ETSI GS NFV-SOL 002 V3.6.1 (2022-02)

Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; RESTful protocols specification for the Ve-Vnfm 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-SOL002ed361

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

Contents

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 protocols and data models fulfilling the requirements specified in ETSI GS NFV-IFA 008 [1] for the interfaces used over the Ve-Vnfm 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] ETSI GS NFV-IFA 008: "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification".

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

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

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

[4] Recommendation ITU-T X.733: "Information technology - Open Systems Interconnection - Systems Management: Alarm reporting function".

[5] ETSI GS NFV-IFA 027: "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Performance Measurements Specification".

[6] ETSI GS NFV-SOL 013: "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; Specification of common aspects for RESTful NFV MANO APIs".

[7] ETSI GS NFV-IFA 011: "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; VNF Descriptor and Packaging Specification".

[8] IETF RFC 8141: "Uniform Resource Names (URNs)".

NOTE: Available at https://tools.ietf.org/rfc/rfc8141.txt.

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 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[i.2] ETSI GS NFV-SOL 003: "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; RESTful protocols specification for the Or-Vnfm Reference Point".

[i.3] ETSI GS NFV-SOL 001: "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; NFV descriptors based on TOSCA specification".

[i.4] OpenAPI™ Specification.

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

[i.5] Void.

[i.6] Void.

[i.7] Void.

[i.8] Void.

[i.9] ETSI GS NFV-SOL 015: "Network Functions Virtualisation (NFV); Protocols and Data Models; Specification of Patterns and Conventions for RESTful NFV-MANO APIs".

[i.10] ETSI GS NFV-IFA 007: "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GS NFV 003 [i.1] and the following apply:

LCM workflow: set of operations, including resource management operations towards the VIM, that are executed by the VNFM to perform a lifecycle management operation

NOTE: Examples for LCM workflows are VNFM-internal procedures associated with an LCM operation, and LCM scripts contained in the VNF package.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API	Application Programming Interface
CP	Connection Point
CPD	CP Descriptor
DHCP	Dynamic Host Configuration Protocol
EM	Element Manager
ETSI	European Telecommunications Standards Institute
FM	Fault Management
GS	Group Specification
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
IETF	Internet Engineering Task Force
IFA	Interfaces and Architecture
IP	Internet Protocol

ITU-T	International Telecommunications Union - Telecommunication
JSON	JavaScript Object Notation
LCCN	Life Cycle Change Notification
LCM	Lifecycle Management
MAC	Media Access Control
MANO	MANagement and Orchestration
NFV	Network Functions Virtualisation
NFVO	NFV Orchestrator
NS	Network Service
PM	Performance Management
REST	Representational State Transfer
RFC	Request For Comments
TOSCA	Topology and Orchestration Specification for Cloud Applications
URI	Uniform Resource Identifier
VDU	Virtualisation Deployment Unit
VIM	Virtualised Infrastructure Manager
VL	Virtual Link
VLD	VL Descriptor
VNF	Virtualised Network Function
VNFC	VNF Component
VNFD	VNF Descriptor
VNFLCM	Virtualised Network Function LifeCycle Management
VNFM	VNF Manager

4 General aspects

4.1 Overview

The present document defines the protocol and data model for the following interfaces used over the Ve-Vnfm reference point, in the form of RESTful Application Programming Interface (API) specifications:

- VNF Lifecycle Management interface (as produced by the VNFM towards the EM/VNF).
- VNF Performance Management interface (as produced by the VNFM towards the EM).
- VNF Fault Management interface (as produced by the VNFM towards the EM).
- VNF Indicator interface (as produced by the EM/VNF towards the VNFM).
- VNF Configuration interface (as produced by the VNF towards the VNFM).
- VNF LCM coordination interface (as produced by the EM/VNF towards the VNFM).

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
VNF Lifecycle Management interface	2.2.0
VNF Performance Management interface	2.1.0
VNF Fault Management interface	1.4.0
VNF Indicator interface	1.3.1
VNF Configuration interface	1.2.0
VNF LCM Coordination interface	1.0.0

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 008 [1]. In clause 4, general aspects are specified that apply to multiple APIs on the Ve-Vnfm reference point. In addition, the provisions in clauses 4.5, 6.8 and 9 of ETSI GS NFV-SOL 013 [6] 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 008 [1]. Annex B contains explanations of key concepts.

Even though the different 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 [6] 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: VnfInstanceSubscriptionFilter

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

Table 4.4.1.5-1: Definition of the VnfInstanceSubscriptionFilter data type

Attribute name	Data type	Cardinality	Description
vnfdIds	Identifier	0..N	If present, match VNF instances that were created based on a VNFD identified by one of the vnfdId values listed in this attribute. See note 1.
vnfProductsFromProviders	Structure (inlined)	0..N	If present, match VNF instances that belong to VNF products from certain providers. See note 1.
>vnfProvider	String	1	Name of the VNF provider to match.
>vnfProducts	Structure (inlined)	0..N	If present, match VNF instances that belong to VNF products with certain product names, from one particular provider.
>>vnfProductName	String	1	Name of the VNF product to match.

Attribute name	Data type	Cardinality	Description
>>versions	Structure (inlined)	0..N	If present, match VNF instances that belong to VNF products with certain versions and a certain product name, from one particular provider.
>>>vnfSoftwareVersion	Version	1	Software version to match.
>>>vnfdVersions	Version	0..N	If present, match VNF instances that belong to VNF products with certain VNFD versions, a certain software version and a certain product name, from one particular provider.
vnfInstanceIds	Identifier	0..N	If present, match VNF instances with an instance identifier listed in this attribute. See note 2.
vnfInstanceNames	String	0..N	If present, match VNF instances with a VNF Instance Name listed in this attribute. See note 2.
NOTE 1: The attributes "vnfdIds" and "vnfProductsFromProviders" are alternatives to reference to VNF instances that are based on certain VNFDs in a filter. They should not be used both in the same filter instance, but one alternative should be chosen. NOTE 2: The attributes "vnfInstanceIds" and "vnfInstanceNames" are alternatives to reference to particular VNF instances in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.			

4.4.1.6 Void

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 [6] shall apply.

Table 4.4.2.2-1: Simple data types

Type name	Description
IdentifierInVnfd	An identifier that is unique within a VNF descriptor. Representation: string of variable length.
IdentifierInVim	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.
IdentifierInVnf	An identifier that is unique for the respective type within a VNF instance, but that need not be globally unique. Representation: string of variable length.
IdentifierLocal	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.

4.4.2.3 Enumerations

4.4.2.3.1 Introduction

This clause defines enumerations that are referenced from data types in multiple interfaces. In addition, the enumerations defined in clause 7.2.3 of ETSI GS NFV-SOL 013 [6] shall apply to be available for referencing from data type definitions in the present document.

4.4.2.3.2 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 VNFM as the follow-up to this coordination. The LcmCoordResultType shall comply with the provisions defined in table 4.4.2.3.2-1.

Table 4.4.2.3.2-1: 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 VNFM. The related LCM operation shall be aborted by transitioning into the state "FAILED_TEMP".

4.5 Void

4.6 Void

4.7 Void

5 VNF Lifecycle Management interface

5.1 Description

This interface allows the VNF/EM to invoke VNF lifecycle management operations of VNF instances towards the VNFM, and to subscribe to notifications regarding VNF lifecycle changes provided by the VNFM.

The operations defined for VNF through this interface are:

- Query VNF
- Scale VNF
- Scale VNF to Level
- Heal VNF
- Get Operation Status
- Subscribe
- Query Subscription Information
- Terminate Subscription
- Notify

The operations defined for EM through this interface are:

- Create VNF Identifier
- Query VNF
- Modify VNF Information
- Delete VNF Identifier
- Instantiate VNF
- Scale VNF
- Scale VNF to Level

- Change VNF Flavour
- Terminate VNF
- Heal VNF
- Operate VNF
- Change external VNF connectivity
- Change current VNF package
- Create VNF/VNFC snapshot
- Revert to VNF/VNFC snapshot
- Query VNF/VNFC snapshot information
- Delete VNF/VNFC snapshot information
- Get Operation Status
- Subscribe
- Query Subscription Information
- Terminate Subscription
- Notify

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

5.1a API version

For the VNF lifecycle management interface version as specified in the present document, the MAJOR version field shall be 2, the MINOR version field shall be 2, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [6] 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 [6]. The string "vnflcm" 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 VNF lifecycle management interface.

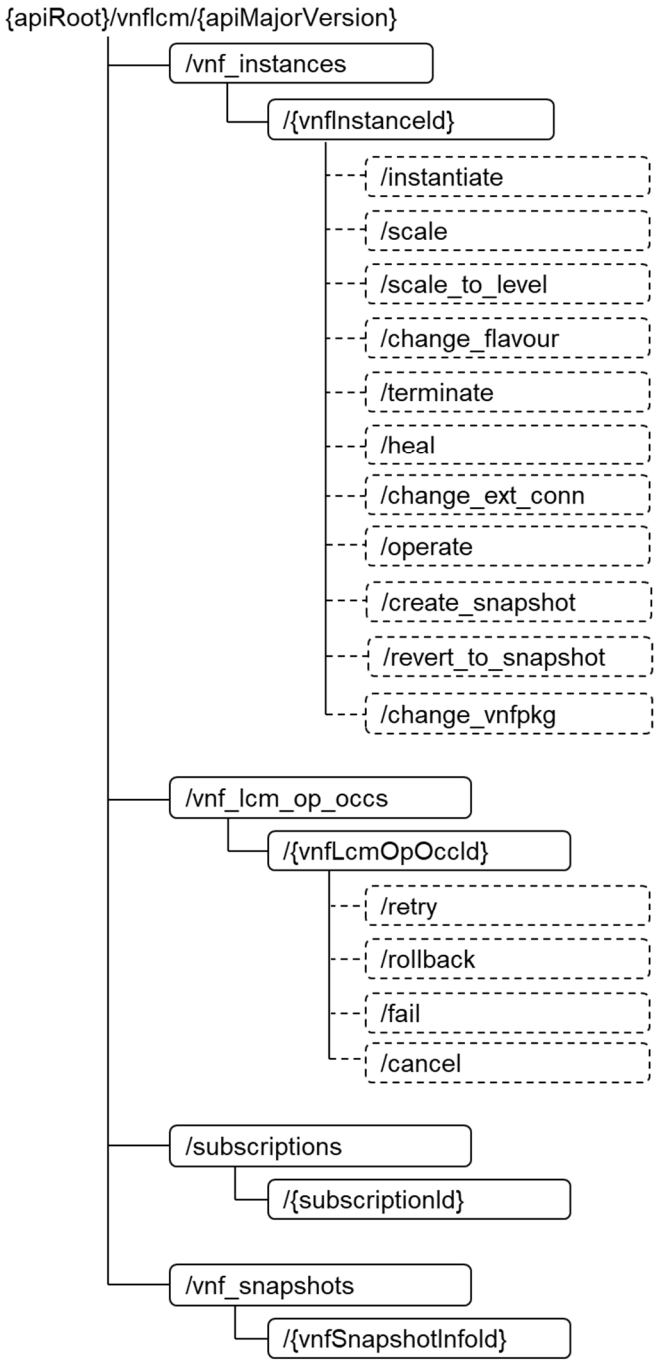

Figure 5.2-1: Resource URI structure of the VNF Lifecycle Management Interface

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

The VNFM 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 VNFM shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [6].

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

Resource name	Resource URI	HTTP Method	Cat	Meaning
VNF instances	/vnf_instances	GET	M	Query multiple VNF instances.
		POST	M	Create a new "Individual VNF instance" resource.

Resource name	Resource URI	HTTP Method	Cat	Meaning
Individual VNF instance	/vnf_instances/{vnfInstanceId}	GET	M	Read an "Individual VNF instance" resource.
		PATCH	M	Modify VNF instance information.
		DELETE	M	Delete an "Individual VNF instance" resource.
Instantiate VNF task	/vnf_instances/{vnfInstanceId} /instantiate	POST	M	Instantiate a VNF.
Scale VNF task	/vnf_instances/{vnfInstanceId} /scale	POST	M	Scale a VNF instance incrementally.
Scale VNF to Level task	/vnf_instances/{vnfInstanceId} /scale_to_level	POST	M	Scale a VNF instance to a target level.
Change VNF flavour task	/vnf_instances/{vnfInstanceId} /change_flavour	POST	M	Change the deployment flavour of a VNF instance.
Terminate VNF task	/vnf_instances/{vnfInstanceId} /terminate	POST	M	Terminate a VNF instance.
Heal VNF task	/vnf_instances/{vnfInstanceId} /heal	POST	M	Heal a VNF instance.
Operate VNF task	/vnf_instances/{vnfInstanceId} /operate	POST	M	Operate a VNF instance.
Change external VNF connectivity task	/vnf_instances/{vnfInstanceId} /change_ext_conn	POST	M	Change the external connectivity of a VNF instance.
Change current VNF package task	/vnf_instances/{vnfInstanceId} /change_ vnfpkg	POST	M	Change the current VNF package on which a VNF instance is based.
Create VNF snapshot task	/vnf_instances/{vnfInstanceId} /create_snapshot	POST	M	Create a VNF snapshot.
Revert to VNF snapshot task	/vnf_instances/{vnfInstanceId} /revert_to_snapshot	POST	M	Revert a VNF instance to a VNF snapshot.
VNF LCM operation occurrences	/vnf_lcm_op_occs	GET	M	Query information about multiple VNF lifecycle management operation occurrences.
Individual VNF LCM operation occurrence	/vnf_lcm_op_occs/{vnfLcmOp OccId}	GET	M	Read information about an individual VNF lifecycle management operation occurrence.
Retry operation task	/vnf_lcm_op_occs/{vnfLcmOp OccId}/retry	POST	M	Retry a VNF lifecycle management operation occurrence.
Rollback operation task	/vnf_lcm_op_occs/{vnfLcmOp OccId}/rollback	POST	M	Rollback a VNF lifecycle management operation occurrence.
Fail operation task	/vnf_lcm_op_occs/{vnfLcmOp OccId}/fail	POST	M	Mark a VNF lifecycle management operation occurrence as failed.
Cancel operation task	/vnf_lcm_op_occs/{vnfLcmOp OccId}/cancel	POST	M	Cancel a VNF lifecycle management operation occurrence.
VNF snapshots	/vnf_snapshots	GET	M	Query multiple VNF snapshots.
		POST	M	Create an individual VNF snapshot resource.
Individual VNF snapshot	/vnf_snapshots/{vnfSnapshotI nfoId}	GET	M	Read an individual VNF snapshot resource.
		DELETE	M	Delete VNF snapshot resource.
Subscriptions	/subscriptions	POST	M	Subscribe to VNF lifecycle 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 VNF lifecycle change.
		GET	See note	Test the notification endpoint.
NOTE: The VNFM shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the EM or VNF. If the EM or VNF supports invoking the POST method on the "Subscriptions" resource towards the VNFM, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.				

Table 5.4.1.2-1 specifies the preconditions and postconditions applicable to the task resources used to trigger the different VNF lifecycle management operations triggered by task resources.

5.3 Sequence diagrams (informative)

5.3.1 Flow of the creation of a VNF instance resource

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

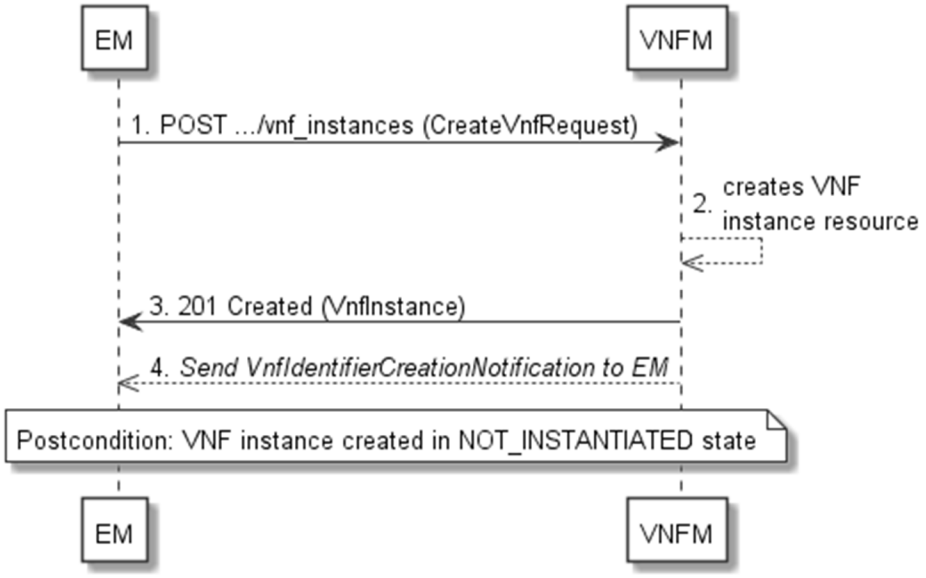

Figure 5.3.1-1: Flow of the creation of a VNF instance resource

NOTE: Due to possible race conditions, the 201 response and the VnfIdentifierCreationNotification can arrive in any order at the EM.

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

1. The EM sends a POST request to the "VNF Instances" resource including in the payload body a data structure of type "CreateVnfRequest".

2. The VNFM creates a new "Individual VNF instance" resource in NOT_INSTANTIATED state, and the associated VNF instance identifier.

3. The VNFM returns a 201 Created response containing a representation of the "Individual VNF instance" resource just created by the VNFM, and provides the URI of the newly-created resource in the "Location" HTTP header. See note above.

4. The VNFM sends a VNF Identifier Creation Notification (see clause 5.3.9) to the EM to indicate the creation of the "Individual VNF instance" resource and the associated VNF instance identifier. See note above.

Postcondition: Upon successful completion, a new "Individual VNF instance" resource has been created in "NOT_INSTANTIATED" state.

5.3.2 Flow of the deletion of a VNF instance resource

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

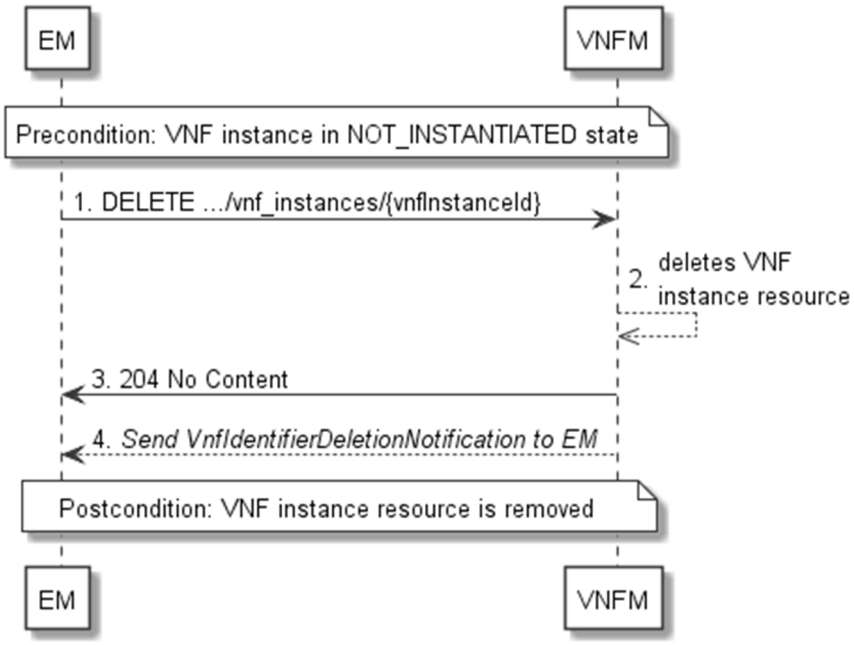

Figure 5.3.2-1: Flow of the deletion of a VNF instance resource

NOTE: Due to possible race conditions, the 204 response and the VnfIdentifierDeletionNotification can arrive in any order at the EM.

Precondition: The resource representing the VNF instance to be deleted needs to be in NOT_INSTANTIATED state.

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

1. EM sends a DELETE request to the "Individual VNF Instance" resource.
2. The VNFM deletes the "Individual VNF instance" resource and the associated VNF instance identifier.
3. The VNFM returns a "204 No Content" response with an empty payload body. See note above.
4. The VNFM sends to the EM a VnfIdentifierDeletionNotification to indicate the deletion of the "Individual VNF instance" resource and the associated VNF instance identifier. See note above.

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

Error handling: If the "Individual VNF instance" resource is not in NOT_INSTANTIATED state, the VNFM rejects the deletion request.

5.3.3 Flow of VNF lifecycle management operations triggered by task resources

This clause describes the general sequence for VNF Lifecycle Management operations that operate on VNF 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 5.4.1.2-1.

This flow is applicable to the following operations:

- Instantiate VNF
- Scale VNF
- Scale VNF to Level
- Change VNF flavour
- Operate VNF

- Heal VNF
- Change external VNF connectivity
- Change current VNF package
- Create VNF snapshot
- Revert to VNF snapshot
- Terminate VNF

Figure 5.3.3-1 illustrates the general lifecycle management flow. Placeholders in this flow allow for differentiating between the operations and are marked with double angular brackets "<<…>>".

NOTE 1: The API consumer can be either EM or VNF depending on the operations.

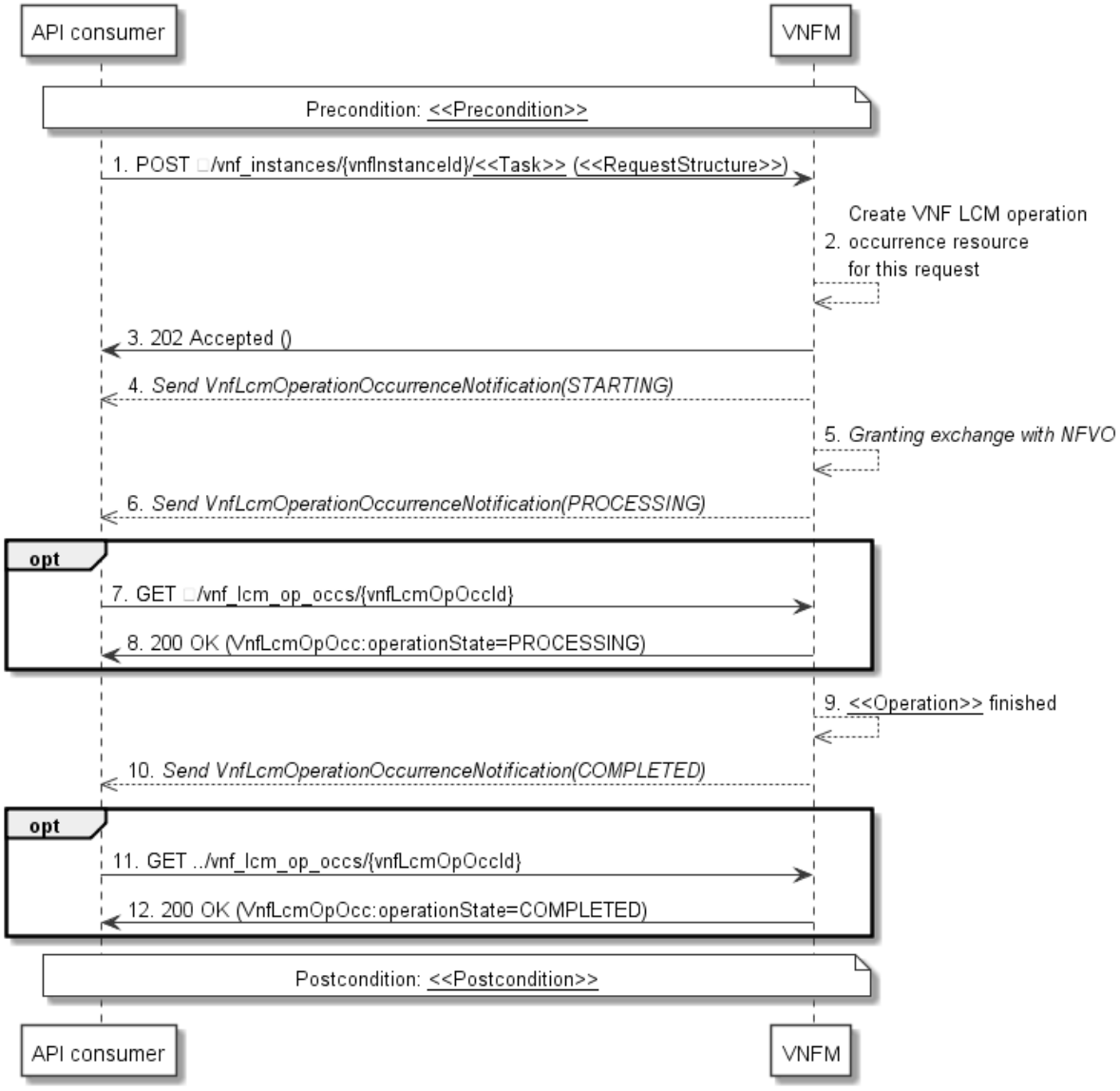

Figure 5.3.3-1: General flow of VNF lifecycle management operations triggered by task resources

NOTE 2: Due to possible race conditions, the 202 response and the "STARTING" VnfLcmOperationOccurrenceNotification can arrive in any order at the API consumer (EM/VNF).

