#67 (including EA) <ul style="list-style-type: none"> <li>- NFVSOL(18)000244_SOL005ed251:_Different_names_for_virtual_link_descriptor_id</li> <li>- NFVSOL(18)000245_SOL005ed251:_Adding_status_codes</li> <li>- NFVSOL(18)000259R1_SOL005ed251:_Add_two_monitoringParameter_data_types</li> <li>- NFVSOL(18)000279_SOL005ed251:_Attribute_selectors</li> <li>- NFVSOL(18)000280_SOL005ed251:_Fixing_the_sequence_of_400_response_code_definitions</li> <li>- NFVSOL(18)000281_SOL005ed251:_Small_fix_to_the_description_of_the_400_error_code</li> <li>- NFVSOL(18)000311_SOL005ed251:_Small_fix_replace_queried_by_read</li> <li>- NFVSOL(18)000318_SOL005ed251:_Add_IFA027_reference</li> <li>- NFVSOL(18)000319_SOL005ed251:_Attribute_filters</li> <li>- NFVSOL(18)000320_SOL005ed251:_String_and_number_data_types</li> <li>- NFVSOL(18)000321_SOL005ed251:_Mirror_of_NFVSOL(18)000341r2</li> <li>- NFVSOL(18)000332R2_SOL005ed251:_Add_annex_with_a_reference_to_OpenAPI_repository</li> <li>- NFVSOL(18)000334_SOL005ed251:_Add_VL_profile_id_in_NsrtualLinkInfo_data_type</li> <li>- NFVSOL(18)000341_SOL005ed251:_Clarifying_association_from_VnfLinkPort to VnfcCp and VnfExtCp</li> </ul>
July 2018	2.4.6	Version 2.4.6 based on contributions that were agreed at the NFVSOL#68 and NFVSOL#69 (including EA) <ul style="list-style-type: none"> <li>- NFVSOL(18)000347_SOL005ed251:_Normative_attribute_filters_support</li> <li>- NFVSOL(18)000350_SOL005ed251:_Metadata_for_CPs_and_extCPs</li> <li>- NFVSOL(18)000355_SOL005ed251:_Add_relative_URIs_for_links_in_notifications</li> <li>- NFVSOL(18)000356_SOL005ed251:_Retry_as_reaction_to_error_responses_during_notification_delivery</li> <li>- NFVSOL(18)000357_SOL005ed251:_NestedNsInstanceData_for_the_InstantiateNs_and_UpdateNs_operations</li> <li>- NFVSOL(18)000358_SOL005ed251:_Add_ParamsForNestedNS_to_the_InstantiateNs_operations</li> <li>- NFVSOL(18)000359_SOL005ed251:_Clarify_linkage_between_vnfcCP_and_vnfcExtCP</li> <li>- NFVSOL(18)000360_SOL005ed251:_Attach_metadata_to_extCPs</li> <li>- NFVSOL(18)000391_SOL005ed251:_Define_minor_version_number</li> <li>- NFVSOL(18)000392_SOL005ed251:_Attribute_selector_attribute_filter_small_fixes</li> </ul>
July 2018	2.4.7	Version 2.4.7 based on contributions that were agreed at NFVSOL#71 (including EA) <ul style="list-style-type: none"> <li>- NFVSOL(18)000361_SOL005ed251:_Fix_NFP_Management</li> <li>- NFVSOL(18)000438_SOL005ed251:_Ensure_consistency_with_SOL003_on_VnfLinkPortInfo_and_VnfExtCpInfo</li> </ul>

Date	Version	Information about changes
August 2018	2.4.8	Version 2.4.8 based on contributions that were agreed at NFVSOL#72 and NFVSOL#72a (including EA) <ul style="list-style-type: none"> <li>- NFVSOL(18)000456R1_SOL005ed251:_Version_management</li> <li>- NFVSOL(18)000458_SOL005ed251:_Version_signaling</li> <li>- NFVSOL(18)000461R1_SOL005ed251: Define_patch_version_number</li> <li>- NFVSOL(18)000462_SOL005ed251: Closing_pagination_gap</li> <li>- NFVSOL(18)000473R1_SOL005ed251: Add_note_to_MAJOR_version_field</li> </ul>
September 2018	2.5.1	Publication