Precondition: The precondition depends on the actual operation and is described by the template parameter <<Precondition>>. Table 5.4.1.2-1 specifies the applicable precondition.

A VNF lifecycle operation, as illustrated in figure 5.3.3-1, consists of the following steps:

1. The API consumer sends a POST request to the <<Task>> resource that represents the lifecycle operation to be executed on the VNF 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 5.4.1.2-1.

2. The VNFM creates a new "Individual VNF LCM operation occurrence" resource for the request.

3. The VNFM returns a "202 Accepted" response with an empty payload body and a "Location:" HTTP header that points to the new "Individual VNF LCM operation occurrence" resource, i.e. it includes the URI of that resource which is "…/vnf_lcm_op_occs/{vnfLcmOpOccId}". See note 2.

4. The VNFM sends to the API consumer a lifecycle management operation occurrence notification (see clause 5.3.9) to indicate the start of the lifecycle management operation occurrence with the "STARTING" state. See note 2.

5. VNFM and NFVO exchange granting information.

6. The VNFM sends to the API consumer a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate that the VNF LCM operation occurrence enters the "PROCESSING" state.

7. If desired, the API consumer can poll the "Individual VNF LCM operation occurrence" resource to obtain information about the ongoing operation by sending a GET request to the resource that represents the VNF LCM operation occurrence.

8. In the response to that request, the VNFM returns to the API consumer information of the operation, such as the operation status, by providing in the payload body a data structure of type "VnfLcmOpOcc".

9. The VNFM has finished the operation <<Operation>>.

10. The VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate the completion of the lifecycle management operation occurrence with the success state "COMPLETED".

11. If desired, the API consumer can send a new GET request to the "Individual VNF LCM operation occurrence" resource.

12. In the response to that request, the VNFM returns to the API consumer information about the result of the operation, by providing in the payload body a data structure of type "VnfLcmOpOcc".

Postcondition: The postcondition depends on the actual operation and is described by the template parameter <<Postcondition>>. Table 5.4.1.2-1 specifies the applicable precondition.

Error handling: If the VNF 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 VNF lifecycle management operation occurrence related to this VNF lifecycle management operation. Table 5.4.1.2-1 defines how the flow described above is parameterized for the different VNF lifecycle management operations.

5.3.4 Flow of automatic invocation of VNF scaling and VNF healing

This clause describes the sequence for the automatic invocation of "Scale VNF", "Scale VNF to Level" and "Heal VNF" operations by the VNFM, also known as "auto-scale" and "auto-heal". The criteria based on which the VNFM decides when to invoke an automatic scaling or automatic healing are outside the scope of the present document and can include certain changes in monitoring parameters that are monitored by the VNFM by PM jobs or thresholds, changes in VNF indicator values that are polled by the VNFM or that are reported to the VNFM via "VnfIndicatorValueChangeNotification" messages. Auto-scaling and auto-healing can be enabled and disabled by the EM by modifying the appropriate "isAutoscaleEnabled" and "isAutohealEnabled" configurable properties of the VNF using the sequence flow according to clause 5.3.6.

This flow is applicable to the automatic invocation of the following operations:

- Scale VNF
- Scale VNF to Level

- Heal VNF

Figure 5.3.4-1 illustrates the flow.

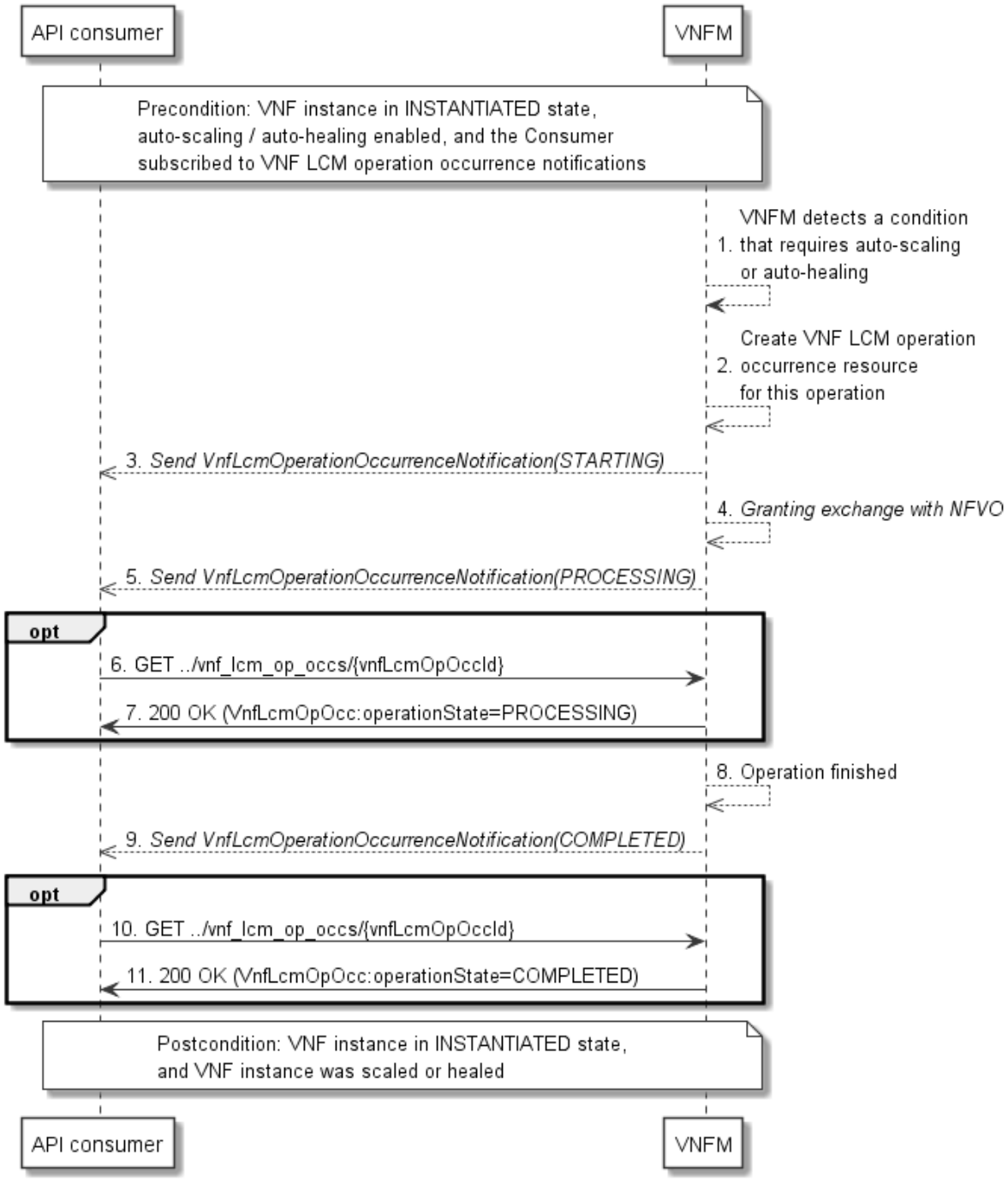

Figure 5.3.4-1: Flow of auto-scaling and auto-healing

Precondition: The VNF instance is in INSTANTIATED state, auto-scaling/auto-healing is enabled, and the API consumer is subscribed to VNF LCM operation occurrence notifications.

The automatic invocation of a VNF scaling or VNF healing operation, as illustrated in figure 5.3.4-1, consists of the following steps:

1. The VNFM detects a condition that triggers auto-scaling (Scale VNF or Scale VNF To Level) or auto-healing (Heal VNF) of the VNF, and selects the appropriate parameters for the operation.

2. The VNFM creates an "Individual VNF LCM operation occurrence" resource for the operation.

3. The VNFM sends to the API consumer a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate the start of the lifecycle management operation occurrence.
4. The VNFM and the NFVO exchange granting information.
5. The VNFM sends to the API consumer a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate that the VNF LCM operation occurrence enters the "PROCESSING" state.
6. If desired, the API consumer can poll the "Individual VNF lifecycle management operation occurrence" resource to obtain information about the ongoing operation by sending a GET request to the resource that represents the VNF lifecycle management operation occurrence.
7. In the response to that request, the VNFM returns to the API consumer information of the operation, such as the operation status, by providing in the payload body a data structure of type "VnfLcmOpOcc".
8. The VNFM has finished the operation.
9. The VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate the completion of the lifecycle management operation occurrence.
10. If desired, the API consumer can send a new GET request to the "Individual VNF lifecycle management operation occurrence" resource.
11. In the response to that request, the VNFM returns to the API consumer information about the result of the operation, by providing in the payload body a data structure of type "VnfLcmOpOcc".

Postcondition: The VNF instance in INSTANTIATED state, and the VNF instance has been scaled or healed as appropriate.

Error handling: If the VNF 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 VNF lifecycle management operation occurrence related to this VNF lifecycle management operation.

5.3.5 Flow of the Query VNF operation

This clause describes a sequence for querying/reading information about a VNF instance.

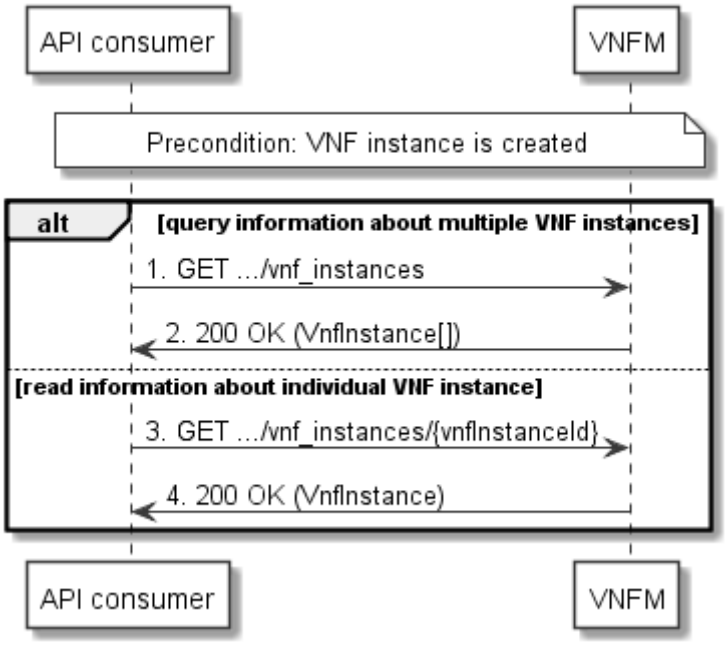

Figure 5.3.5-1: Flow of VNF instance query/read

Precondition: The resource representing the VNF instance has been created.

VNF instance query, as illustrated in figure 5.3.5-1, consists of the following steps:

1. If the API consumer intends to query all VNF instances, it sends a GET request to the "VNF instances" resource.
2. The VNFM returns a "200 OK" response to the API consumer, and includes zero or more data structures of type "VnfInstance" in the payload body.
3. If the API consumer intends to read information about a particular VNF instance, it sends a GET request to the "Individual VNF instance" resource, addressed by the appropriate VNF instance identifier in its resource URI.
4. The VNFM returns a "200 OK" response to the API consumer, and includes one data structure of type "VnfInstance" in the payload body.

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

5.3.6 Flow of the Modify VNF Information operation

This clause describes a sequence for updating information about a VNF instance.

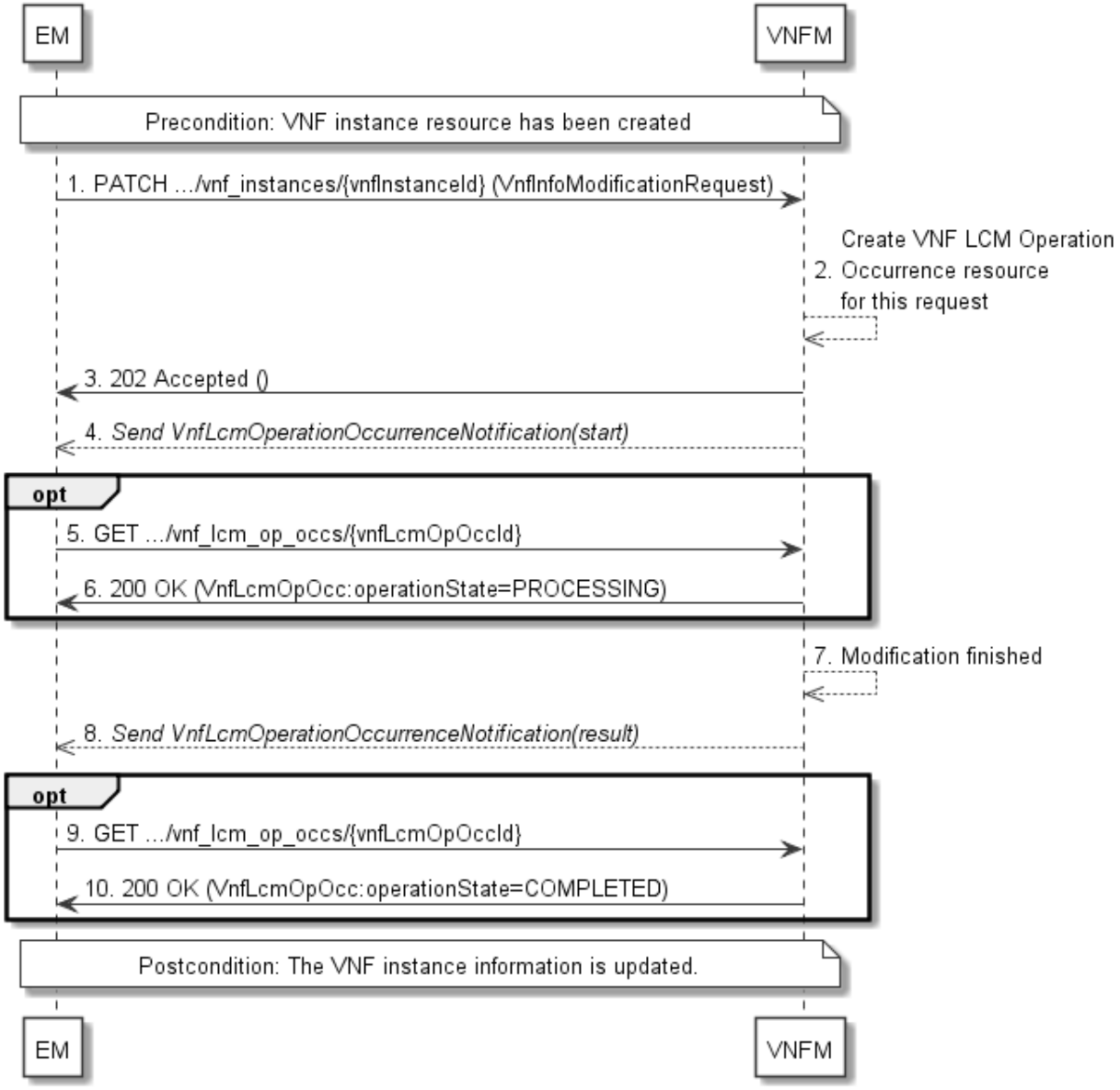

Figure 5.3.6-1: Flow of the modification of VNF instance information

NOTE: Due to possible race conditions, the 202 response and the VnfLcmOperationOccurrenceNotification can arrive in any order at the EM.

Precondition: The resource representing the VNF instance has been created.

Updating the VNF instance information, as illustrated in figure 5.3.6-1, consists of the following steps:

1. The EM sends a PATCH request to the "Individual VNF instance" resource of the VNF instance that is to be operated and includes in the payload body a data structure of type "VnfInfoModificationRequest".
2. The VNFM creates an Individual "VNF LCM operation occurrence" resource for the request.
3. The VNFM returns a "202 Accepted" response with an empty payload body and a "Location" HTTP header that points to the new "Individual VNF LCM operation occurrence" resource, i.e. it includes the URI of that resource which is "…/vnf_lcm_op_occs/{vnfLcmOpOccId}". See note above.
4. The VNFM sends to the EM a lifecycle management operation occurrence notification (see clause 5.3.9) to indicate the start of the operation. See note above.
5. If desired, the EM can poll the "Individual VNF LCM operation occurrence" resource to obtain information about the ongoing operation by sending a GET request to the resource that represents the VNF lifecycle management operation occurrence.
6. In the response to that request, the VNFM returns to the EM information of the operation, such as the operation status, by providing in the payload body a data structure of type "VnfLcmOpOcc".
7. The VNFM has finished the modification operation.
8. The VNFM sends a lifecycle management operation occurrence notification (see clause 5.3.9) to indicate the completion of the operation, and the performed changes. See note above.
9. If desired, the EM can send a new GET request to the "Individual VNF LCM operation occurrence" resource.
10. In the response to that request, the VNFM returns to the EM information about the result of the operation, by providing in the payload body a data structure of type "VnfLcmOpOcc".

Postcondition: Upon successful completion, information of the VNF instance is updated.

Error handling: If the updating of VNF instance information 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 VNF lifecycle management operation occurrence related to this VNF LCM operation.

5.3.7 Flow of the Get Operation Status operation

This clause describes a sequence for obtaining the status of a VNF lifecycle management operation occurrence.

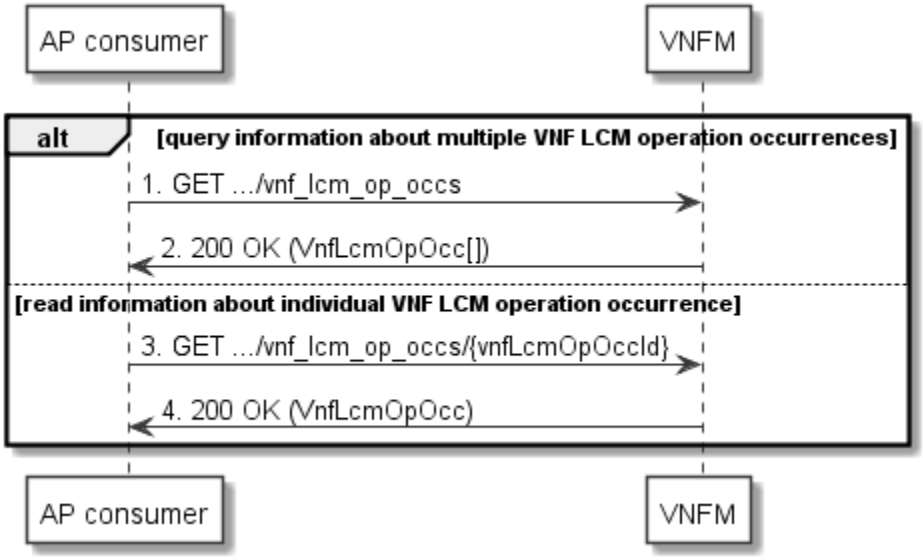

Figure 5.3.7-1: Flow of Get VNF lifecycle operation status

Obtaining the VNF lifecycle operation status, as illustrated in figure 5.3.7-1, consists of the following steps:

1. If the API consumer intends to query all VNF lifecycle management operation occurrences, it sends a GET request to the "VNF LCM operation occurrences" resource.

2. The VNFM returns a "200 OK" response to the API consumer, and includes zero or more data structures of type "VnfLcmOpOcc" in the payload body.

3. If the API consumer intends to read information about a particular VNF LCM operation occurrence, it sends a GET request to the "Individual VNF LCM operation occurrence" resource, addressed by the appropriate VNF LCM operation occurrence identifier in its resource URI.

4. The VNFM returns a "200 OK" response to the API consumer, and includes one data structure of type "VnfLcmOpOcc" in the payload body.

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

5.3.8 Flow of managing subscriptions

This clause describes the procedure for creating, querying/reading and terminating subscriptions to notifications related to VNF lifecycle management.

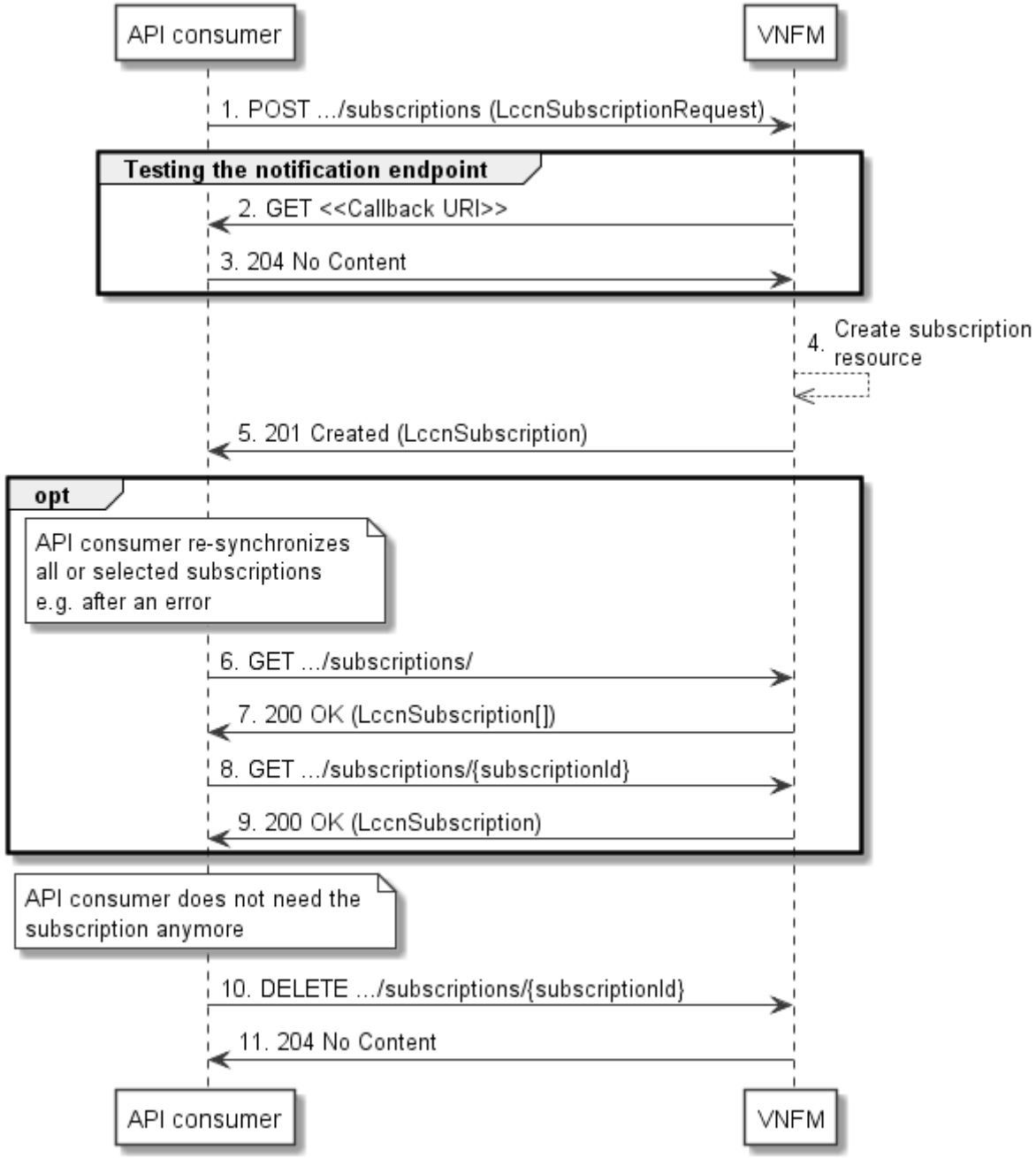

Figure 5.3.8-1: Flow of managing subscriptions

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

1. The API consumer 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 VNFM will subsequently send notifications about events that match the filter.

2. To test the notification endpoint that has been registered by the API consumer as part of the subscription, the VNFM sends a GET request to the notification endpoint URI.

3. The API consumer returns a "204 No Content" response to indicate success.

4. The VNFM creates a new subscription to notifications related to VNF lifecycle changes, and an "Individual subscription" resource that represents this subscription.

5. The VNFM returns a 201 Created response containing a data structure of type "LccnSubscription" representing the "Individual subscription" resource just created by the VNFM, 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 API consumer can query information about its subscriptions by sending a GET request to the resource representing the subscriptions.

7. In that case, the VNFM returns a "200 OK" response that contains zero or more representations of all existing subscriptions that were created by the API consumer.

8. If desired, e.g. to recover from an error situation, the API consumer can read information about a particular subscription by sending a GET request to the resource representing that individual subscription.

9. In that case, the VNFM returns a "200 OK" response that contains a representation of that individual subscription.

10. If the API consumer 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 API consumer acknowledges the successful termination of the subscription by returning a "204 No Content" response.

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

5.3.9 Flow of sending notifications

This clause describes the procedure for sending notifications.

NOTE 1: Notifications merely report to subscribed NFV-MANO entities the state changes of a VNF instance and/or 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 VNF 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.

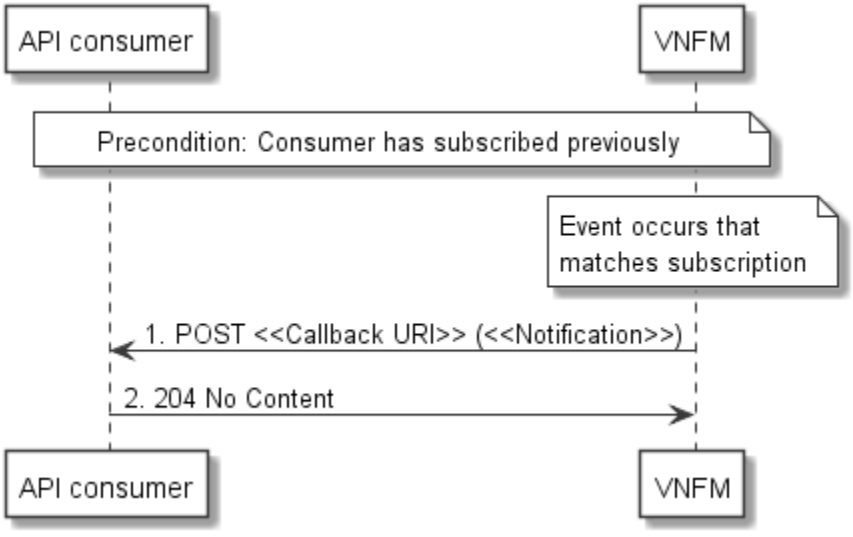

Figure 5.3.9-1: Flow of sending notifications

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

Precondition: The API consumer has subscribed previously to notifications related to VNF lifecycle management:

1. If an event occurs that matches the filtering criteria which are part of the subscription, the VNFM generates a notification that includes information about the event, and sends it in the body of a POST request to the URI which the API consumer 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 7.5.2.5, 7.5.2.6 and 7.5.2.8).

2. The API consumer acknowledges the successful delivery of the notification by returning a "204 No Content" response.

Error handling: If the VNFM does not receive the "204 No Content" response from the API consumer, it can retry sending the notification.

5.3.10 Flow of retrying a VNF lifecycle management operation

This clause describes a sequence for retrying a VNF lifecycle management operation occurrence that is represented by an "Individual VNF 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 VNF lifecycle management errors is provided in clause 5.6.

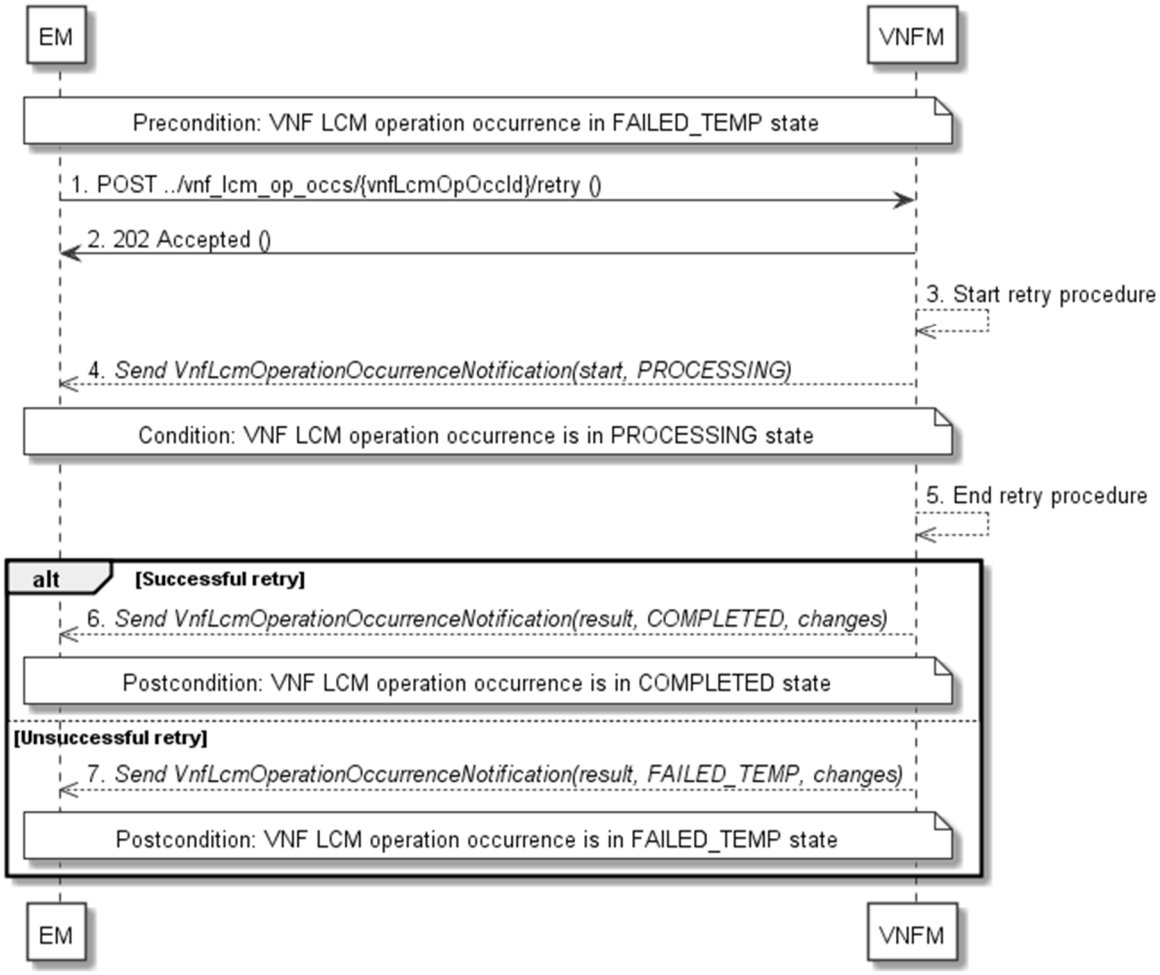

Figure 5.3.10-1: Flow of retrying a VNF lifecycle management operation

NOTE: Due to possible race conditions, the 202 response and the "PROCESSING" VnfLcmOperationOccurrenceNotification can arrive in any order at the EM.

Precondition: The VNF lifecycle management operation occurrence is in FAILED_TEMP state.

Retrying a VNF lifecycle operation, as illustrated in figure 5.3.10-1, consists of the following steps:

1. The EM sends a POST request with an empty body to the "Retry operation task" resource of the VNF LCM operation occurrence that is to be retried.
2. The VNFM returns a "202 Accepted" response. See note above.
3. The VNFM starts the retry procedure.
4. The VNFM sends a VNF lifecycle management operation occurrence notification of type "start" to indicate that the VNF LCM operation occurrence enters the "PROCESSING" state. See note above.
5. The VNFM finishes the retry procedure.
6. On successful retry, the VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate successful completion of the operation, and inform the EM about the virtualised resources changes.
7. On unsuccessful retry, the VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate an intermediate error (retry failed) of the operation, and to inform the EM about the virtualised resources changes.

Postcondition: The VNF lifecycle management operation occurrence is in one of the following states: FAILED_TEMP, COMPLETED. COMPLETED is a terminal state (see clause 5.6.2.2).

Error handling: The operation is rejected in case the "Individual VNF LCM operation occurrence" resource is in any other state than FAILED_TEMP, or in case Retry is not supported by for the particular VNF LCM operation for the particular VNF.

5.3.11 Flow of rolling back a VNF lifecycle management operation

This clause describes a sequence for rolling back a VNF lifecycle management operation occurrence that is represented by an "Individual VNF 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 VNF lifecycle management errors is provided in clause 5.6.

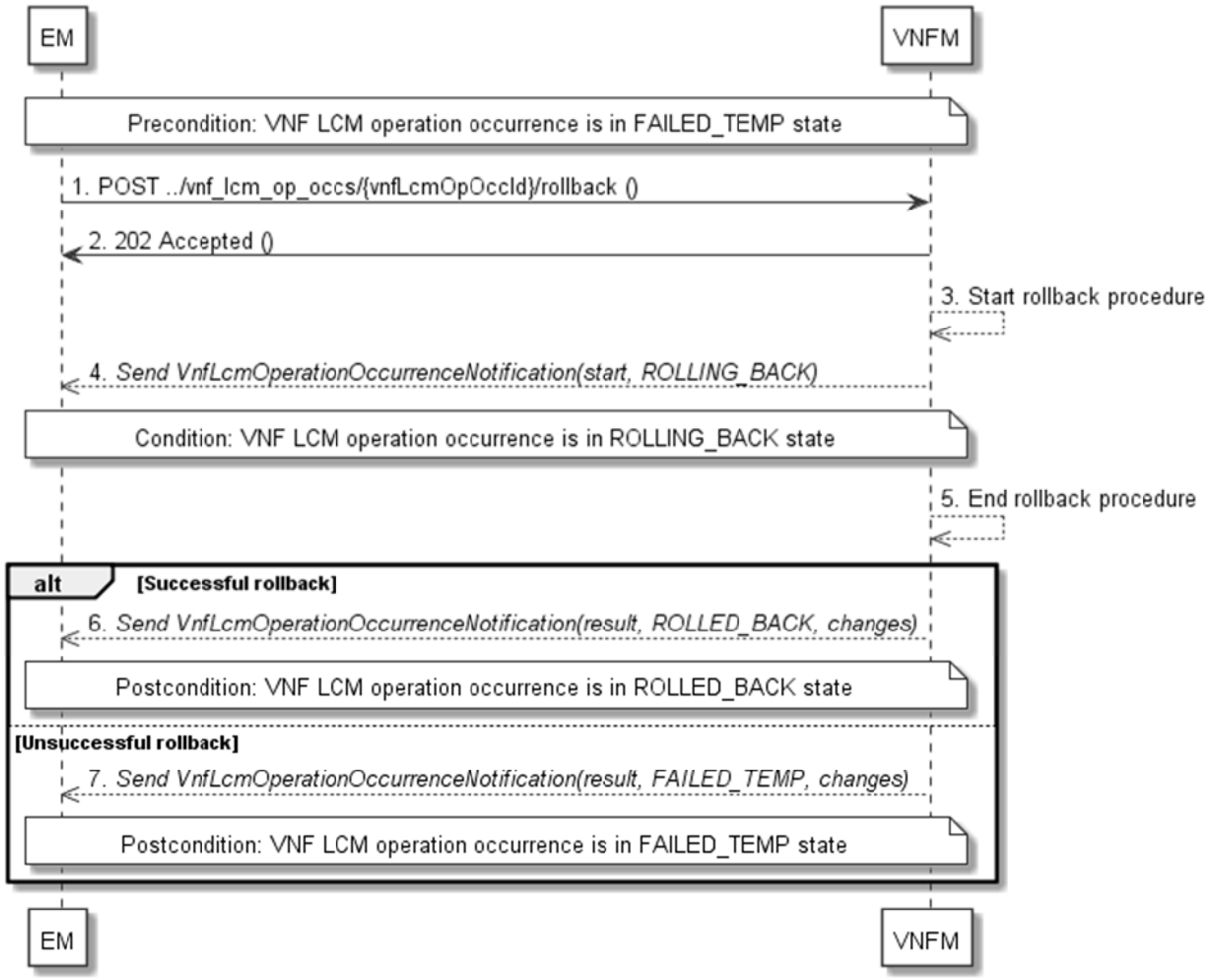

Figure 5.3.11-1: Flow of rolling back a VNF lifecycle management operation

NOTE: Due to possible race conditions, the 202 response and the "ROLLING_BACK" VnfLcmOperationOccurrenceNotification can arrive in any order at the EM.

Precondition: The VNF lifecycle management operation occurrence is in FAILED_TEMP state.

Initiating the rollback of a VNF lifecycle management operation, as illustrated in figure 5.3.11-1, consists of the following steps:

1. The EM sends a POST request with an empty body to the "Rollback operation task" resource of the VNF LCM operation occurrence that is to be rolled back.
2. The VNFM returns a "202 Accepted" response. See note above.
3. The VNFM starts the rollback procedure.
4. The VNFM sends a VNF lifecycle management operation occurrence notification of type "start" to indicate that the VNF LCM operation occurrence enters the "ROLLING_BACK" state. See note above.
5. The VNFM finishes the rollback procedure.
6. On successful rollback, the VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate successful completion of the operation, and inform the EM about the virtualised resources changes.
7. On unsuccessful retry, the VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate an intermediate error (rollback failed) of the operation, and to inform the EM about the virtualised resources changes.

Postcondition: The VNF lifecycle management operation occurrence is in one of the following states: FAILED_TEMP, ROLLED_BACK. ROLLED_BACK is a terminal state (see clause 5.6.2.2).

Error handling: The operation is rejected in case the VNF lifecycle management operation occurrence is in any other state than FAILED_TEMP, or in case Rollback is not supported for the particular VNF LCM operation for the particular VNF.

5.3.12 Flow of failing a VNF lifecycle management operation

This clause describes a sequence for declaring as "failed" a VNF lifecycle management operation occurrence that is represented by an "Individual VNF 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 HealVnf, or non-graceful VNF termination, on the affected VNF instance.

A comprehensive description of the handling of VNF lifecycle management errors is provided in clause 5.6.

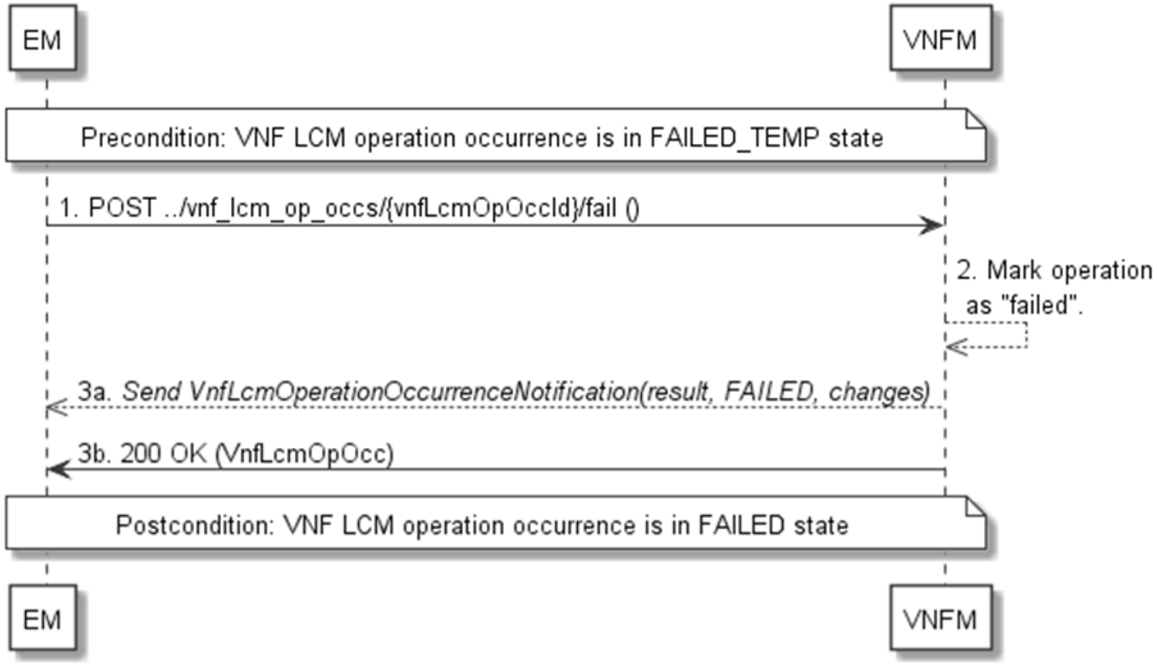

Figure 5.3.12-1: Flow of declaring a VNF lifecycle management operation as failed

NOTE: Due to possible race conditions, the 200 response and the "FAILED" VnfLcmOperationOccurrenceNotification can arrive in any order at the EM.

Precondition: The VNF lifecycle management operation occurrence is in FAILED_TEMP state.

Declaring a VNF lifecycle management operation as failed, as illustrated in figure 5.3.12-1, consists of the following steps:

1. The EM sends a POST request with an empty body to the "Fail operation task" resource of the VNF LCM operation occurrence that is to be marked as failed.

2. The VNFM marks the operation as failed.

3a. The VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate the final failure of the operation, and to inform the EM about the virtualised resources changes. See note above.

3b. Furthermore, it returns a "200 OK" response, and includes in the body a VnfLcmOpOcc structure. The order in which the response and the notification arrive at the EM is not defined. See note above.

Postcondition: The VNF lifecycle management operation occurrence is FAILED state. This is a terminal state (see clause 5.6.2.2).

Error handling: The operation is rejected in case the VNF lifecycle management operation occurrence is in any other state than FAILED_TEMP.

5.3.13 Flow of cancelling a VNF lifecycle management operation

This clause describes a sequence for cancelling an ongoing VNF LCM operation occurrence, or a rollback of a VNF 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 VNF lifecycle management errors is provided in clause 5.6.

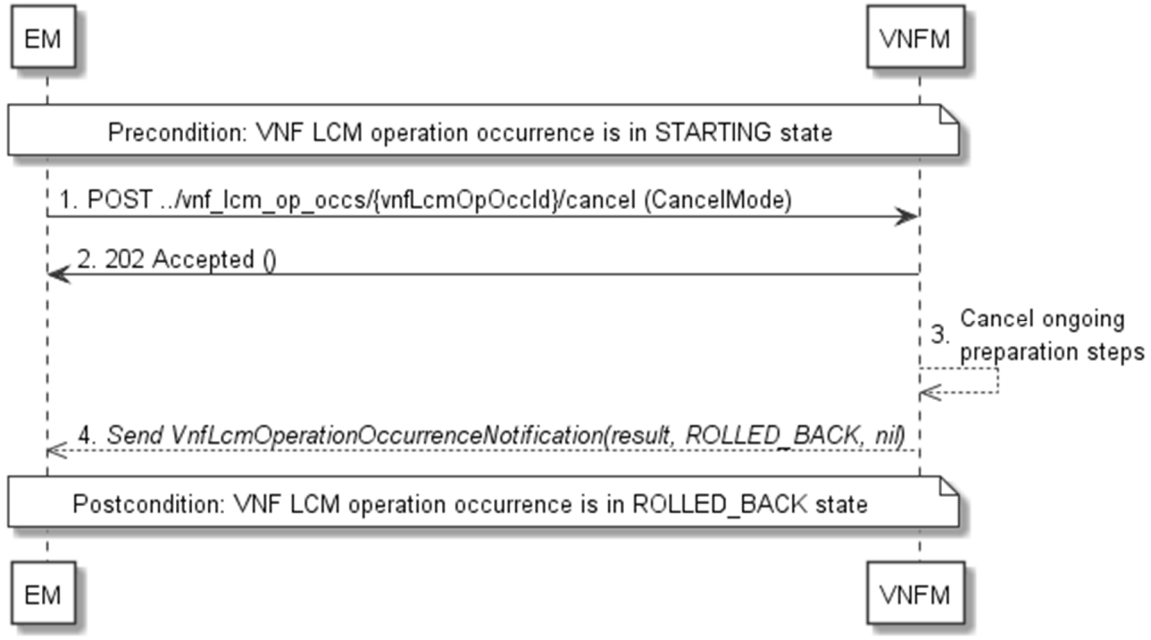

Figure 5.3.13-1: Flow of cancelling a VNF lifecycle management operation in "STARTING" state

NOTE 1: Due to possible race conditions, the 202 response and the "ROLLED_BACK" VnfLcmOperationOccurrenceNotification can arrive in any order at the EM.

Precondition: The VNF lifecycle management operation occurrence is in STARTING state.

Cancelling a VNF lifecycle operation when it is in STARTING state, as illustrated in figure 5.3.13-1, consists of the following steps:

1. The EM sends a POST request with a "CancelMode" structure in the body to the "Cancel operation task" resource of the VNF LCM operation occurrence that is to be cancelled.
2. The VNFM returns a "202 Accepted" response. See note 1 above.
3. The VNFM cancels the ongoing preparation operations.
4. The VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate an intermediate error (cancelled) of the operation, and to inform the EM that there were no virtualised resources changes. See note 1 above.

Postcondition: The VNF lifecycle management operation occurrence is in ROLLED_BACK state.

Error handling: The operation is rejected in case the VNF lifecycle operation occurrence is in any other state than STARTING.