October 2018	2.5.2	Version 2.5.2 based on contributions that were agreed at NFVSOL#78, NFVSOL#79 and NFVSOL#80 (including EA) <ul style="list-style-type: none"> <li>- NFVSOL(18)000559r1_SOL005ed261_Mirror_of_552r5</li> <li>- NFVSOL(18)000569_Proposal_to_solve_inconsistencies_on_ModifyVnfInfoData_data_type</li> <li>- NFVSOL(18)000588r3_SOL005_Corrections</li> </ul> Editorials <ul style="list-style-type: none"> <li>- Updated Rapporteur name</li> </ul>
February 2019	2.5.3	Version 2.5.3 based on contributions that were agreed at NFVSOL#89 (EA) <ul style="list-style-type: none"> <li>- NFVSOL(19)000045_SOL005ed261_Mirror_of_581r2</li> <li>- NFVSOL(19)000048_SOL005ed261_Mirror_of_016</li> <li>- NFVSOL(19)000049_SOL005ed261_Mirror_of_040</li> </ul>
February 2019	2.5.4	Version 2.5.4 based on contributions that were agreed at NFVSOL#90 (EA) <ul style="list-style-type: none"> <li>- NFVSOL(19)000044_SOL005ed261_Mirror_of_584r2</li> <li>- NFVSOL(19)000047_SOL005ed261_Mirror_of_015</li> <li>- NFVSOL(19)000092r2_SOL005ed261_Alignment_of_Attributes_of_the_table_GET_method_of_the_PM_jobs</li> <li>- NFVSOL(19)000094r2_SOL005ed261_Mirror_of_057</li> <li>- NFVSOL(19)000095_SOL005ed261_Mirror_of_022r3</li> <li>- NFVSOL(19)000123r2_SOL005ed261_API_versions_update</li> </ul>
April 2019	2.6.1	Publication
July 2019	2.6.2	Version 2.6.2 based on contributions that were agreed at NFVSOL#96, NFVSOL #101 and NFVSOL #105 (EA) <ul style="list-style-type: none"> <li>- NFVSOL(19)000220r1_SOL005ed271_Small_fix_to_PnfExtCpData_in_Clause_6_5_3_16-1</li> <li>- NFVSOL(19)000264r2_SOL005Ed271_-_Clause_6_5_3_22_-_ParamsForVnf</li> <li>- NFVSOL(19)000403_SOL005ed271_Update_on_Alarm_data_type_descriptions</li> <li>- NFVSOL(19)000404r1_SOL005ed271_Alignment_of_clauses_4_4_2</li> <li>- NFVSOL(19)000405_SOL005ed271_Addition_of_standard_text_on_methods_of_clause_6_4_13</li> <li>- NFVSOL(19)000409_SOL005ed271_Addition_periodicity_of_triggering_to_Performance_Notification_data_type</li> <li>- NFVSOL(19)000417r1_SOL005ed271_General_normative_update</li> <li>- NFVSOL(19)000418_SOL005ed271_Inclusion_of_Content-Type_header_to_PATCH_method</li> <li>- NFVSOL(19)000419r1_SOL005ed271_Clarification_of_success_processing_of_the_asynchronous_request_in_POST_methods</li> <li>- NFVSOL(19)000420r2_SOL005ed271_Updating_of_the_Cancel_operation_description</li> </ul> Editorial fixes
October 2019	2.6.3	Version 2.6.3 based on contributions that were agreed at NFVSOL#108, NFVSOL#109, NFVSOL#113, NFVSOL#114 and NFVSOL#115 <ul style="list-style-type: none"> <li>- NFVSOL(19)000511r1_SOL005ed271_Aligning_version_indication_with_SOL015</li> <li>- NFVSOL(19)000530_SOL005ed271_mirror_of_change_5_in_454</li> <li>- NFVSOL(19)000540_SOL005ed271_Update_of_the_interfaces_descriptions_with_API_version_information_retrieval</li> <li>- NFVSOL(19)000542r1_SOL005ed271_Alignment_of_resources_descriptions</li> <li>- NFVSOL(19)000547_SOL005ed271_Addition_of_note_in_IpOverEthernetAddressInfo_datatype</li> <li>- NFVSOL(19)000549_SOL005ed271_Addition_of_the_id_in_Notifications_datatypes</li> <li>- NFVSOL(19)000565r1_SOL005ed271_Minor_updates_on_PM_and_FM_interfaces</li> <li>- NFVSOL(19)000575r2_SOL005ed271_Addresssing_Ns_Update_operation_data_type_definition_gap</li> </ul>