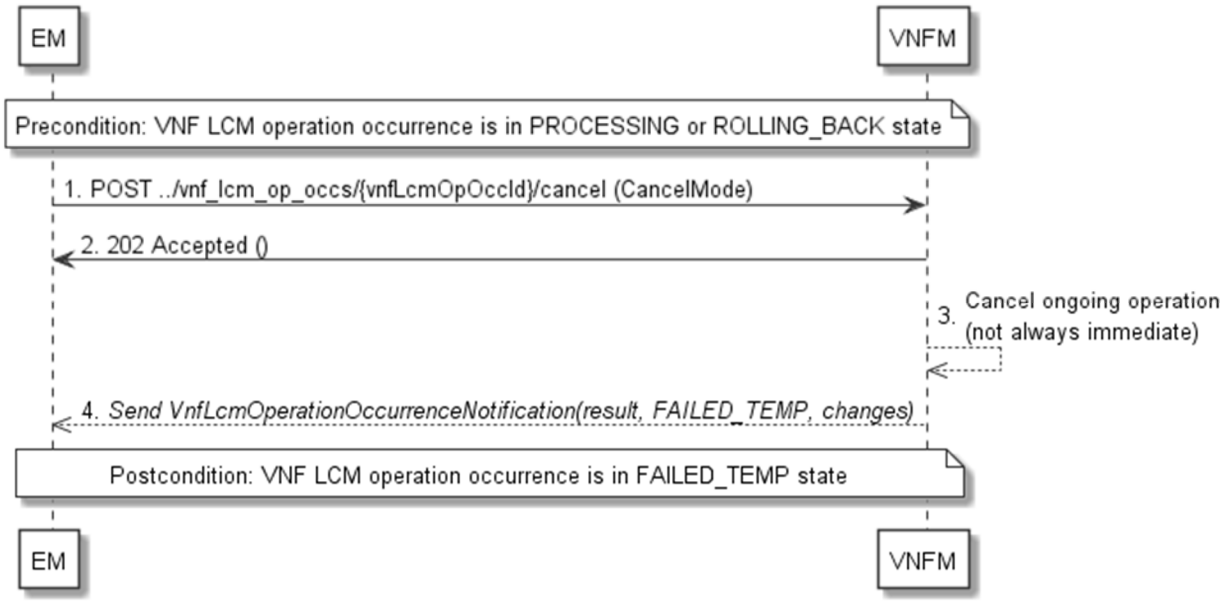

Figure 5.3.13-2: Flow of cancelling a VNF lifecycle management operation in "PROCESSING" or "ROLLING_BACK" state

NOTE 2: Due to possible race conditions, the 202 response and the "FAILED_TEMP" VnfLcmOperationOccurrenceNotification can arrive in any order at the EM.

Precondition: The VNF lifecycle management operation occurrence is in PROCESSING or ROLLING_BACK state.

Cancelling a VNF lifecycle operation when it is in "PROCESSING" or "ROLLING_BACK" state, as illustrated in figure 5.3.13-2, consists of the following steps:

1. The EM sends a POST request with a "CancelMode" structure in the body to the "Cancel operation task" resource of the VNF LCM operation occurrence that is to be cancelled.
2. The VNFM returns a "202 Accepted" response. See note 2 above.
3. The VNFM cancels the ongoing LCM operation. This can take some time.
4. The VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate an intermediate error (cancelled) of the operation, and to inform the EM about the virtualised resources changes. See note 2 above.

Postcondition: The VNF lifecycle management operation occurrence is FAILED_TEMP state.

Error handling: The operation is rejected in case the VNF lifecycle management operation occurrence is in any other state than PROCESSING or ROLLING_BACK, or in case Cancel is not supported for the particular VNF LCM operation for the particular VNF.

5.3.14 Flow of creation of a VNF snapshot resource

This clause describes the procedure for the creation of an "Individual VNF snapshot" resource.

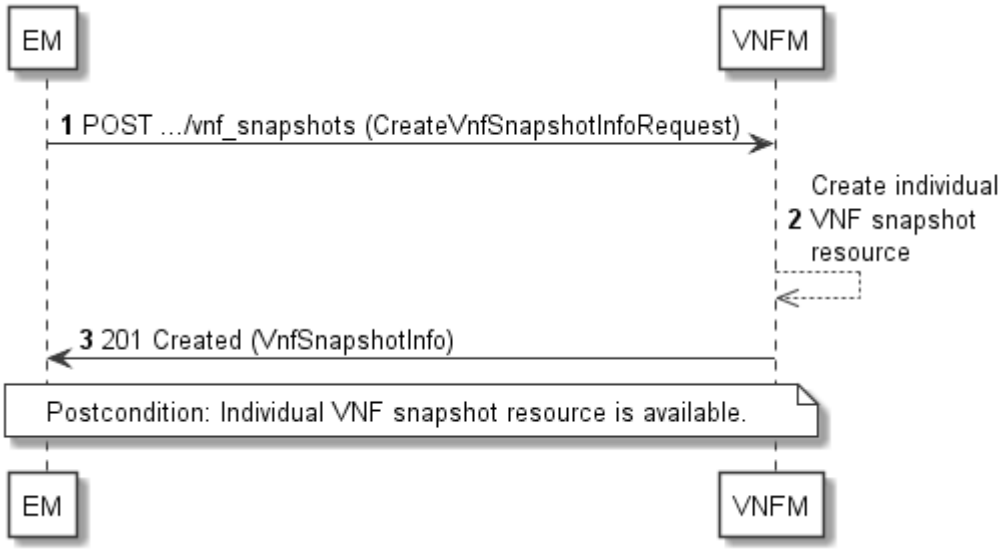

Figure 5.3.14-1: Flow of creation of a VNF snapshot resource

The procedure consists of the following steps as illustrated in figure 5.3.14-1:

1. The EM sends a POST request to the "VNF snapshots" resource and includes in the payload body a data structure of type "CreateVnfSnapshotInfoRequest".
2. The VNFM creates a new "individual VNF snapshot" resource.
3. The VNFM returns a "201 Created" response containing a representation of the individual VNF snapshot resource and a "Location" HTTP header that points to the new "individual VNF snapshot" resource.

Postcondition: The resource representing the VNF/VNFC snapshot has been created and is available.

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

5.3.15 Flow of the Query VNF Snapshot operation

This clause describes a sequence for querying/reading information about one or more VNF/VNFC snapshots.

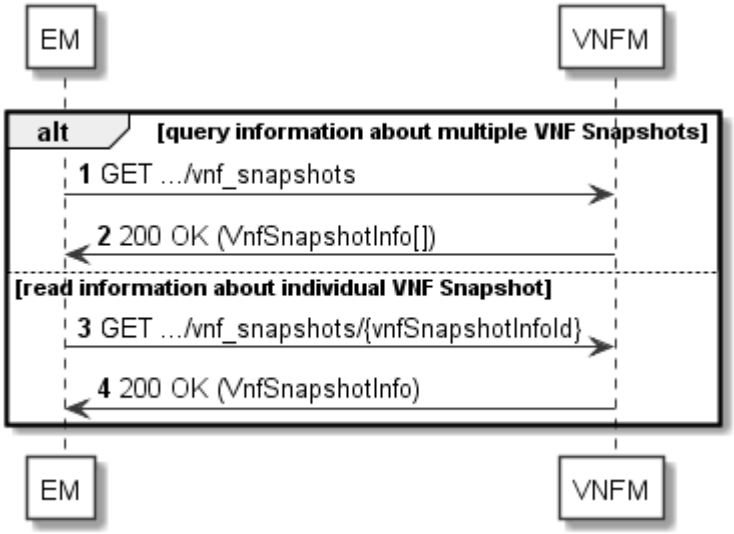

Figure 5.3.15-1: Flow of VNF snapshot query/read

VNF snapshot query/read, as illustrated in figure 5.3.15-1, consists of the following steps:

1. If the EM intends to query all VNF snapshots, it sends a GET request to the "VNF snapshots" resource.
2. The VNFM returns a "200 OK" response to the EM, and includes zero or more data structures of type "VnfSnapshotInfo" in the payload body.
3. If the EM intends to read information about a particular VNF snapshot, it sends a GET request to the "Individual VNF snapshot" resource, addressed by the appropriate VNF snapshot information identifier in its resource URI.
4. The VNFM returns a "200 OK" response to the EM, 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.

5.3.16 Flow of the deletion of a VNF snapshot resource

This clause describes the procedure for the deletion of a VNF/VNFC snapshot resource.

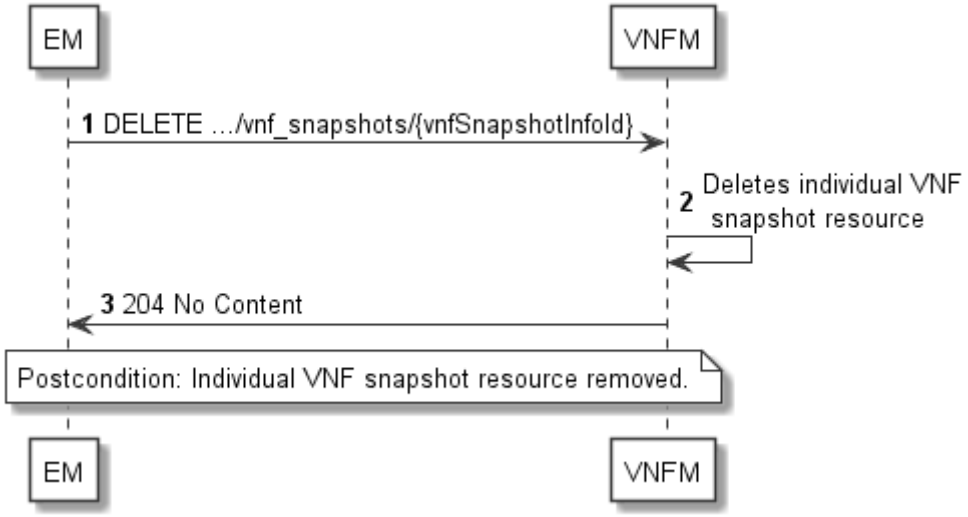

Figure 5.3.16-1: Flow of the deletion of a VNF snapshot resource

The procedure consists of the following steps as illustrated in figure 5.3.16-1:

1. EM sends a DELETE request to the "Individual VNF snapshot" resource.
2. The VNFM deletes the VNF snapshot resource and the associated VNF snapshot.
3. The VNFM 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.

5.4 Resources

5.4.1 Introduction

5.4.1.1 Overview

This clause defines all the resources and methods provided by the VNF lifecycle management interface.

5.4.1.2 Task resources that trigger VNF LCM operations

A number of resources are defined as task resources to trigger VNF LCM operations that are potentially long-running (e.g. Instantiate VNF, Scale VNF). To represent each occurrence of such a VNF LCM operation, an "Individual VNF LCM operation occurrence" resource is created as defined in clause 5.4.13.

When successfully executing the POST method on a task resource that triggers a VNF 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 VNF LCM operation occurrence" resource as defined in clause 5.4.13 shall be created, which represents the underlying VNF LCM operation occurrence that is executed by the VNFM. The VNFM shall set the "operationState" in the representation of the "Individual VNF LCM operation occurrence" resource to "STARTING".

2) Notifications of type "VnfLcmOperationOccurrenceNotification" shall be triggered as part of executing the underlying VNF LCM operation occurrence as defined in clauses 5.5.2.17 and 5.6.2.

3) If the VNFM has successfully completed the underlying VNF LCM operation occurrence:

   a) It shall update the representation of the "Individual VNF 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 VNF 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 VNFM shall send a notification of type "VnfLcmOperationOccurrenceNotification" with the "operationState" attribute set to "COMPLETED" as defined in clause 5.6.2.

4) If executing the underlying VNF LCM operation occurrence by the VNFM has failed in the "STARTING" phase, the VNFM shall send a notification of type "VnfLcmOperationOccurrenceNotification" with the "operationState" attribute set to "ROLLED_BACK" as defined in clause 5.6.2. It shall also set the "operationState" attribute in the representation of the aforementioned "Individual VNF LCM operation occurrence" resource to the value "ROLLED_BACK".

5) If executing the underlying VNF LCM operation occurrence by the VNFM has failed with no option to recover, the VNFM shall send a notification of type "VnfLcmOperationOccurrenceNotification" with the "operationState" attribute set to "FAILED" as defined in clause 5.6.2. It shall also set the "operationState" attribute in the representation of the aforementioned "Individual VNF LCM operation occurrence" resource to the value "FAILED", and shall reflect, at its best knowledge, the changes performed during the LCM operation in the representation of that resource.

6) If executing the underlying VNF LCM operation occurrence by the VNFM has failed temporarily, the VNFM shall send a notification of type "VnfLcmOperationOccurrenceNotification" with the "operationState" attribute set to "FAILED_TEMP" as defined in clause 5.6.2. It shall also set the "operationState" attribute in the representation of the aforementioned "Individual VNF 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.

The preconditions and postconditions for a successful execution of each of the VNF lifecycle management operations triggered by the corresponding task resources shall be as defined in table 5.4.1.2-1.

Table 5.4.1.2-1: Preconditions, postconditions, and parameterization of the flow for different VNF lifecycle management operations

Operation	Precondition	Task	RequestStructure	Postcondition
Instantiate VNF	VNF instance created and in NOT_INSTANTIATED state	instantiate	InstantiateVnfRequest	VNF instance in INSTANTIATED state.
Scale VNF	VNF instance in INSTANTIATED state	scale	ScaleVnfRequest	VNF instance still in INSTANTIATED state and VNF has been scaled.
Scale VNF to Level	VNF instance in INSTANTIATED state	scale_to_level	ScaleVnfToLevel Request	VNF instance still in INSTANTIATED state and VNF has been scaled.
Change VNF flavour	VNF instance in INSTANTIATED state	change_flavour	ChangeVnfFlavour Request	VNF instance still in INSTANTIATED state and VNF deployment flavour changed.
Operate VNF	VNF instance in INSTANTIATED state	operate	OperateVnfRequest	VNF instance still in INSTANTIATED state and VNF operational state changed.
Heal VNF	VNF instance in INSTANTIATED state	heal	HealVnfRequest	VNF instance still in INSTANTIATED state.
Change external VNF connectivity	VNF instance in INSTANTIATED state	change_ext_conn	ChangeExtVnfConnectivityRequest	VNF instance still in INSTANTIATED state and external connectivity of the VNF is changed.
Change current VNF package	VNF instance in INSTANTIATED state	change_vnfpkg	ChangeCurrentVnfPkg Request	VNF instance still in INSTANTIATED state and is now based on another VNF package.
Create VNF snapshot	VNF instance in INSTANTIATED state and individual VNF snapshot resource is available	create_snapshot	CreateVnfSnapshotRequest	VNF instance still in INSTANTIATED state and a VNF snapshot has been created.
Revert to VNF snapshot	VNF instance in INSTANTIATED state	revert_to_snapshot	RevertToVnfSnapshot Request	VNF instance still in INSTANTIATED state and VNF has been reverted to the snapshot status.
Terminate VNF	VNF instance in INSTANTIATED state	terminate	TerminateVnfRequest	VNF instance in NOT_INSTANTIATED state.

5.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [6] are part of the VNF lifecycle management interface.

5.4.2 Resource: VNF instances

5.4.2.1 Description

This resource represents VNF instances. The API consumer can use this resource to create "Individual VNF instance" resources, and to query VNF instances.

5.4.2.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances

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 [6].
apiMajorVersion	See clause 5.1a.

5.4.2.3 Resource methods

5.4.2.3.1 POST

The POST method creates a new VNF instance resource based on a VNF package that is onboarded and in "ENABLED" state.

This method shall follow the provisions specified in 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.

As the result of successfully executing this method, a new "Individual VNF instance" resource as defined in clause 5.4.3 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 VnfIdentifierCreationNotification shall be triggered as part of successfully executing this method as defined in clause 5.5.2.18.

When initiating the creation of a VNF instance resource, the passed metadata values can differ from the default values for metadata, if any, declared in the VNFD.

The VNFM shall apply the "metadata" attributes in the "CreateVnfRequest" data structure in the payload body to the "metadata" attribute in the "VnfInstance" data structure on top of the default values that were obtained from the VNFD according to the rules of JSON Merge Patch (see IETF RFC 7396 [3]). For all metadata keys defined in the VNFD, the VNFM shall ensure that the content of the resulting "metadata" attributes is valid against the data type definitions in the VNFD. The absence of a metadata item that is marked "required" in the VNFD shall not be treated as an error. In case a "metadata" child attribute is not defined in the VNFD, the VNFM shall consider it valid in case it is a valid JSON structure.

In case of an error, the operation shall be rejected with a "422 Unprocessable Entity" error.

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	
	CreateVnfRequest	1	The VNF creation parameters, as defined in clause 5.5.2.3.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfInstance	1	201 Created	Shall be returned when a new "Individual VNF instance" resource and the associated VNF instance identifier has been created successfully. The response body shall contain a representation of the created VNF instance, as defined in clause 5.5.2.2. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created VNF instance.
	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 [6], 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 VNF package referenced by the "vnfdId" attribute in the "CreateVnfRequest" structure is not in the "ENABLED" state or does not exist. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

5.4.2.3.2 GET

The GET method queries information about multiple VNF instances.

This method shall follow the provisions specified in 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	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [6]. The VNFM shall support receiving this parameter as part of the URI query string. The EM may supply this parameter. All attribute names that appear in the VnfInstance and in data types referenced from it shall be supported by the VNFM 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 [6] for details. The VNFM 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 [6] for details. The VNFM 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 [6] for details. The VNFM should support this parameter.

Name	Cardinality	Description
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [6] for details. The VNFM shall support this parameter. The following attributes shall be excluded from the VnfInstance structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: - vnfConfigurableProperties - instantiatedVnfInfo - metadata - extensions
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] 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**
	VnfInstance	0..N	200 OK	Shall be returned when information about zero or more VNF instances has been queried successfully. The response body shall contain in an array the representations of zero or more VNF instances, as defined in clause 5.5.2.2. 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 [6], respectively. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] 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 [6].
	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 VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [6].
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

5.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.3 Resource: Individual VNF instance

5.4.3.1 Description

This resource represents an individual VNF instance. The API consumer can use this resource to modify and delete the underlying VNF instance, and to read information about the VNF instance.

5.4.3.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}

The base resource URI variables for this resource are 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 [6].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance. 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 instance" 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 VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.3.3.2 GET

The GET method retrieves information about a VNF instance by reading an "Individual VNF instance" resource.

This method shall follow the provisions specified in 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**
	VnfInstance	1	200 OK	Shall be returned when information about an individual VNF instance has been read successfully. The response body shall contain a representation of the VNF instance, as defined in clause 5.5.2.2.
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

5.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.3.3.4 PATCH

This method modifies an "Individual VNF instance" resource.

Changes to the VNF configurable properties are applied to the configuration in the VNF instance, and are reflected in the representation of this resource. Other changes are applied to the VNF instance information managed by the VNFM, and are reflected in the representation of this resource.

This method shall follow the provisions specified in 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.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 5.4.1.2.

The VNFM shall apply the "metadata", "extensions" and "vnfConfigurableProperties" attributes in the "VnfInfoModificationRequest" data structure in the payload body to the existing "extensions" and "vnfConfigurableProperties" attributes from the "VnfInstance" data structure according to the rules of JSON Merge Patch (see IETF RFC 7396 [3]).

The VNFM shall ensure that the content of the child attributes of the resulting "metadata", "extensions" and "vnfConfigurableProperties" attributes is valid against the data type definitions of these child attributes in the VNFD.

In case a "metadata" child attribute is not defined in the VNFD, the VNFM shall consider it valid in case it is a valid JSON structure.

NOTE 1: "Extensions" and "vnfConfigurableProperties" child attributes are always declared in the VNFD.

If the VNF instance is in the "INSTANTIATED" state, the validation shall also include ensuring the presence of all "extensions" and "vnfConfigurableProperties" child attributes that are marked as "required" in the VNFD.

NOTE 2: This allows to build the set of "extensions" and "vnfConfigurableProperties" incrementally prior VNF instantiation but ensures their completeness for an instantiated VNF instance.

The absence of a metadata item that is marked "required" in the VNFD shall not be treated as an error.

In case of an error during validation, the operation shall be automatically rolled back, and appropriate error information shall be provided in the "VnfLcmOperationOccurrenceNotification" message and the "VnfLcmOpOcc" data structure. The processing of changes to the "metadata"/"extensions"/"vnfConfigurableProperties" attributes shall be performed in the "PROCESSING" phase of the LCM operation. The change shall be atomic, i.e. the result of intermediate stages shall not be visible in the API. In case of successful completion of the processing and validation, the attributes shall be provided in the "VnfInstance" data structure and the operation shall proceed towards successful completion.

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

	Data type	Cardinality	Description	
Request body	VnfInfoModificationRequest	1	Parameters for the VNF modification, as defined in clause 5.5.2.12. The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [3].	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. On success, the HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation. 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 VNF instance" resource. Typically, this is due to the fact that another LCM operation is ongoing. 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 [6]	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 [6] may be returned.

5.4.3.3.5 DELETE

This method deletes an "Individual VNF instance" resource.

This method shall follow the provisions specified in 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.

As the result of successfully executing this method, the "Individual VNF instance" resource shall not exist any longer. A notification of type "VnfIdentifierDeletionNotification" shall be triggered as part of successfully executing this method as defined in clause 5.5.2.19.

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	Shall be returned when the "Individual VNF instance" resource and the associated VNF identifier were 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 "Individual VNF instance" resource is in INSTANTIATED state. 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 [6]	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 [6] may be returned.

5.4.4 Resource: Instantiate VNF task

5.4.4.1 Description

This task resource represents the "Instantiate VNF" operation. The API consumer can use this resource to instantiate a VNF instance.

5.4.4.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/instantiate

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 [6].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	The identifier of the VNF instance to be instantiated. 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 instance" 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

The POST method instantiates a VNF instance.

This method shall follow the provisions specified in tables 5.4.4.3.1-1 and 5.4.4.3.1-2 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 5.4.1.2.

In addition, once the VNFM has successfully completed the underlying VNF LCM operation occurrence, it shall set the "instantiationState" attribute to the value "INSTANTIATED" and the "vnfState" attribute to the value "STARTED" in the representation of the "Individual VNF instance" resource.

When instantiating a VNF instance, the values of the extensions and/or VNF configurable properties passed in the instantiation request can differ from the values in the "VnfInstance" data structure that were initialized from default values, if any, declared in the VNFD.

The VNFM shall apply the "extensions" and "vnfConfigurableProperties" attributes in the "InstantiateVnfRequest" data structure in the payload body to the existing "extensions" and "vnfConfigurableProperties" attributes from the "VnfInstance" data structure according to the rules of JSON Merge Patch (see IETF RFC 7396 [3]). The VNFM shall ensure that the content of the resulting "extensions" and "vnfConfigurableProperties" attributes is valid against the VNFD (including the presence of all child attributes that are marked as "required" in the VNFD). In case of an error during validation, the operation shall be automatically rolled back, and appropriate error information shall be provided in the "VnfLcmOperationOccurrenceNotification" message and the "VnfLcmOpOcc" data structure. The processing of changes to the "extensions"/"vnfConfigurableProperties" attributes shall be performed in the "STARTING" phase of the LCM operation. The change shall be atomic, i.e. the result of intermediate stages shall not be visible in the API. In case of successful completion of the processing and validation, the attributes shall be provided in the "VnfInstance" data structure and the operation shall proceed to obtain the grant.

Table 5.4.4.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.4.3.1-2: Details of the POST request/response on this resource

Request body	**Data type**	**Cardinality**	**Description**	
	InstantiateVnfRequest	1	Parameters for the VNF instantiation, as defined in clause 5.5.2.4.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation.
	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 "Individual VNF instance" resource is in INSTANTIATED state or that a required (see note) child attribute of the "extensions" attribute has not been set. 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 [6]	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 [6] may be returned.
NOTE: Required attributes are marked as "required" in the VNFD.				

5.4.4.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.5 Resource: Scale VNF task

5.4.5.1 Description

This task resource represents the "Scale VNF" operation. The API consumer can use this resource to request scaling a VNF instance.

It depends on the VNF capabilities, and is declared in the VNFD, whether this operation is supported for a particular VNF.

See clause B.2 for an explanation of VNF scaling.

5.4.5.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/scale

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 [6].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance to be scaled. 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 instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.5.3 Resource methods

5.4.5.3.1 POST

The POST method requests to scale a VNF instance resource incrementally.