Date	Version	Information about changes
November 2019	2.6.4	<p>Version 2.6.4 based on contributions that were agreed at NFVSOL#116, NFVSOL#117 and NFVSOL#118</p> <ul style="list-style-type: none"> <li>- NFVSOL(19)000468_SOL005_fixes_related_to_IFA027</li> <li>- NFVSOL(19)000524_SOL005ed271_mirror_of_525_Marking_non-MANO_artifacts_in_VNF</li> <li>- NFVSOL(19)000526r2_SOL005ed271_Mirror_of_482r1_Fetching_the_manifest</li> <li>- NFVSOL(19)000533r2_SOL005ed271_Mirror_of_477r2_Bulk_fetch_of_package_artifacts</li> <li>- NFVSOL(19)000535_SOL005ed271_Mirror_of_483_Exposing_MaxScaleLevel</li> <li>- NFVSOL(19)000539r3_SOL005ed271_Clarify_passing_of_external_connectivity_information</li> <li>- NFVSOL(19)000545r2_SOL005ed271_Alignment_of_LCM_flows</li> <li>- NFVSOL(19)000546r1_SOL005ed271_Alignment_of_resources_and_datatypes_normative_descriptions</li> <li>- NFVSOL(19)000567r5_SOL005ed271_NSD_archive_and_PNFD_archive_management_changes</li> <li>- NFVSOL(19)000571r2_SOL005ed271_support_for_downloading_external_artifacts_from_URI</li> <li>- NFVSOL(19)000583r1_SOL005ed271_mirror_of_581_fixing_the_PM_interface_wrt_subscriptions</li> <li>- NFVSOL(19)000590r2_SOL005ed271_Moving_pre_and_post-conditions_into_normative_clauses</li> <li>- NFVSOL(19)000601_SOL005ed271_fixes_to_FM_interface</li> <li>- NFVSOL(19)000604_SOL005ed271_PATCH_alarm_acknowledge_status</li> <li>- NFVSOL(19)000637_SOL005ed271_mirror_of_523_example_of_artifactPath_in_GET</li> <li>- NFVSOL(19)000649r4_SOL005ed271_Missing_error_state_in_VNF_Package_onboarding</li> <li>- NFVSOL(19)000651_SOL005ed271_Correcting_description_of_individual_VNF_package_content_resource</li> <li>- NFVSOL(19)000652r1_SOL005ed271_Normative_postconditions_after_uploading_VNF_package_content</li> <li>- NFVSOL(19)000668_SOL005ed271_Mirror_of_659_Fixing_non-supprt_of_range_request</li> <li>- NFVSOL(19)000672r2_SOL005Ed271_-_Initial_configurable_properties_values_Mirror_of_576</li> <li>- NFVSOL(19)000676r1_SOL005ed271_mirror_of_675r3_Obtaining_artifact_security_info_via_the_API</li> <li>- NFVSOL(19)000678_SOL005ed271_mirror_of_667_fixing_VnfPackageChangeNotification_condition</li> <li>- NFVSOL(19)000681_SOL005ed271_mirror_of_679_adding_error_response_for_failed_notification_endpoint_test</li> <li>- NFVSOL(19)000694_SOL005ed217_adding_support_for_OAuth2_for_VNF_Package_Download</li> <li>- NFVSOL(19)000698_SOL005ed271_mirror_of_667_fixes_to_IFA_mapping_annex_related_to_dropping_subscriptions_from_PM_API</li> <li>- NFVSOL(19)000709_SOL005ed271_Mirror_of_708_-_clarify_the_zip_file_structure_returned_by_the_fetch_VNFD_operation</li> <li>- NFVSOL(19)000715_SOL005ed271_mirror_of_712r1_mark_for_testing_and_license_artifacts</li> <li>- NFVSOL(19)000723r1_SOL005ed271_mirror_of_722_VNF_package_metadata_in_VnfPkgInfo</li> <li>- NFVSOL(19)000733_SOL005ed271_fixed_multiple_small_bugs</li> <li>- NFVSOL(19)000750_SOL005ed271_mirror_of_474r1_version</li> <li>- NFVSOL(19)000769_SOL005ed271_mirror_of_755_Fix_for_the_disabled_VNF_packages</li> <li>- NFVSOL(19)000778r1_SOL005ed271_mirror_of_752r1_rapporteur_s_cleanup</li> <li>- NFVSOL(19)000782r1_SOL005ed271_API_versions_for_V2_7_1</li> <li>- NFVSOL(19)000786r1_SOL005ed271_mirror_of_653r2_Representing_the_artifact_path_of_external_artifacts</li> </ul> <p>This version contains the changes of NFVSOL(19)000661 as a rapporteur's action.</p>