This method shall follow the provisions specified in 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.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 5.4.1.2.

In addition, once the VNFM has successfully completed the underlying VNF LCM operation occurrence, it shall reflect the result of scaling the VNF instance by updating the "scaleStatus" attribute in the representation of the "Individual VNF instance" resource.

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**	
	ScaleVnfRequest	1	Parameters for the scale VNF operation, as defined in clause 5.5.2.5.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation.
	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 [6], 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 VNF instance 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	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 "Individual VNF instance" resource is in NOT_INSTANTIATED state, that another lifecycle management operation is ongoing, or that a required (see note) child attribute of the "extensions" attribute has not been set. 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 [6]	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 [6] may be returned.
NOTE: Required attributes are marked as "required" in the VNFD.				

5.4.5.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.6 Resource: Scale VNF to Level task

5.4.6.1 Description

This task resource represents the "Scale VNF to Level" operation. The API consumer can use this resource to request scaling of a VNF instance to a target level.

It depends on the VNF capabilities, and is declared in the VNFD, whether this operation is supported for a particular VNF.

See clause B.2 for an explanation of VNF scaling.

5.4.6.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/scale_to_level

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 [6].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance to be scaled to a target level. 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 instance" 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

The POST method requests to scale a VNF instance resource to a target level.

This method shall follow the provisions specified in tables 5.4.6.3.1-1 and 5.4.6.3.1-2 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 5.4.1.2.

In addition, once the VNFM has successfully completed the underlying VNF LCM operation occurrence, it shall reflect the result of scaling the VNF instance by updating the "scaleStatus" attribute in the representation of the "Individual VNF instance" resource.

Table 5.4.6.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.6.3.1-2: Details of the POST request/response on this resource

Request body	**Data type**	**Cardinality**	**Description**	
	ScaleVnfToLevelRequest	1	Parameters for the scale VNF to Level operation, as defined in clause 5.5.2.6.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation.
	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 [6], 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 VNF instance 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	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 VNF instance resource is in NOT_INSTANTIATED state, that another lifecycle management operation is ongoing, or that a required (see note) child attribute of the "extensions" attribute has not been set. 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 [6]	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 [6] may be returned.
NOTE: Required attributes are marked as "required" in the VNFD.				

5.4.6.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.6.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.6.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.7 Resource: Change VNF Flavour task

5.4.7.1 Description

This task resource represents the "Change VNF Flavour" operation. The API consumer can use this resource to change the deployment flavour for a VNF 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.

5.4.7.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/change_flavour

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 [6].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	The identifier of the VNF instance of which the deployment flavour is requested to be changed. 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 instance" 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

The POST method changes the deployment flavour of a VNF instance.

This method shall follow the provisions specified in tables 5.4.7.3.1-1 and 5.4.7.3.1-2 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 5.4.1.2.

In addition, once the VNFM has successfully completed the underlying VNF LCM operation occurrence, it shall set the "flavourId" attribute in the representation of the "Individual VNF instance" resource to the value of the "newFlavourId" attribute passed in the "ChangeVnfFlavourRequest" data in the POST request.

When initiating a change of the current VNF flavour, the values of the extensions and/or VNF configurable properties, can differ between the new flavour and the old flavour of the VNF instance.

The VNFM shall apply the "extensions" and "vnfConfigurableProperties" attributes in the "ChangeVnfFlavourRequest" data structure in the payload body to the existing "extensions" and "vnfConfigurableProperties" attributes from the "VnfInstance" data structure according to the rules of JSON Merge Patch (see IETF RFC 7396 [3]). The VNFM shall ensure that the content of the resulting "extensions" and "vnfConfigurableProperties" attributes is valid against the VNFD (which includes ensuring the presence of all child attributes that are marked as "required" in the VNFD). In case of an error, the operation shall be automatically rolled back, and appropriate error information shall be provided in the "VnfLcmOperationOccurrenceNotification" message and the "VnfLcmOpOcc" data structure. The processing of changes to the "extensions"/"vnfConfigurableProperties" attributes shall be performed in the "STARTING" phase of the LCM operation. The change shall be atomic, i.e. the result of intermediate stages shall not be visible in the API. In case of successful completion of the processing and validation, the attributes shall be provided in the "VnfInstance" data structure and the operation shall proceed to obtain the grant.

Table 5.4.7.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.7.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	ChangeVnfFlavourRequest	1	Parameters for the Change VNF Flavour operation, as defined in clause 5.5.2.7.	
Response body	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation.
	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 [6], 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 VNF instance 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	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 "Individual VNF instance" resource is in NOT_INSTANTIATED state, that another lifecycle management operation is ongoing, or that a required (see note) child attribute of the "extensions" attribute has not been set. 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 [6]	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 [6] may be returned.
NOTE: Required attributes are marked as "required" in the VNFD.				

5.4.7.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.7.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.7.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.7.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.8 Resource: Terminate VNF task

5.4.8.1 Description

This task resource represents the "Terminate VNF" operation. The API consumer can use this resource to terminate a VNF instance.

5.4.8.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/terminate

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 [6].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	The identifier of the VNF instance to be terminated. 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 instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.8.3 Resource methods

5.4.8.3.1 POST

The POST method triggers the VNFM to terminate a VNF instance and to request to the VIM the release of its used virtualised resources.

This method shall follow the provisions specified in tables 5.4.8.3.1-1 and 5.4.8.3.1-2 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 5.4.1.2.

In addition, once the VNFM has successfully completed the underlying VNF LCM operation occurrence, it shall set the "instantiationState" attribute in the representation of the "Individual VNF instance" resource to the value "NOT_INSTANTIATED".

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

Name	Cardinality	Description
none supported		

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

Request body	Data type	Cardinality	Description	
	TerminateVnfRequest	1	Parameters for the VNF termination, as defined in clause 5.5.2.8.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation.
	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 "Individual VNF instance" resource is in NOT_INSTANTIATED state, or that another lifecycle management operation is ongoing, or that a required (see note) child attribute of the "extensions" attribute has not been set. 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 [6]	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 [6] may be returned.
NOTE: Required attributes are marked as "required" in the VNFD.				

5.4.8.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.8.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.8.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.8.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.9 Resource: Heal VNF task

5.4.9.1 Description

This task resource represents the "Heal VNF" operation. The API consumer can use this resource to request healing a VNF instance.

It depends on the VNF capabilities, and is declared in the VNFD, whether this operation is supported for a particular VNF.

5.4.9.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/heal

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 [6].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance to be healed. 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 instance" 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

The POST method requests to heal a VNF instance.

This method shall follow the provisions specified in tables 5.4.9.3.1-1 and 5.4.9.3.1-2 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 5.4.1.2.

Table 5.4.9.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.9.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	HealVnfRequest	1	Parameters for the Heal VNF operation, as defined in clause 5.5.2.9.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing, but the processing has not been completed. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation.
	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 [6], 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 VNF instance 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	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 "Individual VNF instance" resource is in NOT_INSTANTIATED state, that another lifecycle management operation is ongoing, or that a required (see note) child attribute of the "extensions" attribute has not been set. 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 [6]	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 [6] may be returned.
NOTE: Required attributes are marked as "required" in the VNFD.				

5.4.9.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.9.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.9.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.9.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.10 Resource: Operate VNF task

5.4.10.1 Description

This task resource represents the "Operate VNF" operation. The API consumer can use this resource to operate a VNF instance.

It depends on the VNF capabilities, and is declared in the VNFD, whether this operation is supported for a particular VNF.

The "Operate VNF" operation enables requesting to change the operational state of a VNF instance, including starting and stopping the VNF instance.

NOTE 1: These operations are complementary to instantiating and terminating a VNF.

NOTE 2: In the present document, only starting and stopping the VNF instances is supported. Extension of this operation to support other VNF state changes is left for future specification.

A VNF instance can be in the following states:

- STARTED: the VNF instance is up and running.
- STOPPED: the VNF instance has been shut down, i.e. all its VNFC instances have been stopped.

In the state STOPPED, the virtualisation containers, where the VNFC instances of the VNF run, are shut down but not deleted. In addition, if the workflow requires a graceful stop, as part of this process the VNFM (API producer) will interact with VNF/EM to gracefully stop the VNF application. Once a VNF is instantiated, i.e. all instantiation steps have been completed, the VNF instance is in the state STARTED.

5.4.10.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/operate

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
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance to be operated. 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 instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.10.3 Resource methods

5.4.10.3.1 POST

The POST method changes the operational state of a VNF instance.

This method shall follow the provisions specified in tables 5.4.10.3.1-1 and 5.4.10.3.1-2 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 5.4.1.2.

In addition, once the VNFM has successfully completed the underlying VNF LCM operation occurrence, it shall set the "vnfState" attribute in the representation of the "Individual VNF instance" resource to the value of the "changeStateTo" attribute passed in the "OperateVnfRequest" data in the POST request.

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

Name	Cardinality	Description
none supported		

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

	Data type	Cardinality	Response Codes	Description
Request body	OperateVnfRequest	1	Parameters for the Operate VNF operation, as defined in clause 5.5.2.10.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation.
	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 [6], 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 VNF instance 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	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 VNF instance resource is in NOT_INSTANTIATED state, that another lifecycle management operation is ongoing, or that a required (see note) child attribute of the "extensions" attribute has not been set. 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 [6]	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 [6] may be returned.
NOTE: Those attributes are marked as "required" in the VNFD.				

5.4.10.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.10.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.10.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.10.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.11 Resource: Change external VNF connectivity task

5.4.11.1 Description

This task resource represents the "Change external VNF connectivity" operation. The API consumer can use this resource to change the external connectivity of a VNF instance. The types of changes that this operation supports are:

- Disconnect external CPs that are connected to a particular external VL, and connect them to a different external VL.
- 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 [7]. 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.
- 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.

- 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 [7], and connect them to a particular external VL. Creation of the parent port with this operation is not supported.

VNFs shall support this operation. This operation may be service-disruptive.

5.4.11.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/change_ext_conn

This resource shall support the resource URI variables defined in table 5.4.11.2-1.

Table 5.4.11.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance of which the external connectivity is requested to be changed. 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 instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.11.3 Resource methods

5.4.11.3.1 POST

The POST method changes the external connectivity of a VNF instance.

This method shall follow the provisions specified in tables 5.4.11.3.1-1 and 5.4.11.3.1-2 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 5.4.1.2.

Table 5.4.11.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.11.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	ChangeExtVnfConnectivityRequest	1		Parameters for the Change external VNF connectivity operation, as defined in clause 5.5.2.11.
Response body	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the instantiation operation.
	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 another lifecycle management operation is ongoing, or that a required (see note) child attribute of the "extensions" attribute has not been set. 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 [6]	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 [6] may be returned.
NOTE: Required attributes are marked as "required" in the VNFD.				

5.4.11.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.11.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.11.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.11.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.11a Resource: Change current VNF package task

5.4.11a.1 Description

This operation enables the API consumer to request the VNFM to change the current VNF Package, i.e. the VNF package on which a VNF instance is based.

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. 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.

As part of changing the current VNF Package, the VNFM shall be capable to add temporary virtualised resources used in the modification process, e.g. virtualised resources for a VNFC which will be responsible for handling or supporting the change of the current VNF Package process. The need for temporary virtualised resources shall be indicated as "tempResource" during the VNF LCM operation granting exchange. In addition, the VNFM shall be capable to add and remove virtualised resources as required for the "change of current VNF Package" process. The need for addition and removal of existing virtualised resources shall be indicated as "addResource" and "removeResource" in the VNF LCM operation granting exchange.

The following applies to the existing resources of the VNF instance: In the course of the successful execution of this operation, the VNFM shall replace or update those resources of the VNF instance that are based on descriptors (e.g. VDUs, VLDs, CPDs) that have changed between source and destination VNFD to align them with the updated descriptors, with the only allowed exception that the references to software images need not be updated if the resources are not replaced. Further, the VNFM shall remove resources that relate to descriptors in the source VNFD that have no corresponding descriptor in the destination VNFD. For newly-created resources, the VNFM shall use the descriptors of the destination VNFD.

All VNFs shall support this operation. This operation may be service-disruptive.

It is declared in the VNFD whether a change from a particular "source" VNF package to a particular "destination" VNF package is possible. The evaluation of this information shall take place in the "STARTING" phase of the LCM operation. In case the evaluation shows that such change is not possible, the operation shall be automatically rolled back.

In the representation of the VNF instance (see clause 5.5.2.2), there are a number of structures that relate to a particular VNFD, which is reflected by these structures having an attribute of type "IdentifierInVnfd". During the course of the execution of this operation, or due to its final failure, these structures may either refer to the source VNFD or to the destination VNFD of the operation and are accompanied by a "vnfdId" attribute to signal which VNFD they relate to. If that attribute is present, it signals the VNFD that applies to the data structure. If that attribute is absent and the operation is in the "STARTING" phase, the source VNF package is referenced by default. If that attribute is absent and the operation is in any of the phases after "STARTING", the destination VNF package is referenced by default.

5.4.11a.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/change_ vnfpkg

This resource shall support the resource URI variables defined in table 5.4.11a.2-1.

Table 5.4.11a.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance of which the underlying VNF package is requested to be changed. 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 instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.11a.3 Resource methods

5.4.11a.3.1 POST

The POST method changes the current VNF package on which the VNF instance is based.

This method shall follow the provisions specified in tables 5.4.11a.3.1-1 and 5.4.11a.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 5.4.1.2.

During a change of the current VNF package, the allowed and required extensions and/or VNF configurable properties and their data types, as well as the metadata data types, can differ between the source and the destination VNFD.

The VNFM shall process the child attributes of extensions and VNF configurable properties during the execution of the "Change current VNF package" as follows:

1) First, "extensions" and "vnfConfigurableProperties" child attributes which are not defined in the source VNFD but are defined in the destination VNFD with initial values shall be created automatically and shall be populated by these values.

2) Second, the "extensions" and "vnfConfigurableProperties" attributes in the "ChangeCurrentVnfPkgRequest" data structure in the payload body shall be applied to the existing "extensions" and "vnfConfigurableProperties" attributes from the "VnfInstance" data structure according to the rules of JSON Merge Patch (see IETF RFC 7396 [3]):

    a) For those "extensions" and "vnfConfigurableProperties" child attributes that were already defined in the source VNFD and of which the data type has changed in the destination VNFD and whose current value is not compatible with the new data type, input information is expected to be provided by the API

consumer in a way that is compatible with applying the new information on top of the current value using JSON Merge Patch.

b) For those new "extensions" and "vnfConfigurableProperties" child attributes that are not defined in the source VNFD but are defined in the destination VNFD without initial values and that are required, all information needed to populate them is expected to be provided by the API consumer.

c) Additional changed values can be provided by the API consumer.

3) To clean up, "extensions" and "vnfConfigurableProperties" child attributes that are no longer supported in the destination VNFD and that have not been deleted by explicit input shall be deleted automatically by the VNFM.

4) The VNFM shall validate the resulting "extensions" and "vnfConfigurableProperties" against the destination VNFD (which includes ensuring the presence of all child attributes that are marked as "required" in the VNFD). In case of an error, the operation shall be automatically rolled back, and appropriate error information shall be provided in the "VnfLcmOperationOccurrenceNotification" message and the "VnfLcmOpOcc" data structure.

In addition, the VNFM shall process the metadata attributes during the execution of the "Change current VNF package" as follows:

NOTE 1: Metadata changes can not be signalled as part of invoking the "Change current VNF package" operation.

1) "Metadata" child attributes which are not defined in the source VNFD, are defined in the destination VNFD with initial values and does not exist in the VNF instance shall be created automatically and shall be populated by these initial values.

2) "Metadata" child attributes that are defined in the source VNFD but not in the destination VNFD shall be kept in the "VnfInstance" data structure unchanged.

3) "Metadata" child attributes that are not defined in the source VNFD, are defined in the destination VNFD and exist in the "VnfInstance" structure at the time when the "Change current VNF package" operation is in the "STARTING" phase shall be handled as follows:

a) If these child attributes in the VnfInstance have a data type compatible with the definition in the destination VNFD, the VNFM shall keep the existing value.

b) If these child attributes in the VnfInstance have a data type *in*compatible with the definition in the destination VNFD, the VNFM shall automatically roll back the operation and shall provide appropriate error information in the "VnfLcmOperationOccurrenceNotification" message and the "VnfLcmOpOcc" data structure.

4) "Metadata" child attributes that are defined in both the source VNFD and the destination VNFD and whose data type is changed in the destination VNFD compared to the data type defined in the source VNFD, in a way that validation would fail against the data type definition of that attribute in the destination VNFD, shall be handled as follows: The VNFM shall automatically roll back the operation and shall provide appropriate error information in the "VnfLcmOperationOccurrenceNotification" message and the "VnfLcmOpOcc" data structure.

NOTE 2: To address the cases 3b and 4, the API consumer can delete the affected colliding "metadata" child attributes or update their content by using the PATCH operation on the "individual VNF instance" resource prior to invoking the "Change current VNF package" operation.

The validation that the changes to the "extensions"/"vnfConfigurableProperties"/"metadata" attributes can be processed without issues shall be performed in the "STARTING" phase of the LCM operation. In case of successful completion of the validation, the operation shall proceed to obtain the grant.

Further, in the "VnfExtCpData" structure under the "ExtVirtualLinkData" structure in the "ChangeCurrentVnfPkgRequest", the API consumer need not explicitly delete (by setting them to null) those "cpConfig" entries that relate to CPDs which are present in the source VNFD but not in the destination VNFD. Before the successful completion of the operation, the VNFM shall remove these entries from the list that is exposed in the "currentVnfExtCpData" attribute of the "ExtVirtualLinkInfo".

Further, in the "extManagedVirtualLinks" attribute in the "ChangeCurrentVnfPkgRequest", the API consumer may still provide those entries that relate to virtual link descriptors which are present in the source VNFD but not in the destination VNFD. Before the successful completion of the operation, the VNFM shall remove these entries from the list that is exposed in the "extManagedVirtualLinks" attribute of the "VnfInstance" structure.

Table 5.4.11a.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.11a.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	ChangeCurrentVnfPkgRequest	1	Parameters for the Change current VNF package operation, as defined in clause 5.5.2.11a.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the instantiation operation.
	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 another lifecycle management operation is ongoing. 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 [6]	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 [6] may be returned.

5.4.11a.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.11a.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.11a.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.11a.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.12 Resource: VNF LCM operation occurrences

5.4.12.1 Description

This resource represents VNF lifecycle management operation occurrences. The API consumer can use this resource to query status information about multiple VNF lifecycle management operation occurrences.

5.4.12.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_lcm_op_occs

The base resource URI variables for this resource are defined in table 5.4.12.2-1.

Table 5.4.12.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 5.1a.

5.4.12.3 Resource methods

5.4.12.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.12.3.2 GET

The API consumer can use this method to query status information about multiple VNF lifecycle management operation occurrences.

This method shall follow the provisions specified in tables 5.4.12.3.2-1 and 5.4.12.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.12.3.2-1: 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 [6]. The VNFM shall support receiving this parameter as part of the URI query string. The EM/VNF may supply this parameter. All attribute names that appear in the VnfLcmOpOcc and in data types referenced from it shall be supported by the VNFM 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 [6] for details. The VNFM 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 [6] for details. The VNFM 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 [6] for details. The VNFM should support this parameter.

Name	Cardinality	Description
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [6] for details. The VNFM shall support this parameter. The following attributes shall be excluded from the VnfLcmOpOcc structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: - operationParams - error - resourceChanges - changedInfo - changedExtConnectivity - lcmCoordinations - modificationsTriggeredByVnfPkgChange - warnings
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource.

Table 5.4.12.3.2-2: 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**
	VnfLcmOpOcc	0..N	200 OK	Shall be returned when status information for zero or more VNF lifecycle management operation occurrences has been queried successfully. The response body shall contain in an array the status information about zero or more VNF lifecycle operation occurrences, as defined in clause 5.5.2.13. 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 [6], respectively. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] 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 [6].
	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 VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [6].
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

5.4.12.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.12.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.12.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.13 Resource: Individual VNF LCM operation occurrence

5.4.13.1 Description

This resource represents a VNF lifecycle management operation occurrence. The API consumer can use this resource to read status information about an individual VNF 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 VNFM may remove an "Individual VNF LCM operation occurrence" resource some time 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 VNFM waits before deleting such a resource is defined by means outside the scope of the present document.

5.4.13.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_lcm_op_occs/{vnfLcmOpOccId}

The base resource URI variables for this resource are defined in table 5.4.13.2-1.

Table 5.4.13.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 5.1a.
vnfLcmOpOccId	Identifier of a VNF lifecycle management operation occurrence. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a PATCH or POST request triggering a VNF LCM operation. It can also be retrieved from the "vnfLcmOpOccId" attribute in the VnfLcmOperationOccurrenceNotification.	

5.4.13.3 Resource methods

5.4.13.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.13.3.2 GET

The API consumer can use this method to retrieve status information about a VNF lifecycle management operation occurrence by reading an "Individual VNF LCM operation occurrence" resource.

This method shall follow the provisions specified in tables 5.4.13.3.2-1 and 5.4.13.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.13.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.13.3.2-2: 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**
	VnfLcmOpOcc	1	200 OK	Shall be returned when information about a VNF LCM operation occurrence has been read successfully. The response body shall contain status information about a VNF lifecycle management operation occurrence (see clause 5.5.2.13).
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

5.4.13.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.13.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.13.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

See clause 5.4.13.1 for a definition related to the removal of an "Individual VNF LCM operation occurrence" resource.

5.4.14 Resource: Retry operation task

5.4.14.1 Description

This task resource represents the "Retry operation" operation. The API consumer can use this resource to initiate retrying a VNF lifecycle operation that is in a transient failure state. See also clause 5.6.2.3.

5.4.14.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_lcm_op_occs/{vnfLcmOpOccId}/retry

This resource shall support the resource URI variables defined in table 5.4.14.2-1.

Table 5.4.14.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 5.1a.
vnfLcmOpOccId	Identifier of a VNF 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 PATCH or POST request triggering a VNF LCM operation. It can also be retrieved from the "vnfLcmOpOccId" attribute in the VnfLcmOperationOccurrenceNotification.	

5.4.14.3 Resource methods

5.4.14.3.1 POST

The POST method initiates retrying a VNF lifecycle operation if that operation has experienced a temporary failure, i.e. the related "Individual VNF LCM operation occurrence" resource is in "FAILED_TEMP" state.

This method shall follow the provisions specified in tables 5.4.14.3.1-1 and 5.4.14.3.1-2 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 5.6.2.2 shall be emitted to indicate that the underlying VNF LCM operation occurrence proceeds.

Table 5.4.14.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.14.3.1-2: 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**
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. 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 [6], 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 VNF 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	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the VNF LCM operation occurrence. Typically, this is due to the fact that the VNF LCM operation occurrence is not in FAILED_TEMP state or another error handling action is starting such as rollback or fail. 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 [6]	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 [6] may be returned.

5.4.14.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.14.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.14.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.14.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.15 Resource: Rollback operation task

5.4.15.1 Description

This task resource represents the "Rollback operation" operation. The API consumer can use this resource to initiate rolling back a VNF lifecycle operation. See also clause 5.6.2.3.

5.4.15.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_lcm_op_occs/{vnfLcmOpOccId}/rollback

This resource shall support the resource URI variables defined in table 5.4.15.2-1.

Table 5.4.15.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 5.1a.
vnfLcmOpOccId	Identifier of a VNF 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 PATCH or POST request triggering a VNF LCM operation. It can also be retrieved from the "vnfLcmOpOccId" attribute in the VnfLcmOperationOccurrenceNotification.	