January 2020	2.7.1	Publication

Date	Version	Information about changes
February 2020	3.0.1	<p>Version 3.0.1 based on contributions that were agreed at NFVSOL WG meetings</p> <ul style="list-style-type: none"> <li>- BWC:NFVSOL(19)000252r1_SOL005ed311_FEAT15_Clause_6_Adding_VNF_snapshot_operation_and_data_types</li> <li>- BWC:NFVSOL(19)000335_SOL005ed311_FEAT15_Clause_6_5_3_Data_type_of_DeleteVnfSnapshotData_and_RevertVnfToSnapshotData</li> <li>- BWC:NFVSOL(19)000358r3_SOL005ed311_FEAT04_NFVI_Capacity_about_subscription_and_notification</li> <li>- BWC:NFVSOL(19)000359r2_SOL005ed311_FEAT04_introduce_NFVI_Capacity_Information_interface</li> <li>- BWC:NFVSOL(19)000372r1_SOL005ed311_FEAT04_Query_NFVI_Capacity_Information_Operation</li> <li>- BWC:NFVSOL(19)000373r1_SOL005ed311_FEAT04_Add_NFVI_Capacity_thresholds</li> <li>- NBWCP:NFVSOL(19)000543r2_SOL005ed331_FEAT02_Mirror_of_508_Add_changeCurrentVnfPackage_LCM</li> <li>- BWC:NFVSOL(19)000734_SOL005ed331_FEAT10_Add_specification_for_Multi-Site_Connectivity_Service</li> <li>- BWC:NFVSOL(19)000797r1_SOL005ed331_NSD_archive_and_PNFD_archive_artifact_handling_operations</li> <li>- BWC:NFVSOL(19)000817_SOL005ed331_FEAT10_Bug_fixes_and_clarifications</li> <li>- BWC:NFVSOL(19)000850r2_SOL005ed331_mirror_of_619r2_FEAT02_extensions_confprops_and_metadata_during_ChangeCurrentVnfPackage</li> <li>- BWC:NFVSOL(20)000062_SOL005ed331_Annex_A_-_clause_A_7</li> <li>- BWC:NFVSOL(20)000063_SOL005ed331_Feature_annex</li> </ul> <p>Editorials:</p> <ul style="list-style-type: none"> <li>- Changed Release number to 3</li> </ul>
March 2020	3.0.2	<p>Contributions incorporated on version 3.0.2</p> <ul style="list-style-type: none"> <li>- BWC:NFVSOL(19)000813r2_SOL005ed331_FEAT15_VNF_Snapshot_Package_management_API</li> <li>- BWC:NFVSOL(20)000014_SOL005ed331_mirror_of_12_adding_missing_extensions_and_vnfConfigurableProperties_to_ChangeVnfFlavour</li> <li>- BWC:NFVSOL(20)000053r1_SOL005ed331_MECinNFV_placement_constraint_update</li> <li>- BWC:NFVSOL(20)000070_SOL005ed331_mirror_of_NFVSOL_20_000034_fixing_enumeration_of_PackageUsageStateType</li> <li>- BWC:NFVSOL(20)000080_SOL005ed331_FEAT10_Specification_of_WanConnectIonData</li> <li>- BWC:NFVSOL(20)000081_SOL005ed331_FEAT10_Handling_of_WAN_connectivity_information</li> <li>- BWC:NFVSOL(20)000082r2_SOL005ed331_FEAT10_Annex_multi-site_connectivity</li> <li>- BWC:NFVSOL(20)000112_SOL005ed331_Backported_111_Fixes_of_the_sequence_diagrams_postconditions</li> </ul>
April 2020	3.0.3	<p>Contributions incorporated on version 3.0.3</p> <ul style="list-style-type: none"> <li>- BWC:NFVSOL(20)000130r1_SOL005ed331_FEAT15_EN_resolution_bulk_mirrors_of_083_085r2_087r1_116r1</li> <li>- BWC:NFVSOL(20)000174_SOL005ed331_mirror_of_173r1_Bulk_fetch_of_external_artifacts</li> <li>- BWC:NFVSOL(20)000169r5_SOL005ed331_locationConstraint_update</li> <li>- BWC:NFVSOL(20)000188_SOL005ed331_mirror_of_187_Fix_a_Bug_of_the_Data_Type_VirtualStorageResourceInfo</li> <li>- BWC:NFVSOL(20)000213r1_SOL005ed331_Support_of_Trunking</li> <li>- BWC:NFVSOL(20)000234r2_SOL005ed331_FEAT15_Moving_VNF_snapshot_pkg_API</li> <li>- BWC:NFVSOL(20)000251_SOL005ed331_FEAT15_Mirror_of_186_Moving_checksum_into_SOL013</li> <li>- BWC:NFVSOL(20)000254_SOL005ed331_mirror_of_253_Small_fixes</li> <li>- BWC:NFVSOL(20)000275_SOL005ed331_FEAT15_Updates_to_VNF_snapshot_due_to_moving_VnfSnapshotPkg</li> <li>- BWC:NFVSOL(20)000318_SOL005ed331_Forward_mirror_of_317_Fixing_notifying_information_about_extLinkPort</li> </ul>

Date	Version	Information about changes
June 2020	3.0.4	<p>Contributions incorporated on version 3.0.4</p> <ul style="list-style-type: none"> <li>- BWC:NFVSOL(20)000235r1_SOL005ed331_FEAT10_Addressing_technical_EN_on_MSCS_support</li> <li>- BWC:NFVSOL(20)000252_SOL005ed331_FEAT10_Addressing_EN_in_MSCS_informative_annex</li> <li>- BWC:NFVSOL(20)000276r1_SOL005ed331_FEAT04_Addressing_EN_and_subscriptions_alignments</li> <li>- BWC:NFVSOL(20)000284_SOL005ed331_mirror_of_42r5_VnfcResourceInfo_bugfix</li> <li>- BWC:NFVSOL(20)000296_SOL005ed331_Forward_mirror_of_229r2_SOL016_review_alignments</li> <li>- BWC:NFVSOL(20)000336_SOL005ed331_FEAT15_Correction_on_top_of_234r2</li> <li>- BWC:NFVSOL(20)000364r1_SOL005ed331_Forward_mirror_of_294_Guidelines_link_ports_notifications</li> <li>- BWC:NFVSOL(20)000372_SOL005ed331_mirror_of_349_adding_PM_job_id_to_notification</li> <li>- BWC:NFVSOL(20)000392_SOL005ed331_FEAT15_Correction_cancel_operation_VnfSanpshotPkgm_API</li> <li>- BWC:NFVSOL(20)000419_SOL005ed331_Changes_related_to_the_testing_of_the_notification_endpoint</li> <li>- BWC:NFVSOL(20)000420r1_SOL005ed331_Mirror_of_416_VnfExtCpData_EN_resolution</li> <li>- BWC:NFVSOL(20)000421r1_SOL005ed331_Adding_422_error_code_to_the_response_bodies</li> <li>- NBWCP:NFVSOL(20)000423r1_SOL005ed331_mirror_of_210r1_ChangeExtVnfConnectivity_using_patch_semantics</li> <li>- BWC:NFVSOL(20)000435r1_SOL005ed331_Forward_mirror_of_433_Correction_external_images_VNF_package</li> <li>- BWC:NFVSOL(20)000455r1_SOL005ed331_mirror_of_159r1_Short_LcmOpOccNotifications</li> <li>- BWC:NFVSOL(20)000485r2_SOL005ed331_API_versions_update</li> <li>- BWC:NFVSOL(20)000486r4_SOL005ed331_Update_features_annex</li> <li>- BWC:NFVSOL(20)000487r1_SOL005ed331_FEAT15_Clean-up_remaining_EN</li> <li>- BWC:NFVSOL(20)000489_SOL005ed331_mirror_of_444_Fixing_EN_on_ModificationsTriggeredByVnfPkgChange</li> <li>- BWC:NFVSOL(20)000490_SOL005ed331_Bugfix_and_removing_EN_in_the_resource_NS_Instances</li> <li>- BWC:NFVSOL(20)000492r1_SOL005ed331_Ericsson_review_comments</li> <li>- BWC:NFVSOL(20)000508_SOL005ed331_ChangeVnfFlavourRequest_description_fix</li> <li>- BWC:NFVSOL(20)000524_SOL005ed331_FEAT15_Partial_mirror_of_407_Updating_VnfSnapshot_and_VnfcSnapshotInfos</li> <li>- BWC:NFVSOL(20)000526_SOL005ed331_API_versions_correction_on_top_of_485r2</li> <li>- BWC:NFVSOL(20)000536r2