5.4.15.3 Resource methods

5.4.15.3.1 POST

The POST method initiates rolling back a VNF lifecycle operation if that operation has experienced a temporary failure, i.e. the related "Individual VNF LCM operation occurrence" resource is in "FAILED_TEMP" state. In case of rolling back an occurrence of the "InstantiateVnf" operation, the VNFM shall request to the VIM the release of the virtualised resources that were allocated for the related VNF instance. The "rollback" task shall be supported by the VNFM for any VNF LCM operation occurrence that represents an "InstantiateVnf" operation in FAILED_TEMP state.

This method shall follow the provisions specified in tables 5.4.15.3.1-1 and 5.4.15.3.1-2 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 5.6.2.2 shall be emitted to indicate that rollback of the underlying VNF LCM operation occurrence is attempted.

Table 5.4.15.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.15.3.1-2: 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. 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 [6], 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 VNF 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	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the VNF LCM operation occurrence. Typically, this is due to the fact that the VNF LCM operation occurrence is not in FAILED_TEMP state or another error handling action is starting such as retry or fail. 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 [6]	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 [6] may be returned.

5.4.15.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.15.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.15.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.15.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.16 Resource: Fail operation task

5.4.16.1 Description

This task resource represents the "Fail operation" operation. The API consumer can use this resource to mark a VNF lifecycle management operation occurrence as "finally failed", i.e. change the state of the related VNF LCM operation occurrence 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 5.6.2.3.

5.4.16.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_lcm_op_occs/{vnfLcmOpOccId}/fail

This resource shall support the resource URI variables defined in table 5.4.16.2-1.

Table 5.4.16.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 5.1a.
vnfLcmOpOccId	Identifier of a VNF 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 PATCH or POST request triggering a VNF LCM operation. It can also be retrieved from the "vnfLcmOpOccId" attribute in the VnfLcmOperationOccurrenceNotification.	

5.4.16.3 Resource methods

5.4.16.3.1 POST

The POST method marks a VNF lifecycle management operation occurrence as "finally failed" if that operation occurrence is in "FAILED_TEMP" state.

This method shall follow the provisions specified in tables 5.4.16.3.1-1 and 5.4.16.3.1-2 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 5.6.2.2 shall be emitted to indicate that the execution of the underlying VNF LCM operation occurrence has finally and unrecoverably failed.

Table 5.4.16.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.16.3.1-2: 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**
	VnfLcmOpOcc	1	200 OK	Shall be returned when the state of the VNF lifecycle management operation occurrence has been changed successfully. The response body shall include a representation of the "Individual VNF lifecycle operation occurrence" resource.
	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 [6], 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 VNF 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	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the VNF LCM operation occurrence. Typically, this is due to the fact that the VNF LCM operation occurrence is not in FAILED_TEMP state or another error handling action is starting such as retry or rollback. 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 [6]	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 [6] may be returned.

5.4.16.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.16.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.16.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.16.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.17 Resource: Cancel operation task

5.4.17.1 Description

This task resource represents the "Cancel operation" operation. The API consumer can use this resource to cancel an ongoing VNF lifecycle operation. See also clause 5.6.2.3.

5.4.17.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_lcm_op_occs/{vnfLcmOpOccId}/cancel

This resource shall support the resource URI variables defined in table 5.4.17.2-1.

Table 5.4.17.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 5.1a.
vnfLcmOpOccId	Identifier of a VNF 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 PATCH or POST request triggering a VNF LCM operation. It can also be retrieved from the "vnfLcmOpOccId" attribute in the VnfLcmOperationOccurrenceNotification.	

5.4.17.3 Resource methods

5.4.17.3.1 POST

The POST method initiates cancelling an ongoing VNF lifecycle operation while it is being executed or rolled back, i.e. the related "Individual VNF LCM operation occurrence" resource is either in "STARTING" or "PROCESSING" or "ROLLING_BACK" state.

This method shall follow the provisions specified in tables 5.4.17.3.1-1 and 5.4.17.3.1-2 for URI query parameters, request and response data structures, and response codes.

Before returning the "202 Accepted" response, the VNFM shall update the "isCancelPending" and "cancelMode" attributes in the representation of the parent resource according to the provisions in clause 5.5.2.13.

In case of success of processing the asynchronous request:

1) If the request has been processed in "STARTING" state, the "operationState" attribute in the representation of the parent resource shall be changed to "ROLLED_BACK".

2) 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 VNFM shall update the "isCancelPending" and "cancelMode" attributes in the representation of the parent resource according to the provisions in clause 5.5.2.13 to reflect the new status, and the applicable "result" notification according to clause 5.6.2.2 shall be emitted to indicate that the execution of the underlying VNF 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 5.4.17.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.17.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Description	
Request body	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	Shall be returned when the request has been accepted for processing. 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 [6], 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 VNF 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	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the VNF LCM operation occurrence. Typically, this is due to the fact that the operation occurrence is not in STARTING, PROCESSING or ROLLING_BACK state. 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 [6]	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 [6] may be returned.

5.4.17.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.17.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.17.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.17.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.18 Resource: Subscriptions

5.4.18.1 Description

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

5.4.18.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/subscriptions

This resource shall support the resource URI variables defined in table 5.4.18.2-1.

Table 5.4.18.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 5.1a.

5.4.18.3 Resource methods

5.4.18.3.1 POST

The POST method creates a new subscription.

This method shall follow the provisions specified in tables 5.4.18.3.1-1 and 5.4.18.3.1-2 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 5.4.19 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 EM or VNF, and might make sense only in very rare use cases. Consequently, the VNFM may either allow creating an "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 5.4.18.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.18.3.1-2: Details of the POST request/response on this resource

Request body	**Data type**	**Cardinality**	**Description**	
	LccnSubscriptionRequest	1	Details of the subscription to be created, as defined in clause 5.5.2.15.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	LccnSubscription	1	201 Created	Shall be returned when the subscription has been created successfully. The response body shall contain a representation of the created "Individual subscription" resource. The HTTP response shall include a "Location:" HTTP header that points to the created "Individual subscription" resource.
	n/a		303 See Other	Shall be returned if a subscription with the same callback URI and the same filter already exists and the policy of the VNFM is to not create redundant subscriptions. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource. The response body shall be empty.
	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 [6], 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 VNFM has tested the Notification endpoint as described in clause 5.4.20.3.2 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 [6]	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 [6] may be returned.

5.4.18.3.2 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 tables 5.4.18.3.2-1 and 5.4.18.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.18.3.2-1: 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 [6]. The VNFM shall support receiving this parameter as part of the URI query string. The EM may supply this parameter. All attribute names that appear in the LccnSubscription and in data types referenced from it shall be supported by the VNFM in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource.

Table 5.4.18.3.2-2: 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**
	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 5.5.2.16. 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 [6]. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] 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 [6].
	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 VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [6].
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

5.4.18.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.18.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.18.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.19 Resource: Individual subscription

5.4.19.1 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 VNF lifecycle management.

5.4.19.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/subscriptions/{subscriptionId}

This resource shall support the resource URI variables defined in table 5.4.19.2-1.

Table 5.4.19.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
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 "Individual subscription" resource, It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.19.3 Resource methods

5.4.19.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.19.3.2 GET

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

This method shall follow the provisions specified in tables 5.4.19.3.2-1 and 5.4.19.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.19.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.19.3.2-2: 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**
	LccnSubscription	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 [6]	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 [6] may be returned.

5.4.19.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.19.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.19.3.5 DELETE

The DELETE method terminates an individual subscription.

This method shall follow the provisions specified in tables 5.4.19.3.5-1 and 5.4.19.3.5-2 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 5.4.19.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.19.3.5-2: 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 [6]	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 [6] may be returned.

5.4.20 Resource: Notification endpoint

5.4.20.1 Description

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

5.4.20.2 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 5.4.20.2-1.

Table 5.4.20.2-1: Resource URI variables for this resource

Name	Definition
none supported	

5.4.20.3 Resource methods

5.4.20.3.1 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 tables 5.4.20.3.1-1 and 5.4.20.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.20.3.1-1: 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 5.4.20.3.1-2.

Table 5.4.20.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	VnfLcmOperationOccurrenceNotification	1	A notification about lifecycle changes triggered by a VNF LCM operation occurrence.	
	VnfIdentifierCreationNotification	1	A notification about the creation of a VNF identifier and the related "Individual VNF instance" resource.	
	VnfIdentifierDeletionNotification	1	A notification about the deletion of a VNF identifier and the related "Individual VNF instance" resource.	
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 [6]	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 [6] may be returned.

5.4.20.3.2 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 tables 5.4.20.3.2-1 and 5.4.20.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.20.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.20.3.2-2: 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**
	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 [6]	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 [6] may be returned.

5.4.20.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.20.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.20.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.21 Resource: Create VNF snapshot task

5.4.21.1 Description

This task resource represents the "Create VNF Snapshot" operation. The API consumer can use this resource to request creating a VNF/VNFC snapshot from a VNF/VNFC instance.

It depends on the VNF capabilities, and is declared in the VNFD, whether this operation is supported for a particular VNF.

5.4.21.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/create_snapshot

This resource shall support the resource URI variables defined in table 5.4.21.2-1.

Table 5.4.21.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance from which a VNF snapshot is to be created. 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 VNF instance resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.21.3 Resource methods

5.4.21.3.1 POST

The POST method requests taking a snapshot a VNF instance and populating a previously created VNF snapshot resource (refer to clause 5.4.23.3.1) with the snapshot content.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 5.4.1.2.

In addition, once the VNFM has successfully completed the underlying VNF LCM operation occurrence, it shall reflect the result of the VNF snapshot creation by updating the corresponding "Individual VNF snapshot" resource indicated by the "vnfSnapshotInfoId" attribute of the "CreateVnfSnapshotRequest" that is included in the payload body of the request.

This method shall follow the provisions specified in tables 5.4.21.3.1-1 and 5.4.21.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.21.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.21.3.1-2: Details of the POST request/response on this resource

Request body	**Data type**	**Cardinality**	**Description**	
	CreateVnfSnapshotRequest	1	Parameters for the "Create VNF/VNFC Snapshot" operation, as defined in clause 5.5.2.21.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request was accepted for processing, but the processing has not been completed. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "VNF LCM operation occurrence" resource corresponding to the operation.
	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 [6], 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 VNF instance 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	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 VNF instance resource is in NOT_INSTANTIATED state, or that another lifecycle management operation is ongoing. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall 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 [6], 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 provided identifier of the target "Individual VNF snapshot" resource for the VNF snapshot is invalid. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

5.4.21.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.21.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.21.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.21.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.22 Resource: Revert to VNF snapshot task

5.4.22.1 Description

This task resource represents the "Revert-to VNF Snapshot" operation. The API consumer can use this resource to request reverting a VNF/VNFC instance to a VNF/VNFC snapshot.

During the revert-to VNF/VNFC snapshot process, the VNFM shall perform and record the changes on the VNF/VNFC components and related resources, via the corresponding AffectedVnfc, AffectedVirtualLink, and AffectedVirtualStorage as follows:

- A component instance whose identifier is the same in between the "to be reverted" VNF/VNFC instance and the snapshot information, its change shall be signalled as "MODIFIED".
- A component instance whose snapshot information is present in the VNF/VNFC snapshot, but such component is not present in the "to be reverted" VNF/VNFC instance, its change shall be signalled as "ADDED".

- A component instance which is present in the "to be reverted" VNF/VNFC instance, but whose snapshot information is not present in the VNF/VNFC snapshot, the component shall be terminated, and its change shall be signalled as "REMOVED".

During the "revert to VNF snapshot" process, for VNF constituents (e.g. VNFC, connection points, etc.) from the VNF snapshot that are added or modified in the "to be reverted" VNF instance, the VNFM shall assign the original identifier value present in the VNF snapshot in the case that the identifier value setting for such a VNF constituent is the responsibility of the VNFM. The identifier of the VNF instance shall not be modified in the reversion process.

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.

5.4.22.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/revert_to_snapshot

This resource shall support the resource URI variables defined in table 5.4.22.2-1.

Table 5.4.22.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance for the VNF snapshot to be reverted to. 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 VNF instance resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.22.3 Resource methods

5.4.22.3.1 POST

The POST method requests reverting a VNF/VNFC instance to a VNF/VNFC snapshot.

This method shall follow the provisions specified in tables 5.4.22.3.1-1 and 5.4.22.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.22.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.22.3.1-2: Details of the POST request/response on this resource

Request body	**Data type**	**Cardinality**	**Description**	
	RevertToVnfSnapshotRequest	1	Parameters for the Revert-to VNF/VNFC snapshot operation, as defined in clause 5.5.2.24.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request was accepted for processing, but the processing has not been completed. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "VNF LCM operation occurrence" resource corresponding to the operation.
	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 [6], 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 VNF instance 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	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 VNF instance resource is in NOT_INSTANTIATED state, or that another lifecycle management operation is ongoing. 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 [6]	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 [6] may be returned.

5.4.22.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.22.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.22.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.22.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.23 Resource: VNF snapshots

5.4.23.1 Description

This resource represents VNF snapshots. The API consumer can use this resource to create individual VNF snapshot resources and to query information of the VNF/VNFC snapshots.

5.4.23.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_snapshots

This resource shall support the resource URI variables defined in table 5.4.23.2-1.

Table 5.4.23.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 5.1a.

5.4.23.3 Resource methods

5.4.23.3.1 POST

The POST method creates a new individual VNF snapshot resource.

As a result of successfully executing this method, a new "Individual VNF snapshot" resource as defined in clause 5.4.24 shall have been created.

The creation of an "Individual VNF snapshot" resource can be performed for two reasons:

- To create an "Individual VNF snapshot" resources that can later be populated by a new VNF snapshot taken from a VNF instance (refer to clause 5.4.21.3.1).
- To create an "Individual VNF snapshot" resource that can be populated with information gathered from a VNF snapshot package extraction. In this case, the API consumer indicates the source of the VNF snapshot package in the payload body of the POST request to the present resource.

In the second case, for a successful execution of the operation, the values in the "VnfSnapshotInfo" data structure representing the "Individual VNF snapshot" resource shall be applied as follows:

- If the request (refer to clause 5.5.2.20) includes the "vnfSnapshotPkgId" attribute, the VNFM shall first fetch the VNF snapshot record from the source VNF snapshot package signalled by this identifier attribute in the request and then apply the "VnfSnapshotInfo" from the fetched VNF snapshot record.

This method shall follow the provisions specified in tables 5.4.23.3.1-1 and 5.4.23.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.23.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.23.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Description	
Request body	CreateVnfSnapshotInfoRequest	1	The VNF snapshot resource creation parameters, as defined in clause 5.5.2.20.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfSnapshotInfo	1	201 Created	Shall be returned when an individual VNF snapshot resource has been created successfully. The response body shall contain a representation of the new individual VNF snapshot resource, as defined in clause 5.5.2.22. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the "individual VNF snapshot" resource.
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

5.4.23.3.2 GET

The GET method queries information about multiple VNF/VNFC snapshots.

This method shall follow the provisions specified in tables 5.4.23.3.2-1 and 5.4.23.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.23.3.2-1: 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 [6]. The VNFM shall support receiving this parameter as part of the URI query string. The EM may supply this parameter. All attribute names that appear in the VnfSnapshot and in data types referenced from it shall be supported by the VNFM 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 [6] for details. The VNFM 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 [6] for details. The VNFM 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 [6] for details. The VNFM 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 [6] for details. The VNFM 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 - vnfcSnapshots
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource.

Table 5.4.23.3.2-2: 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	Shall be returned when information about zero or more VNF snapshots was queried successfully. The response body shall contain in an array the representations of zero or more individual VNF snapshot resources, as defined in clause 5.5.2.22. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] 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 [6].
	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 VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [6].
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

5.4.23.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.23.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.23.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.24 Resource: Individual VNF snapshot

5.4.24.1 Description

This resource represents an individual VNF snapshot. The API consumer can use this resource to read information about the VNF/VNFC snapshot, and to delete the VNF/VNFC snapshot.

5.4.24.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_snapshots/{vnfSnapshotInfoId}

The base resource URI variables for this resource are defined in table 5.4.24.2-1.

Table 5.4.24.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 5.1a.
vnfSnapshotInfoId	Identifier of the individual VNF snapshot 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 VNF snapshot resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.24.3 Resource methods

5.4.24.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.24.3.2 GET

The GET method retrieves information about a VNF /VNFC snapshot by reading an individual VNF snapshot resource.

This method shall follow the provisions specified in tables 5.4.24.3.2-1 and 5.4.24.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.24.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.24.3.2-2: 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 5.5.2.22.
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

5.4.24.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.24.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

5.4.24.3.5 DELETE

This method deletes an individual VNF snapshot resource and the associated VNF snapshot information managed by the VNFM, and any resource associated to the VNF/VNFC snapshot managed by the VIM.

As the result of successfully executing this method, the "Individual VNF snapshot" resource shall not exist any longer.

This method shall follow the provisions specified in tables 5.4.24.3.5-1 and 5.4.24.3.5-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.24.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.24.3.5-2: 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 resource and the associated VNF/VNFC snapshot were 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 VNF snapshot is in use by some operation such as reverting a VNF instance to a VNF snapshot or creating a VNF snapshot package. 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 [6]	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 [6] may be returned.

5.5 Data model

5.5.1 Introduction

This clause defines the request and response data structures of the VNF 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 Introduction

This clause defines the data structures to be used in resource representations and notifications.

5.5.2.2 Type: VnfInstance

This type represents a VNF instance. It shall comply with the provisions defined in table 5.5.2.2-1.

NOTE: Clause B.3.2 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 5.5.2.2-1: 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. This attribute can be modified with the PATCH method.
vnfInstanceDescription	String	0..1	Human-readable description of the VNF instance. This attribute can be modified with the PATCH method.
vnfdId	Identifier	1	Identifier of the VNFD on which the VNF instance is based. 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.

Attribute name	Data type	Cardinality	Description
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. The declaration of configurable properties in the VNFD can optionally contain the specification of initial values. See note 2, note 3 and note 4. The VNFM shall reject requests to write configurable properties that are not declared in the VNFD with a "422 Unprocessable entity" error response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6]. 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: • 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. • 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. These configurable properties can be initialized with default values from the VNFD (see note 4). Configurable properties can be modified with values passed in the request structures of certain LCM operations, such as the InstantiateVnfRequest structure. Further, these configurable properties can be created, modified or deleted with the PATCH method. In addition, the provisions in clause 5.7 shall apply.
instantiationState	Enum (inlined)	1	The instantiation state of the VNF. Permitted values: • NOT_INSTANTIATED: The VNF instance is terminated or not instantiated. • INSTANTIATED: The VNF instance is instantiated.
instantiatedVnfInfo	Structure (inlined)	0..1	Information specific to an instantiated VNF instance. This attribute shall be present if the instantiateState attribute value is INSTANTIATED.
>flavourId	IdentifierInVnfd	1	Identifier of the VNF deployment flavour applied to this VNF instance.
>vnfState	VnfOperationalStateType	1	State of the VNF instance.

Attribute name	Data type	Cardinality	Description
>scaleStatus	ScaleInfo	0..N	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. See clause B.2 for an explanation of VNF scaling.
>maxScaleLevels	ScaleInfo	0..N	Maximum allowed scale levels of the VNF, one entry per aspect. This attribute shall be present if the VNF supports scaling.
>extCpInfo	VnfExtCpInfo	1..N	Information about the external CPs exposed by the VNF instance. 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.
>vipCpInfo	VipCpInfo	0..N	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. 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 notes 5 and 6.
>monitoringParameters	MonitoringParameter	0..N	Active monitoring parameters.
>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 [2].
>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 6.
>virtualStorageResourceInfo	VirtualStorageResourceInfo	0..N	Information about the virtualised storage resources used as storage for the VNF instance.
>vnfcInfo	VnfcInfo	0..N	Information about the VNFC instances.

Attribute name	Data type	Cardinality	Description
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. The declaration of metadata in the VNFD can optionally contain the specification of initial values. See note 2 and note 4. The VNFM shall accept requests to write metadata that are not declared in the VNFD. These attributes can be initialized with default values from the VNFD (see note 4) or with values passed in the CreateVnfRequest structure (see clause 5.4.2.3.1). These attributes can be created, modified or removed with the PATCH method.
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. All extensions that are allowed for the VNF are declared in the VNFD. The declaration of an extension in the VNFD contains information on whether its presence is optional or required, and optionally can specify an initial value. See note 2 and note 4. The VNFM shall reject requests to write extension attributes that are not declared in the VNFD with a "422 Unprocessable entity" error response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6]. 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. These attributes can be initialized with default values from the VNFD (see note 4). These attributes can be modified with values passed in the request structures of certain LCM operations, such as the InstantiateVnfRequest structure. Further, these attributes can be created, modified or deleted with the PATCH method. In addition, the provisions in clause 5.7 shall apply.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>indicators	Link	0..1	Indicators related to this VNF instance, if applicable.

Attribute name	Data type	Cardinality	Description
>instantiate	Link	0..1	Link to the "Instantiate VNF task" resource, if the related operation is possible based on the current status of this VNF instance resource (i.e. VNF instance in NOT_INSTANTIATED state).
>terminate	Link	0..1	Link to the "Terminate VNF task" resource, if the related operation is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>scale	Link	0..1	Link to the "Scale VNF task" resource, if the related operation is supported for this VNF instance, and is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>scaleToLevel	Link	0..1	Link to the "Scale VNF_to_level task" resource, if the related operation is supported for this VNF instance, and is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>changeFlavour	Link	0..1	Link to the "Change VNF_flavour task" resource, if the related operation is supported for this VNF instance, and is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>heal	Link	0..1	Link to the "Heal VNF task" resource, if the related operation is supported for this VNF instance, and is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>operate	Link	0..1	Link to the "Operate VNF task" resource, if the related operation is supported for this VNF instance, and is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>changeExtConn	Link	0..1	Link to the "Change external VNF connectivity task" resource, if the related operation is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>createSnapshot	Link	0..1	Link to the "Create VNF snapshot task" resource, if the related operation is supported for this VNF instance and is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>revertToSnapshot	Link	0..1	Link to the "Revert to VNF snapshot task" resource, if the related operation is supported for this VNF instance and is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>changeCurrentVnfPkg	Link	0..1	Link to the "Change current VNF package task" resource, if the related operation is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).

Attribute name	Data type	Cardinality	Description
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. NOTE 2: ETSI GS NFV-SOL 001 [i.3] specifies the structure and format of the VNFD based on TOSCA specifications. 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. NOTE 4: Upon creation of the VnfInstance structure, the VNFM shall create and initialize all child attributes of "vnfConfigurableProperties", "metadata" and "extensions" that were declared in the VNFD with a defined initial value. The defined initial values can be declared in the VNFD, and/or, in case of "metadata", obtained from the "CreateVnfRequest" structure. Child attributes of "vnfConfigurableProperties", "metadata" and "extensions" that have no defined initial value shall not be created, in order to be consistent with the semantics of the JSON Merge Patch method (see IETF RFC 7396 [3]) that interprets null values as deletion request. NOTE 5: 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 5.5.3.5). NOTE 6: 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.			

5.5.2.3 Type: CreateVnfRequest

This type represents request parameters for the "Create VNF identifier" operation. It shall comply with the provisions defined in table 5.5.2.3-1.

Table 5.5.2.3-1: Definition of the CreateVnfRequest data type

Attribute name	Data type	Cardinality	Description
vnfdId	Identifier	1	Identifier that identifies the VNFD which defines the VNF instance to be created.
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.
metadata	KeyValuePairs	0..1	If present, this attribute provides additional initial values, overriding those obtained from the VNFD, for the "metadata" attribute in "VnfInstance", as defined in clause 5.5.2.2. Provisions for handling metadata during the operation are defined in clause 5.4.2.3.1.