_SOL005ed331_clarification_for_VNFD_NSD_content</li> <li>- BWC:NFVSOL(20)000540_SOL005ed331_FEAT15_Identifier_scope_bug_on_top_of_487r1</li> <li>- BWC:NFVSOL(20)000552_SOL005ed331_Rapporteur_s_clean-up</li> <li>- BWC:NFVSOL(20)000558r1_SOL005ed331_FEAT15_Solving_issue_SOL010_nor_mative_reference</li> <li>- BWC:NFVSOL(20)000560r1_SOL005ed331_Aligning_SOL005_V030004r4_with_SOL003</li> <li>- BWC:NFVSOL(20)000566_SOL005ed331_FEAT15_Issue_alignment_VnfSnapshotPkgInfo</li> <li>- BWC:NFVSOL(20)000570r1_SOL005ed331_LCM-related_fixes</li> </ul>
September 2020	3.3.1	Publication

Date	Version	Information about changes
October 2020	3.3.2	Contributions incorporated on version 3.3.2 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(20)000673r2_SOL005ed341_NS_instantiation_inputs_parameter_clarification</li> <li>- BWC:NFVSOL(20)000687r1_SOL005ed341_FEAT15_Clarifications_about_snapshots_of_images_identifiers_and_packaging</li> <li>- BWC:NFVSOL(20)000699_SOL005ed341_FEAT15_Add_a_state_PROCESSING_of_the_VNF_snapshot_package</li> <li>- BWC:NFVSOL(20)000718r1_SOL005ed341_Fixing_of_the_flows_of_updating_the_callback_URI_of_the_PM_interface</li> <li>- BWC:NFVSOL(20)000726_SOL005ed341_Changes_related_to_test_notification_endpoint_of_PM_interface</li> </ul>
January 2021	3.3.3	Contributions incorporated on version 3.3.3 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(20)000711_SOL005ed351_mirror_of_709r2_VipCp_related_changes_from_IFA_CRs_608_666_and_802r1</li> <li>- NBWC:NFVSOL(20)000795r2_SOL005ed351_Editorials_fixes_and_clean-up</li> <li>- BWC:NFVSOL(20)000801_SOL005ed351_mirror_of_748r6_Adding_Trunk_Logical_Topology_between_VNFC_CP_s</li> </ul>
March 2021	3.3.4	Contributions incorporated on version 3.3.4 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(20)000781r3_SOL005ed351_mirror_of_778_extManagedVirtualLinkInfo_clarification</li> <li>- BWC:NFVSOL(20)000791_SOL005ed351_mirror_of_489_fix_identifier_datatypes_in_VnfExtCpInfo</li> <li>- BWC:NFVSOL(21)000040r2_SOL005ed351_Adding_vnfProfileId_to_InstantiateVnfData</li> <li>- BWC:NFVSOL(21)000057_SOL005ed351_Mirror_of_056_packageSecurityOption_cardinality_fix</li> <li>- BWC:NFVSOL(21)000067r1_SOL005ed351_archiveSecurityOption_cardinality_fix</li> <li>- BWC:NFVSOL(21)000114_SOL005ed351_Fix_of_the_attributes_names_prefixes_in_two_clauses</li> <li>- BWC:NFVSOL(21)000115_SOL005ed351_Fix_of_the_resource_Id_of_nsd_mgmt_in_two_diagrams</li> </ul>
March 2021	3.3.5	Contributions incorporated on version 3.3.5 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(21)000101r3_SOL005ed351_Introduction_of_the_NS_LCM_coord_interface</li> <li>- BWC:NFVSOL(21)000102r4_SOL005ed351_NS_LCM_coord_interface__sequence_diagrams</li> <li>- BWC:NFVSOL(21)000103r2_SOL005ed351_NS_LCM_coord_interface__resources_definition</li> <li>- BWC:NFVSOL(21)000105r3_SOL005ed351_NS_LCM_coord_interface__additional_changes</li> </ul>