5.5.2.4 Type: InstantiateVnfRequest

This type represents request parameters for the "Instantiate VNF" operation It shall comply with the provisions defined in table 5.5.2.4-1.

Table 5.5.2.4-1: Definition of the InstantiateVnfRequest data type

Attribute name	Data type	Cardinality	Description
flavourId	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. If not present, the default instantiation level as declared in the VNFD is instantiated.
extVirtualLinks	ExtVirtualLinkData	0..N	Information about external VLs to connect the VNF to, including configuration information for the CPs via which the VNF instance can attach to this VL.

Attribute name	Data type	Cardinality	Description
			The following applies to the "ExtVirtualLinkData" information provided in this request: Even if the VNF is not instantiated in fully scaled-out state, the API consumer shall provide enough CP configuration records to allow connecting the VNF instance, fully scaled out in all scaling aspects, to the external VLs.
extManagedVirtualLinks	ExtManagedVirtual LinkData	0..N	Information about internal VLs that are managed by other entities than the VNFM. See note.
localizationLanguage	String	0..1	Localization language of the VNF to be instantiated. The value shall comply with the format defined in IETF RFC 5646 [2].
extensions	KeyValuePairs	0..1	If present, this attribute provides modifications to the default values, as obtained from the VNFD, of the "extensions" attribute in "VnfInstance", as defined in clause 5.5.2.2. Provisions for handling extensions during the operation are defined in clause 5.4.4.3.1.
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".
vnfConfigurableProperties	KeyValuePairs	0..1	If present, this attribute provides modifications to the default values, as obtained from the VNFD, of the "vnfConfigurableProperties" attribute in "VnfInstance", as defined in clause 5.5.2.2. Provisions for handling configurable properties during the operation are defined in clause 5.4.4.3.1.
NOTE: 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.			

5.5.2.5 Type: ScaleVnfRequest

This type represents request parameters for the "Scale VNF" operation. It shall comply with the provisions defined in table 5.5.2.5-1. See clause B.2 for an explanation of VNF scaling.

Table 5.5.2.5-1: Definition of the ScaleVnfRequest data type

Attribute name	Data type	Cardinality	Description
type	Enum (inlined)	1	Indicates the type of the scale operation requested. Permitted values: • SCALE_OUT: adding additional VNFC instances to the VNF to increase capacity • SCALE_IN: removing VNFC instances from the VNF in order to release unused capacity.
aspectId	IdentifierInVnfd	1	Identifier of the scaling aspect.
numberOfSteps	Integer	0..1	Number of scaling steps to be executed as part of this Scale VNF operation. It shall be a positive number and the default value shall be 1.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the EM as input to the scaling process, specific to the VNF being scaled, as declared in the VNFD as part of "ScaleVnfOpConfig".

5.5.2.6 Type: ScaleVnfToLevelRequest

This type represents request parameters for the "Scale VNF to Level" operation. It shall comply with the provisions defined in table 5.5.2.6-1. See clause B.2 for an explanation of VNF scaling.

Table 5.5.2.6-1: Definition of the ScaleVnfToLevelRequest data type

Attribute name	Data type	Cardinality	Description
instantiationLevelId	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.
scaleInfo	ScaleInfo	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 EM as input to the scaling process, specific to the VNF being scaled, as declared in the VNFD as part of "ScaleVnfToLevelOpConfig".
NOTE: Either the instantiationLevelId attribute or the scaleInfo attribute shall be included.			

5.5.2.7 Type: ChangeVnfFlavourRequest

This type represents request parameters for the "Change VNF flavour" operation. It shall comply with the provisions defined in table 5.5.2.7-1.

Table 5.5.2.7-1: Definition of the ChangeVnfFlavourRequest data type

Attribute name	Data type	Cardinality	Description
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. If not present, the default instantiation level as declared in the VNFD is instantiated.
extVirtualLinks	ExtVirtualLinkData	0..N	Information about external VLs to connect the VNF to, including configuration information for the CPs via which the VNF instance can attach to this VL. Entries in the list of external VLs that are unchanged need not be supplied as part of this request. The following applies to the "ExtVirtualLinkData" information provided in this request, together with the related "ExtVirtualLinkInfo" information known to the VNFM represented in the "VnfInstance" structure (see clause 5.5.2.2): Even if the VNF is not in fully scaled-out state after changing the flavour, the API consumer shall provide enough CP configuration records to allow connecting the VNF instance, fully scaled out in all scaling aspects, to the external VLs.
extManagedVirtualLinks	ExtManagedVirtualLinkData	0..N	Information about internal VLs that are managed by other entities than the VNFM. See note.
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".
extensions	KeyValuePairs	0..1	If present, this attribute provides modifications to the values of the "extensions" attribute in "VnfInstance", as defined in clause 5.5.2.2. Provisions for handling extensions during the operation, are defined in clause 5.4.7.3.1.
vnfConfigurableProperties	KeyValuePairs	0..1	If present, this attribute provides modifications to the values of the "vnfConfigurableProperties" attribute in "VnfInstance", as defined in clause 5.5.2.2. Provisions for handling VNF configurable properties during the operation, are defined in clause 5.4.7.3.1.
NOTE: 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.			

5.5.2.8 Type: TerminateVnfRequest

This type represents request parameters for the "Terminate VNF" operation. It shall comply with the provisions defined in table 5.5.2.8-1.

Table 5.5.2.8-1: Definition of the TerminateVnfRequest data type

Attribute name	Data type	Cardinality	Description
terminationType	Enum (inlined)	1	Indicates whether forceful or graceful termination is requested. See note. Permitted values: • FORCEFUL: The VNFM will shut down the VNF and release the resources immediately after accepting the request. • GRACEFUL: The VNFM will first arrange to take the VNF out of service after accepting the request. Once the operation of taking the VNF out of service finishes (irrespective of whether it has succeeded or failed) or once the timer value specified in the "gracefulTerminationTimeout" attribute expires, the VNFM will shut down the VNF and release the resources.
gracefulTermination Timeout	Integer	0..1	This 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. If not given and the "terminationType" attribute is set to "GRACEFUL", it is expected that the VNFM waits for the successful taking out of service of the VNF, no matter how long it takes, before shutting down the VNF and releasing the resources.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the EM as input to the termination process, specific to the VNF being terminated, as declared in the VNFD as part of "TerminateVnfOpConfig".
NOTE: In case of forceful termination, the VNF instance is terminated immediately. If the VNF is still in service, this can adversely impact the network service, and therefore, the EM needs to determine if forceful termination is applicable in the particular situation.			

5.5.2.9 Type: HealVnfRequest

This type represents request parameters for the "Heal VNF" operation. It shall comply with the provisions defined in table 5.5.2.9-1.

Table 5.5.2.9-1: Definition of the HealVnfRequest data type

Attribute name	Data type	Cardinality	Description
vnfcInstanceId	Identifier	0..N	List of identifiers of VNFC instances for which a healing action is requested. Each identifier references the "id" attribute in a "VnfcInfo" structure. Cardinality can be "0" to denote that the request applies to the whole VNF and not a specific VNFC instance.
cause	String	0..1	Indicates the reason why a healing procedure is required.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the EM as input to the healing process, specific to the VNF being healed as declared in the VNFD as part of "HealVnfOpConfig".
healScript	String	0..1	Provides link to a script that should be executed as part of the healing action or a set of rules for healing procedure.

5.5.2.10 Type: OperateVnfRequest

This type represents request parameters for the "Operate VNF" operation. It shall comply with the provisions defined in table 5.5.2.10-1.

Table 5.5.2.10-1: Definition of the OperateVnfRequest data type

Attribute name	Data type	Cardinality	Description
vnfcInstanceId	Identifier	0..N	List of identifiers of VNFC instances. Each identifier references the "id" attribute in a "VnfcInfo" structure. Cardinality can be "0" to denote that the request applies to the whole VNF and not a specific VNFC instance.
changeStateTo	VnfOperationalStateType	1	The desired operational state (i.e. started or stopped) to change the VNF/VNFC 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.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the EM as input to the process, specific to the VNF of which the operation status is changed, as declared in the VNFD as part of "OperateVnfOpConfig".
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" 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 has been set to "FORCEFUL", when the "changeStateTo" attribute is equal to "STOPPED" and the "stopType" attribute is absent.			

5.5.2.11 Type: ChangeExtVnfConnectivityRequest

This type represents request parameters for the "Change external VNF connectivity" operation to modify the external connectivity of a VNF instance. It shall comply with the provisions defined in table 5.5.2.11-1.

Table 5.5.2.11-1: Definition of the ChangeExtVnfConnectivityRequest data type

Attribute name	Data type	Cardinality	Description
extVirtualLinks	ExtVirtualLinkData	1..N	Information about external VLs to change (e.g. connect the VNF to) including configuration information for the CPs via which the VNF instance can attach to this VL. Entries in the list of external VLs that are unchanged need not be supplied as part of this request. The following applies to the "ExtVirtualLinkData" information provided in this request, together with the related "ExtVirtualLinkInfo" information known to the VNFM represented in the "VnfInstance" structure (see clause 5.5.2.2): Even if the VNF is not in fully scaled-out state, the API consumer shall provide enough CP configuration records to allow connecting the VNF instance, fully scaled out in all scaling aspects, to the external VLs.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the EM as input to the process, specific to the VNF of which the external connectivity is changed, as declared in the VNFD as part of "ChangeExtVnfConnectivityOpConfig".

The following behaviour applies for the changes that can be performed with this operation:

- To change the connection of external CP instances based on certain external CPDs from a "source" external VL to a different "target" external VL, the identifier of the "target" external VL shall be sent in the "extVirtualLinkId" attribute of the "extVirtualLinks" parameter, and the "extCps" attributes of that parameter shall refer via the "cpdId" attribute to the external CPDs of the corresponding external connection point instances that are to be reconnected to the target external VL.

NOTE: For CP instances that are not part of a trunk this means that all CP instances based on a given external CPD will be reconnected. See clause B.3.3 for an illustration. Likewise, for CP instances that are part of a trunk and have the same segmentationId, all CP instances (subports) based on a given external CPD will be connected, disconnected or reconnected.

- To change the connectivity parameters of the external CPs connected to a particular external VL, including changing addresses, the identifier of that external VL shall be sent in the "extVirtualLinkId" attribute of the "extVirtualLinks" parameter, and the "extCps" attribute of that parameter shall contain at least those entries with modified parameters.

5.5.2.11a Type: ChangeCurrentVnfPkgRequest

This type represents request parameters for the "Change current VNF package" operation to replace the VNF package on which a VNF instance is based. It shall comply with the provisions defined in table 5.5.2.11a-1.

Table 5.5.2.11a-1: Definition of the ChangeCurrentVnfPkgRequest data type

Attribute name	Data type	Cardinality	Description
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, including configuration information for the CPs via which the VNF instance can attach to this VL. Entries in the list that are unchanged need not be supplied as part of this request. The following applies to the "ExtVirtualLinkData" information provided in this request, together with the related "ExtVirtualLinkInfo" information known to the VNFM represented in the "VnfInstance" structure (see clause 5.5.2.2): Even if the VNF is not in fully scaled-out state after the change of the VNF package, the API consumer shall provide enough CP configuration records to allow connecting the VNF instance, fully scaled out in all scaling aspects, to the external VLs.
extManagedVirtualLinks	ExtManagedVirtualLinkData	0..N	Information about internal VLs that are managed by other entities than the VNFM. See note.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the EM as input to the process, specific to the VNF of which the underlying VNF package is changed, as declared in the VNFD as part of "ChangeCurrentVnfPkgOpConfig".
extensions	KeyValuePairs	0..1	If present, this attribute provides modifications to the values of the "extensions" attribute in "VnfInstance", as defined in clause 5.5.2.2. 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.
vnfConfigurableProperties	KeyValuePairs	0..1	If present, this attribute provides modifications to the values of the "vnfConfigurableProperties" attribute in "VnfInstance", as defined in clause 5.5.2.2. 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.
NOTE: 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.			

5.5.2.12 Type: VnfInfoModificationRequest

This type represents attribute modifications for an "Individual VNF instance" resource, i.e. modifications to a resource representation based on the "VnfInstance" data type. The attributes of "VnfInstance" that can be modified according to the provisions in clause 5.5.2.2 are included in the "VnfInfoModificationRequest" data type.

The "VnfInfoModificationRequest" data type shall comply with the provisions defined in table 5.5.2.12-1.

Table 5.5.2.12-1: Definition of the VnfInfoModificationRequest data type

Attribute name	Data type	Cardinality	Description
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.
vnfdId	Identifier	0..1	New value of the "vnfdId" attribute in "VnfInstance". The value "null" is not permitted.
vnfConfigurableProperties	KeyValuePairs	0..1	Modifications of the "vnfConfigurableProperties" attribute in "VnfInstance". If present, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [3]).
metadata	KeyValuePairs	0..1	Modifications of the "metadata" attribute in "VnfInstance". If present, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [3]).
extensions	KeyValuePairs	0..1	Modifications of the "extensions" attribute in "VnfInstance". If present, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [3]).
vnfcInfoModifications	VnfcInfoModifications	0..N	Modifications of certain entries in the "vnfcInfo" attribute array in the "instantiatedVnfInfo" attribute of "VnfInstance" to be used as "newList" as defined below this table.

The following provisions shall apply when modifying an attribute that is an array of objects of type "VnfcInfo" by supplying an array of objects of type "VnfcInfoModifications".

Assumptions:

1) "oldList" is the "VnfcInfo" array to be modified and "newList" is the "VnfcInfoModifications" array that contains the changes.

2) "oldEntry" is an entry in "oldList" and "newEntry" is an entry in "newList".

3) A "newEntry" has a "corresponding entry" if there exists an "oldEntry" that has the same content of the "id" attribute as the "newEntry"; a "newEntry" has no corresponding entry if no such "oldEntry" exists.

4) In any array of "VnfcInfo" resp. "VnfcInfoModifications" structures, 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 updated with the content of "newEntry" as specified for the data type of "newEntry (refer to clause 5.5.3.24 for the data type "VnfcInfoModifications").

5.5.2.12a Type: VnfInfoModifications

This type represents attribute modifications that were performed on an "Individual VNF instance" resource. The attributes that can be included consist of those requested to be modified explicitly in the "VnfInfoModificationRequest" data structure, and additional attributes of the "VnfInstance" data structure that were modified implicitly e.g. when modifying the referenced VNF package.

The "VnfInfoModifications" data type shall comply with the provisions defined in table 5.5.2.12a-1.

Table 5.5.2.12a-1: Definition of the VnfInfoModifications data type

Attribute name	Data type	Cardinality	Description
vnfInstanceName	String	0..1	If present, this attribute signals modifications of the "vnfInstanceName" attribute in "VnfInstance" as defined in clause 5.5.2.12.
vnfInstanceDescription	String	0..1	If present, this attribute signals modifications of the "vnfInstanceDescription" attribute in "VnfInstance", as defined in clause 5.5.2.12.
vnfConfigurableProperties	KeyValuePairs	0..1	If present, this attribute signals modifications of the "vnfConfigurableProperties" attribute in "VnfInstance", as defined in clause 5.5.2.12. In addition, the provisions in clause 5.7 shall apply.
metadata	KeyValuePairs	0..1	If present, this attribute signals modifications of the "metadata" attribute in "VnfInstance", as defined in clause 5.5.2.12.
extensions	KeyValuePairs	0..1	If present, this attribute signals modifications of the "extensions" attribute in "VnfInstance", as defined in clause 5.5.2.12. In addition, the provisions in clause 5.7 shall apply.
vnfdId	Identifier	0..1	If present, this attribute signals modifications of the "vnfdId" attribute in "VnfInstance", as defined in clause 5.5.2.12.
vnfProvider	String	0..1	If present, this attribute signals modifications of the "vnfProvider" attribute in "VnfInstance". See note.
vnfProductName	String	0..1	If present, this attribute signals modifications of the "vnfProductName" attribute in "VnfInstance". See note.
vnfSoftwareVersion	Version	0..1	If present, this attribute signals modifications of the "vnfSoftwareVersion" attribute in "VnfInstance". See note.
vnfdVersion	Version	0..1	If present, this attribute signals modifications of the "vnfdVersion" attribute in "VnfInstance". See note.
vnfcInfoModifications	VnfcInfoModifications	0..N	If present, this attribute signals modifications of certain entries in the "vnfcInfo" attribute array in the "instantiatedVnfInfo" attribute of "VnfInstance", as defined in clause 5.5.2.12.
NOTE: If present, this attribute (which depends on the value of the "vnfdId" attribute) was modified implicitly following a request to modify the "vnfdId" attribute, by copying the value of this attribute from the VNFD in the VNF Package identified by the "vnfdId" attribute.			

5.5.2.13 Type: VnfLcmOpOcc

This type represents a VNF lifecycle management operation occurrence. It shall comply with the provisions defined in table 5.5.2.13-1.

Table 5.5.2.13-1: Definition of the VnfLcmOpOcc data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this VNF lifecycle management operation occurrence.
operationState	LcmOperationStateType	1	The state of the LCM operation.
stateEnteredTime	DateTime	1	Date-time when the current state has been entered.
startTime	DateTime	1	Date-time of the start of the operation.
vnfInstanceId	Identifier	1	Identifier of the VNF instance to which the operation applies.
grantId	Identifier	0..1	Identifier of the grant related to this VNF LCM operation occurrence. Shall be set to the value of the "id" attribute in the "Grant" representing the associated "Individual Grant", if such grant exists.
operation	LcmOperationType	1	Type of the actual LCM operation represented by this VNF LCM operation occurrence.

Attribute name	Data type	Cardinality	Description
isAutomaticInvocation	Boolean	1	Set to true if this VNF LCM operation occurrence has been triggered by an automated procedure inside the VNFM (i.e. ScaleVnf/ScaleVnfToLevel triggered by auto-scale, or HealVnf triggered by auto-heal). Set to false otherwise.
operationParams	Object	0..1	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 5.7 shall apply. The following mapping between operationType and the data type of this attribute shall apply: • INSTANTIATE: InstantiateVnfRequest • SCALE: ScaleVnfRequest • SCALE_TO_LEVEL: ScaleVnfToLevelRequest • CHANGE_FLAVOUR: • ChangeVnfFlavourRequest • OPERATE: OperateVnfRequest • HEAL: HealVnfRequest • CHANGE_EXT_CONN: • ChangeExtVnfConnectivityRequest • TERMINATE: TerminateVnfRequest • MODIFY_INFO: VnfInfoModificationRequest • CREATE_SNAPSHOT: • CreateVnfSnapshotRequest • REVERT_TO_SNAPSHOT: • RevertToVnfSnapshotRequest • CHANGE_VNFPKG: • ChangeCurrentVnfPkgRequest 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.
isCancelPending	Boolean	1	If the VNF LCM operation occurrence is in "STARTING", "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.
>affectedVnfcs	AffectedVnfc	0..N	Information about VNFC instances that were affected during the lifecycle operation. See note 1.
>affectedVirtualLinks	AffectedVirtualLink	0..N	Information about VL instances that were affected during the lifecycle operation. See note 1 and note 3.
>affectedExtLinkPorts	AffectedExtLinkPort	0..N	Information about external VNF link ports that were affected during the lifecycle operation. See note 1.
>affectedVirtualStorages	AffectedVirtualStorage	0..N	Information about virtualised storage instances that were affected during the lifecycle operation. See note 1.
changedInfo	VnfInfoModifications	0..1	Information about the changed VNF instance information, including VNF configurable properties, if applicable. See note 1 and note 2.
affectedVipCps	AffectedVipCp	0..N	Information about virtual IP CP instances that were affected during the execution of the lifecycle management operation.

Attribute name	Data type	Cardinality	Description
changedExtConnectivity	ExtVirtualLinkInfo	0..N	Information about changed external connectivity, if applicable. See note 1.
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 1 and note 2.
vnfSnapshotInfoId	Identifier	0..1	Identifier of the "individual VNF snapshot" resource. Shall be present if applicable to the type of LCM operation, i.e. if the value of the "operation" attribute is either "CREATE_SNAPSHOT" or "REVERT_TO_SNAPSHOT".
lcmCoordinations	Structure (inlined)	0..N	Information about LCM coordination actions (see clause 10) related to this LCM operation occurrence
>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 10.5.2.3). 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 10.4.2.3.1).
>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 VNFM 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 4).
>startTime	DateTime	1	The time when the VNFM has received the confirmation that the coordination action has been started.
>endTime	DateTime	0..1	The time when the VNFM 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 4) 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. EM) • VNF: coordination with the VNF instance
>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 10) that were rejected by 503 error which means they will be tried again after a delay. See note 5.
>coordinationActionName	Identifier	1	Indicator of the actual coordination action.
>rejectionTime	DateTime	1	The time when the VNFM 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. EM) • VNF: coordination with the VNF instance
>delay	DateTime	1	The end of the delay period, as calculated from the startTime and "Retry-After" header.

Attribute name	Data type	Cardinality	Description
warnings	String	0..N	Warning messages that were generated while the operation was executing. If the operation has included 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.
>vnfInstance	Link	1	Link to the VNF instance that the operation applies to.
>grant	Link	0..1	Link to the grant for this operation, if one exists.
>cancel	Link	0..1	Link to the task resource that represents the "cancel" operation for this VNF LCM operation occurrence, if cancelling is currently allowed.
>retry	Link	0..1	Link to the task resource that represents the "retry" operation for this VNF LCM operation occurrence, if retrying is currently allowed.
>rollback	Link	0..1	Link to the task resource that represents the "rollback" operation for this VNF LCM operation occurrence, if rolling back is currently allowed.
>fail	Link	0..1	Link to the task resource that represents the "fail" operation for this VNF LCM operation occurrence, if declaring as failed is currently allowed.
>vnfSnapshot	Link	0..1	Link to the VNF snapshot resource, if the VNF LCM operation occurrence is related to a VNF snapshot. Shall be present if operation="CREATE_SNAPSHOT" or operation="REVERT_TO_SNAPSHOT".

NOTE 1: This allows the API consumer to obtain the information contained in the latest "result" notification if it has not received it due to an error or a wrongly configured subscription filter.
NOTE 2: Not more than one of changedInfo and modificationsTriggeredByVnfPkgChange shall be present.
NOTE 3: For a particular affected VL, there shall be as many "AffectedVirtualLink" entries as needed for signalling the different types of changes, i.e. one per virtual link and change type. For instance, in the case of signaling affected VL instances involving the addition of a particular VL instance with links ports, one "AffectedVirtualLink" entry signals the addition of the VL by using the "changeType" attribute of "AffectedVirtualLink" structure equal to "ADDED", and another "AffectedVirtualLink" entry signals the addition of VNF link ports of the VL by using the "changeType" equal to "LINK_PORT_ADDED".
NOTE 4: A coordination action has timed out if the VNFM 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 5: 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" and "ROLLED_BACK".

5.5.2.14 Type: CancelMode

This type represents a parameter to select the mode of cancelling an ongoing VNF LCM operation occurrence. It shall comply with the provisions defined in table 5.5.2.14-1.

Table 5.5.2.14-1: Definition of the CancelMode data type

Attribute name	Data type	Cardinality	Description
cancelMode	CancelModeType	1	Cancellation mode to apply.

5.5.2.15 Type: LccnSubscriptionRequest

This type represents a subscription request related to notifications about VNF lifecycle changes. It shall comply with the provisions defined in table 5.5.2.15-1.

Table 5.5.2.15-1: 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 8.4 of ETSI GS NFV-SOL 013 [6]. 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".

5.5.2.16 Type: LccnSubscription

This type represents a subscription related to notifications about VNF lifecycle changes. It shall comply with the provisions defined in table 5.5.2.16-1.