March 2021	3.3.6	Contributions incorporated on version 3.3.6 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(21)000076r1_SOL005ed351_mirror_of_71_update_FEAT_Annex_list_regarding_ExtCp_changes</li> <li>- BWC:NFVSOL(21)000092_SOL005ed351_mirror_of_90r1_warnings_in_LcmOpOcc</li> <li>- BWC:NFVSOL(21)000104r4_SOL005ed351_NS_LCM_coord_interface__data_model</li> <li>- BWC:NFVSOL(21)000106r3_SOL005ed351_NS_LCM_coord_interface__changes_in_other_GS_parts</li> <li>- BWC:NFVSOL(21)000149r3_SOL005ed351_Adding_additional_attributes_to_Update_and_Terminate_VNF</li> <li>- BWC:NFVSOL(21)000161_SOL005ed351_Mirror_of_135_lcmcoord_API_addressing_ENs</li> </ul>
April 2021	3.3.7	Contributions incorporated on version 3.3.7 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(21)000111r4_SOL005ed351_FEAT05_Addition_of_priority_in_NS_data_types</li> <li>- BWC:NFVSOL(21)000185r1_SOL005ed351_mirror_of_183r2_notification_delivery_clarification</li> <li>- BWC:NFVSOL(21)000239r1_SOL005ed351_Misspelling_fix_of_data_types_in_clauses_11.4.9.3.2 and 11.4.9.3.3</li> <li>- BWC:NFVSOL(21)000274_SOL005ed351_Mirror_of_207__Handling_of_security_sensitive_properties</li> <li>- BWC:NFVSOL(21)000287_SOL005ed351_Fix_inconsistency_of_values_missed_in_LcmOpNameForChangeNotificationType</li> </ul>

Date	Version	Information about changes
May 2021	3.3.8	Contributions incorporated on version 3.3.8 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(21)000110r3_SOL005ed351_FEAT05_Introduction_of_the_Capacity_Shortage_Notification</li> <li>- BWC:NFVSOL(21)000283r1_SOL005ed351_FEAT05_Mirror_of_182_Add_enumeration_values_of_LcmCoordResultType_for_delay_case</li> </ul>
August 2021	3.3.9	Contributions incorporated on version 3.3.9 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(21)000109r8_SOL005ed351_FEAT05_Flow_of_NS_LCM_operations_with_pre-emption</li> <li>- BWC:NFVSOL(21)000432r2_SOL005ed351_API_versions_update</li> </ul>
October 2021	3.5.1	Publication
October 2021	4.0.1	Contributions incorporated on version 4.0.1 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(21)000534_SOL005ed431_Mirror_of_533_Addition_of_the_flow_to_Update_PNFD_Info_operation</li> </ul> Editorials: <ul style="list-style-type: none"> <li>- Changed release number to 4</li> </ul>
November 2021	4.0.2	Contributions incorporated on version 4.0.2 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(21)000588_SOL005ed431_ENH02_05_Introduction_of_nsScaleInfo_complementing nsInstantiationLevelId</li> <li>- BWC:NFVSOL(21)000590r1_ENH02_04_SOL005ed431_Add_DescriptorId_to_LCM_operations - CR "NBWC" classification was replaced by "BWC"</li> </ul>
December 2021	4.0.3	Contributions incorporated on version 4.0.3 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(21)000601_SOL005ed431_Rel4_mirror_of_600_Fixing_issue_introduced_by_NFVSOL(19)000850r2</li> <li>- BWC:NFVSOL(21)000608r1_SOL005ed431_Mirror_of_607_Fixing_gaps_in_clauses_5_5_1_and_6.5.1</li> <li>- BWC:NFVSOL(21)000611_SOL005ed431_Rel_4_mirror_of_610_fixes_to_NsChangeNotification_description</li> <li>- BWC:NFVSOL(21)000616_SOL005ed431_Rel_4_mirror_of_615_cpConfigId_clarifications</li> <li>- NBWC:NFVSOL(21)000645_SOL005ed431_Mirror_of_644_Fixing_attribute_type_names</li> <li>- BWC:NFVSOL(21)000648r1_SOL005ed431_Mirror_of_647_Fixing_resource_names_in_clause_5_3</li> </ul>
December 2021	4.0.4	Contributions incorporated on version 4.0.4 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(21)000562_SOL005ed431_Rel_4_mirror_of_561_vnfdId_in_resource_info_elements</li> <li>- BWC:NFVSOL(21)000675r1_SOL005ed431_Mirror_of_674_Description_update_for_additionalParams_attribute</li> <li>- BWC:NFVSOL(21)000683r1_SOL005ed431_Mirror_of_682_Fixing_resource_names_in_clause_9_3</li> </ul>
January 2022	4.0.5	Contributions incorporated on version 4.0.5 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(21)000666r1_SOL005ed431_Feature_Annex_update</li> <li>- BWC:NFVSOL(21)000698r2_ENH02_04_SOL005ed431_Add_ExtInvariantId_to_PkgInfo_structures</li> <li>- BWC:NFVSOL(22)000025_SOL005ed431_Removal_of_a_note_in_NsInstance</li> </ul>
March 2022	4.0.6	Contributions incorporated on version 4.0.6 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(21)000549r1_SOL005ed431_ENH02_02_Update_of_data_types</li> <li>- BWC:NFVSOL(21)000550r2_SOL005ed431_ENH02_02_Definition_of_data_types</li> <li>- BWC:NFVSOL(22)000012r3_SOL005ed431_adding_dataFlowMirroringData_in_InstantiateNsRequest</li> <li>- BWC:NFVSOL(22)000013r2_SOL005ed431_adding_DataFlowMirroringJob_in_UpdateNsRequest</li> <li>- BWC:NFVSOL(22)000014r2_SOL005ed431_adding_dataFlowMirroringJobInfo_in_NsInstance</li> <li>- BWC:NFVSOL(22)000037r1_SOL005ed431_FEAT17_Cloud-native_VNFs</li> <li>- BWC:NFVSOL(22)000073_SOL005ed431_Mirror_of_684r1_FEAT17_Add_data_type_MciInfo</li> </ul>
April 2022	4.0.7	Contributions incorporated on version 4.0.7 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(21)000551r1_SOL005ed431_ENH02_02_Update_of_POST_in_instantiate_and_update_tasks</li> <li>- BWC:NFVSOL(22)000121r1_FEAT17_SOL005ed431_Mirror_of_053_ResourceHandle</li> <li>- BWC:NFVSOL(22)000122_SOL005ed431_FEAT17_Runtime_modelling_of_VirtualCp</li> </ul>



Date	Version	Information about changes
April 2022	4.0.8	Contributions incorporated on version 4.0.8 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(21)000548r4_SOL005ed431_ENH02_02_Flow_of_NS_LCM_operations_with_feasibility_check</li> <li>- BWC:NFVSOL(22)000106r5_SOL005ed431_ENH02_02_Flow_to_modify_NS_LCM_info_operation</li> <li>- BWC:NFVSOL(22)000154r4_SOL005ed431_FEAT17_Cloud-native_VNFs</li> <li>- BWC:NFVSOL(22)000172_SOL005ed431_FEAT17_mirror_of_139r2_CpProtocolData</li> <li>- BWC:NFVSOL(22)000173_SOL005ed431_FEAT21_NFV_support_for_5G</li> </ul>
June 2022	4.0.9	Contributions incorporated on version 4.0.9 <ul style="list-style-type: none"> <li>- BWC:NFVSOL(21)000553r7_SOL005ed431_ENH02_02_PATH_method_definition_to_modify_an_individual_NS_LCM_resource</li> <li>- BWC:NFVSOL(22)000066r5_SOL005ed431_HTTP_header_fields_update_in_GET-PATCH_request_response</li> <li>- BWC:NFVSOL(22)000235_SOL005ed431_update_DataFlowDetails_description</li> <li>- BWC:NFVSOL(22)000247r1_SOL005ed431_Feature_Annex_update</li> <li>- BWC:NFVSOL(22)000264r1_SOL005ed431_clarification_of_the_collector</li> <li>- BWC:NFVSOL(22)000283r2_SOL005ed431_API_versions_update</li> </ul>

---

## History

<b>Document history</b>		
V4.3.1	August 2022	Publication