Table 5.5.2.16-1: 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.

5.5.2.17 Type: VnfLcmOperationOccurrenceNotification

This type represents a VNF lifecycle management operation occurrence notification, which informs the receiver of changes in the VNF lifecycle caused by a VNF LCM operation occurrence. It shall comply with the provisions defined in table 5.5.2.17-1. The support of the notification is mandatory.

This notification shall be triggered by the VNFM when there is a change in the state of a VNF LCM operation occurrence that changes the VNF lifecycle, which represents an occurrence of one the following LCM operations:

- Instantiation of the VNF
- Scaling of the VNF instance (including auto-scaling)
- Healing of the VNF instance (including auto-healing)
- Change of the state of the VNF instance (i.e. Operate VNF)
- Change of the deployment flavour of the VNF instance
- Change of the external connectivity of the VNF instance
- Termination of the VNF instance
- Modification of VNF instance information and/or VNF configurable properties through the "PATCH" method on the "Individual VNF instance" resource

- Creation of a VNF snapshot
- Reversion of the VNF instance to a VNF snapshot
- Change of the current VNF package

Clause 5.6.2 defines the states and state transition of a VNF 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 a VNF LCM operation occurrence, it is assumed that the notification is sent by the VNFM before any action (including sending the grant request) 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 API consumer, and the API consumer shall be able to handle such a situation.

If this is a notification about a final or intermediate result state of a VNF 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 VNF 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 VNFM when a particular subscription matches can be controlled by the API consumer using the "verbosity" attribute in the subscription request (see clause 5.5.2.15). 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 "VnfLcmOpOcc" structure (see clause 5.5.2.13) which represents the "Individual LCM operation occurrence" resource associated with each of the notifications.

See clause 5.6.2.2 for further provisions regarding sending this notification, including in cases of handling LCM operation errors.

Table 5.5.2.17-1: Definition of the VnfLcmOperationOccurrenceNotification 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 "VnfLcmOperationOccurrenceNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to. Shall be set to the value of the "id" attribute of the "LccnSubscription" representing the associated "Individual subscription" resource.
timeStamp	DateTime	1	Date-time of the generation of the notification.
notificationStatus	Enum (inlined)	1	Indicates whether this notification reports about the start of a lifecycle operation or the result of a lifecycle operation. Permitted values: • START: Informs about the start of the VNF LCM operation occurrence. • RESULT: Informs about the final or intermediate result of the VNF LCM operation occurrence.
operationState	LcmOperationStateType	1	The state of the VNF LCM operation occurrence.
vnfInstanceId	Identifier	1	The identifier of the VNF instance affected.
operation	LcmOperationType	1	The lifecycle management operation.
isAutomaticInvocation	Boolean	1	Set to true if this VNF LCM operation occurrence has been triggered by an automated procedure inside the VNFM (i.e. ScaleVnf/ScaleVnfToLevel triggered by auto-scale, or HealVnf triggered by auto-heal). Set to false otherwise.

Attribute name	Data type	Cardinality	Description
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 "vnfLcmOpOcc" child attribute of the "_links" attribute.
vnfLcmOpOccId	Identifier	1	The identifier of the VNF lifecycle management operation occurrence associated to the notification. Shall be set to the value of the "id" attribute of the "VnfLcmOpOcc" representing the associated "Individual VNF lifecycle management operation occurrence" resource.
affectedVnfcs	AffectedVnfc	0..N	Information about VNFC instances that were affected during the lifecycle operation. See note 1.
affectedVirtualLinks	AffectedVirtualLink	0..N	Information about VL instances that were affected during the lifecycle operation. See note 1 and note 2.
affectedExtLinkPorts	AffectedExtLinkPort	0..N	Information about external VNF link ports that were affected during the lifecycle operation. See note 1.
affectedVirtualStorages	AffectedVirtualStorage	0..N	Information about virtualised storage instances that were affected during the lifecycle operation. See note 1.
changedInfo	VnfInfoModifications	0..1	Information about the changed VNF instance information, including changed VNF configurable properties. Shall be present if the "notificationStatus" is set to "RESULT", the "operation" attribute is not equal to "CHANGE_VNFPKG", the "verbosity" attribute is set to "FULL" and the operation has performed any changes to VNF instance information, including VNF configurable properties. Shall be absent otherwise. See note 3.
affectedVipCps	AffectedVipCp	0..N	Information about virtual IP CP instances that were affected during the execution of the lifecycle management operation, if this notification represents the result of a lifecycle management operation occurrence. Shall be present if the "notificationStatus" is set to "RESULT", the "verbosity" attribute is set to "FULL" and the operation has made any changes to the VIP CP instances of the VNF instance. Shall be absent otherwise. Only information about VIP CP instances that have been added, deleted or modified shall be provided.
changedExtConnectivity	ExtVirtualLinkInfo	0..N	Information about changed external connectivity, if this notification represents the result of a lifecycle operation occurrence. Shall be present if the "notificationStatus" is set to "RESULT", the "verbosity" attribute is set to "FULL" and the operation has made any changes to the external connectivity of the VNF instance. Shall be absent otherwise. Only information about external VL instances that have been added or modified shall be provided.
modificationsTriggeredByVnfPkgChange	ModificationsTriggeredByVnfPkgChange	0..1	Information about performed changes of "VnfInstance" attributes triggered by changing the current VNF package. Shall be present if the "notificationStatus" is set to "RESULT", the "operation" attribute is equal to "CHANGE_VNFPKG", the "verbosity" attribute is set to "FULL" and the operation has performed any changes to "VnfInstance" attributes. Shall be absent otherwise. See note 3.

Attribute name	Data type	Cardinality	Description
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 [6]). Shall be present if the "operationState" attribute is "FAILED_TEMP", "FAILED" or "ROLLED_BACK" and shall be absent otherwise.
_links	LccnLinks	1	Links to resources related to this notification. The link URIs in this structure shall be set to point to the resources identified by the corresponding identifier attributes in 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 VNF LCM operation occurrence and by any of the error handling procedures for that operation occurrence. NOTE 2: For a particular affected VL, there shall be as many "AffectedVirtualLink" entries as needed for signalling the different types of changes, i.e. one per virtual link and change type. For instance, in the case of signaling affected VL instances involving the addition of a particular VL instance with links ports, one "AffectedVirtualLink" entry signals the addition of the VL by using the "changeType" attribute of "AffectedVirtualLink" structure equal to "ADDED", and another "AffectedVirtualLink" entry signals the addition of VNF link ports of the VL by using the "changeType" equal to "LINK_PORT_ADDED". NOTE 3: Not more than one of changedInfo and modificationsTriggeredByVnfPkgChange shall be present.			

5.5.2.18 Type: VnfIdentifierCreationNotification

This type represents a VNF identifier creation notification, which informs the receiver of the creation of a new "Individual VNF instance" resource and the associated VNF instance identifier. It shall comply with the provisions defined in table 5.5.2.18-1. The support of the notification is mandatory.

This notification shall be triggered by the VNFM when it has created an "Individual VNF instance" resource and the associated VNF instance identifier.

Table 5.5.2.18-1: Definition of the VnfIdentifierCreationNotification 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 "VnfIdentifierCreationNotification" 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.
vnfInstanceId	Identifier	1	The created VNF instance identifier.
_links	LccnLinks	1	Links to resources related to this notification.

5.5.2.19 Type: VnfIdentifierDeletionNotification

This type represents a VNF identifier deletion notification, which informs the receiver of the deletion of a new "Individual VNF instance" resource and the associated VNF instance identifier. It shall comply with the provisions defined in table 5.5.2.19-1. The support of the notification is mandatory.

This notification shall be triggered by the VNFM when it has deleted an "Individual VNF instance" resource and the associated VNF instance identifier.

Table 5.5.2.19-1: Definition of the VnfIdentifierDeletionNotification 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 "VnfIdentifierDeletionNotification" 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.
vnfInstanceId	Identifier	1	The deleted VNF instance identifier.
_links	LccnLinks	1	Links to resources related to this notification.

5.5.2.20 Type: CreateVnfSnapshotInfoRequest

This type represents request parameters for the creation of an "Individual VNF snapshot" resource which can be populated with content obtained by invoking the "Create VNF snapshot" LCM operation or extracted from a VNF snapshot package. It shall comply with the provisions defined in table 5.5.2.20-1.

Table 5.5.2.20-1: Definition of the CreateVnfSnapshotInfoRequest data type

Attribute name	Data type	Cardinality	Description
vnfSnapshotPkgId	Identifier	0..1	Identifier of the VNF snapshot package information held by the NFVO. See note.
NOTE: The present attribute shall be provided if the "Individual VNF snapshot" resource is requested to be created and be filled from a VNF snapshot package extraction.			

5.5.2.21 Type: CreateVnfSnapshotRequest

This type represents request parameters for the "Create VNF Snapshot" LCM operation which takes a snapshot of a VNF instance and populates a previously-created "Individual VNF snapshot" resource with the content of the snapshot. It shall comply with the provisions defined in table 5.5.2.21-1.

Table 5.5.2.21-1: Definition of the CreateVnfSnapshotRequest data type

Attribute name	Data type	Cardinality	Description
vnfSnapshotInfoId	Identifier	1	Identifier of the individual VNF snapshot resource to which the VNF Snapshot is to be associated.
vnfcInstanceId	IdentifierInVnf	0..1	Identifier of the VNFC instance to be snapshotted. Each identifier references the "id" attribute in a "VnfcInfo" structure. If this attribute is provided, only a snapshot of the referred VNFC instance shall be created.
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".
userDefinedData	KeyValuePairs	0..1	User defined data for the VNF snapshot.

5.5.2.22 Type: VnfSnapshotInfo

This type represents an "individual VNF snapshot" resource. It shall comply with the provisions defined in table 5.5.2.22-1.

Table 5.5.2.22-1: 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.

Attribute name	Data type	Cardinality	Description
vnfSnapshotPkgId	Identifier	0..1	Identifier of the VNF snapshot package information held by the EM. 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 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.
>takenFrom	Link	0..1	Link to the VNF instance from which this snapshot was taken. Shall be present when the "Individual VNF snapshot" resource is associated to a VNF snapshot created via the corresponding "Create VNF snapshot" task resource.

5.5.2.23 Type: VnfSnapshot

This type represents a VNF snapshot. It shall comply with the provisions defined in table 5.5.2.23-1.

Table 5.5.2.23-1: 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.
vnfInstanceId	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	0..1	Timestamp indicating when the VNF snapshot has been completed by the VNFM. Shall be present once the VNF snapshot creation has been completed.
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.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.

5.5.2.24 Type: RevertToVnfSnapshotRequest

This type represents request parameters for the "Revert-to VNF Snapshot" operation. It shall comply with the provisions defined in table 5.5.2.24-1.

Table 5.5.2.24-1: Definition of the RevertToVnfSnapshotRequest data type

Attribute name	Data type	Cardinality	Description
vnfSnapshotInfoId	Identifier	0..1	Identifier of the "individual VNF snapshot" resource with the information of the VNF snapshot to be reverted to.
vnfcInstanceId	Identifier	0..1	List of identifiers of the VNFC instance to be reverted. Each identifier references the "id" attribute in a "VnfcInfo" structure. Shall be present if the request is for reverting a specific VNFC instance.
vnfcSnapshotInfoId	Identifier	0..1	Identifier of the VNFC snapshot information with the information of the VNFC snapshot to be reverted to. Shall only be present if the "vnfcInstanceId" is present.

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".

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: ExtVirtualLinkData

This type represents an external VL. It shall comply with the provisions defined in table 5.5.3.2-1.

Table 5.5.3.2-1: Definition of the ExtVirtualLinkData data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	The identifier of the external VL instance. The identifier is assigned by the NFV-MANO entity that manages this VL instance.
vimConnectionId	Identifier	0..1	Identifier of the VIM connection to manage this resource. This attribute shall only be supported and present if VNF-related resource management in direct mode is applicable. See note 1.
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. Entries in the list of external CP data that are unchanged need not be supplied if the ExtVirtualLinkData structure is part of a request or response that modifies the external connectivity.
extLinkPorts	ExtLinkPortData	0..N	Externally provided link ports to be used to connect external connection points to this external VL. If this attribute is not present, the VNFM shall create the link ports on the 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 2.

NOTE 1: The information about the VIM connection referenced by the VIM connection id is known to the VNFM. Moreover, the identifier of the VIM connection provides scope to the resourceId.

NOTE 2: 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 addresss 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 addresss.

5.5.3.3 Type: ExtVirtualLinkInfo

This type represents information about an external VL. It shall comply with the provisions defined in table 5.5.3.3-1.

Table 5.5.3.3-1: 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. The identifier is assigned by the NFV-MANO entity that manages this VL instance.
resourceHandle	ResourceHandle	1	Reference to the resource realizing this VL.
extLinkPorts	ExtLinkPortInfo	0..N	Link ports of 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. See note.
NOTE: This attribute reflects the current configuration information that has resulted from merging into this attribute the "VnfExtCpData" information which was passed as part of the "ExtVirtualLinkData" structure in the input of the most recent VNF LCM operation such as "InstantiateVnfRequest", "ChangeExtVnfConnectivityRequest", "ChangeVnfFlavourRequest" or "ChangeCurrentVnfPkgRequest", or has been provided by the NFVO during the granting procedure. If applying such change results in an empty list of "currentVnfExtCpData" structure instances, the affected instance of "ExtVirtualLinkInfo" shall be removed from its parent data structure.			

5.5.3.4 Type: ExtManagedVirtualLinkData

This type represents an externally-managed internal VL. It shall comply with the provisions defined in table 5.5.3.4-1.

Table 5.5.3.4-1: Definition of the ExtManagedVirtualLinkData data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	The identifier of the externally-managed internal VL instance. The identifier is assigned by the NFV-MANO entity that manages this VL instance.
vnfVirtualLinkDescId	IdentifierInVnfd	1	The identifier of the VLD in the VNFD for this VL.
vimConnectionId	Identifier	0..1	Identifier of the VIM connection to manage this resource. This attribute shall only be supported and present if VNF-related resource management in direct mode is applicable. See note.
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.
NOTE: The information about the VIM connection referenced by the VIM connection id is known to the VNFM. Moreover, the identifier of the VIM connection provides scope to the resourceId.			

5.5.3.5 Type: ExtManagedVirtualLinkInfo

This type provides information about an externally-managed virtual link. It shall comply with the provisions defined in table 5.5.3.5-1.

Table 5.5.3.5-1: 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. The identifier is assigned by the NFV-MANO entity that manages this VL instance.
vnfVirtualLinkDescId	IdentifierInVnfd	1	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.

Attribute name	Data type	Cardinality	Description
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).
networkResource	ResourceHandle	1	Reference to the VirtualNetwork resource providing this VL.
vnfLinkPorts	VnfLinkPortInfo	0..N	Link ports of this VL.

5.5.3.6 Type: VnfExtCpData

This type represents configuration information for external CPs created from a CPD. It shall comply with the provisions defined in table 5.5.3.6-1.

Table 5.5.3.6-1: 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 [3]). 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: 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 3: 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 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.			

5.5.3.6a Type: VnfExtCpConfig

This type represents an externally provided link port or network address information per instance of an external connection point. In case a link port is provided, the VNFM shall use that link port when connecting the external CP to the external VL. In case a link port is not provided, the VNFM shall create a link port on the external VL, and use that link port to connect the external CP to the external VL.

This type shall comply with the provisions defined in table 5.5.3.6a-1.

Table 5.5.3.6a-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 sub-port in a trunk, and if parent ports are supported.
linkPortId	Identifier	0..1	Identifier of a pre-configured link port to which the external CP will be associated. See note.
createExtLinkPort	Boolean	0..1	Indicates to the VNFM the need to create a dedicated link port for the external CP. If set to True, the VNFM shall create a link port. If set to False, the VNFM shall not create a link port. 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 note.

NOTE: The following conditions apply to the attributes "linkPortId" and "cpProtocolData":
1) Void.
2) 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.
3) If the "linkPortId" attribute is absent, the VNFM shall create a link port.
4) If the "cpProtocolData" attribute is absent, the "linkPortId" attribute shall be provided referencing a pre-created link port, and the VNFM can use means outside the scope of the present document to obtain the pre-configured address information for the connection point from the resource representing the link port.
5) 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".

5.5.3.6b Type: CpProtocolData

This type represents network protocol data. It shall comply with the provisions defined in table 5.5.3.6b-1.

Table 5.5.3.6b-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 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.

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.

5.5.3.6c Type: IpOverEthernetAddressData

This type represents network address data for IP over Ethernet. It shall comply with the provisions defined in table 5.5.3.6c-1.

Table 5.5.3.6c-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 VIM. See note 1.
segmentationType	Enum	0..1	Specifies the encapsulation type for the traffics coming in and out of the trunk subport. Permitted values: • VLAN: the subport uses VLAN as encapsulation type. • INHERIT: the subport gets its segmentation type from the network it is connected to. 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 CP 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: IPV4, IPV6.
>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.
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.			

5.5.3.7 Type: ScaleInfo

This type represents the scale level of a VNF instance related to a scaling aspect. It shall comply with the provisions defined in table 5.5.3.7-1.

Table 5.5.3.7-1: Definition of the ScaleInfo data type

Attribute name	Data type	Cardinality	Description
aspectId	IdentifierInVnfd	1	Identifier of 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	Indicates the scale level. The minimum value shall be 0 and the maximum value shall be <= maxScaleLevel as described in the VNFD.

5.5.3.8 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 5.5.3.8-1.

Table 5.5.3.8-1: Definition of the VnfcResourceInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of this VnfcResourceInfo instance.
vduId	IdentifierInVnfd	1	Reference to the applicable VDU in the VNFD. See note 1.
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). See note 4.
computeResource	ResourceHandle	1	Reference to the VirtualCompute resource.
storageResourceIds	IdentifierInVnf	0..N	References to the VirtualStorage resources. 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	All CPs of the VNFC instance.
>id	IdentifierInVnf	1	Identifier of this VNFC CP instance and the associated array entry.
>cpdId	IdentifierInVnfd	1	Identifier of the VDU CPD, cpdId, in the VNFD. See note 1.
>vnfExtCpId	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.
>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) of the VNF instance and shall be absent otherwise.
>parentCpId	KeyValuePairs	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.
>metadata	KeyValuePairs	0..1	Metadata about this CP.
metadata	KeyValuePairs	0..1	Metadata about this resource.

NOTE 1: ETSI GS NFV-SOL 001 [i.3] 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: If only the value or the presence of this attribute is changed in the "VnfcResourceInfo" structure by an LCM operation occurrence, this does not represent a change that requires including a related "AffectedVnfc" structure in the VNF LCM operation occurrence notifications or the "VnfLcmOpOcc" structure related to this LCM operation occurrence.

5.5.3.9 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 5.5.3.9-1.

Table 5.5.3.9-1: Definition of the VnfVirtualLinkResourceInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of this VnfVirtualLinkResourceInfo instance.
vnfVirtualLinkDescId	IdentifierInVnfd	1	Identifier of the VNF Virtual Link Descriptor (VLD) 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). See note.
networkResource	ResourceHandle	1	Reference to the VirtualNetwork resource.
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.
NOTE: If only the value or the presence of this attribute is changed in the "VnfVirtualLinkResourceInfo" structure by an LCM operation occurrence, this does not represent a change that requires including a related "AffectedVirtualLink" structure in the VNF LCM operation occurrence notifications or the "VnfLcmOpOcc" structure related to this LCM operation occurrence.			

5.5.3.10 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 5.5.3.10-1.

Table 5.5.3.10-1: 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). See note.
storageResource	ResourceHandle	1	Reference to the VirtualStorage resource.
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.
NOTE: If only the value or the presence of this attribute is changed in the "VirtualStorageResourceInfo" structure by an LCM operation occurrence, this does not represent a change that requires including a related "AffectedVirtualStorage" structure in the VNF LCM operation occurrence notifications or the "VnfLcmOpOcc" structure related to this LCM operation occurrence.			

5.5.3.11 Type: VnfLinkPortInfo

This type represents a link port of an internal VL of a VNF. It shall comply with the provisions defined in table 5.5.3.11-1.

Table 5.5.3.11-1: 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.
cpInstanceId	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. 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 "extCpInfo" item in the VnfInstance or a "vnfcCpInfo" item of a "vnfcResourceInfo" item in the VnfInstance. See note 1.
cpInstanceType	Enum (inlined)	0..1	Type of the CP instance that is identified by cpInstanceId. Shall be present if "cpInstanceId" is present, and shall be absent otherwise. Permitted values: • VNFC_CP: The link port is connected to a VNFC CP. • EXT_CP: The link port is associated to an external CP. See note 1.
vipCpInstanceId	IdentifierInVnf	0..1	VIP CP instance of the VNF connected to this link port. May be present. See notes 1 and 2.
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 3.

NOTE 1: Either cpInstanceId with cpInstanceType set to "EXT_CP" or any combination of cpInstanceId with cpInstanceType set to "VNFC_CP" and vipCpInstanceId (i.e. one or both of them) shall be present for a VnfLinkPortInfo. In case both cpInstanceId with cpInstanceType set to "VNFC_CP" and vipCpInstanceId are present, the two different CP instances share the linkport.
NOTE 2: Clause A.4 of ETSI GS NFV-IFA 007 [i.10] provides examples for configurations where both vipCpInstanceId and vnfcCpInstanceId are present (UC#5 and UC#5-b), only vnfcCpInstanceId is present (UC#2), or only vipCpInstanceId is present (UC6 and UC#6-b).
NOTE 3: The value of "trunkResourceId" is scoped by the value of "vimConnectionId" in the "resourceHandle" attribute.

5.5.3.12 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 5.5.3.12-1.

Table 5.5.3.12-1: 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.
cpInstanceId	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 "extCpInfo" item in the VnfInstance.
secondaryCpInstanceId	IdentifierInVnf	0..1	Additional external CP of the VNF connected to this link port. If present, this attribute shall refer to a "secondary" ExtCpInfo item in the VNF instance that exposes a virtual IP CP instance which shares this linkport with the external CP instance referenced by the "cpInstanceId" 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 clause A.4 of ETSI GS NFV-IFA 007 [i.10] provide examples for such a configuration. NOTE 2: The value of "trunkResourceId" is scoped by the value of "vimConnectionId" in the "resourceHandle" attribute.			

5.5.3.12a Type: ExtLinkPortData

This type represents an externally provided link port to be used to connect an external connection point to an external VL. It shall comply with the provisions defined in table 5.5.3.12a-1.

Table 5.5.3.12a-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.			

5.5.3.13 Type: ResourceHandle

This type represents the information that allows addressing a virtualised resource that is used by a VNF instance. Information about the resource is available from the VIM. The ResourceHandle type shall comply with the provisions defined in table 5.5.3.13-1.

Table 5.5.3.13-1: Definition of the ResourceHandle data type

Attribute name	Data type	Cardinality	Description
vimConnectionId	Identifier	0..1	Identifier of the VIM connection to manage the resource. This attribute shall only be supported and present if VNF-related resource management in direct mode is applicable. See note 1.
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 or the resource provider.
vimLevelResourceType	String	0..1	Type of the resource in the scope of the VIM or the resource provider. See note 2.
NOTE 1: The information about the VIM connection referenced by the VIM connection id is known to the VNFM. Moreover, the identifier of the VIM connection provides scope to the resourceId. NOTE 2: The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.			

5.5.3.14 Void

5.5.3.15 Void

5.5.3.15a Type: CpProtocolInfo

This type describes the protocol layer(s) that a CP uses together with protocol-related information, like addresses. It shall comply with the provisions defined in table 5.5.3.15a-1.

Table 5.5.3.15a-1: Definition of the CpProtocolInfo data type

Attribute name	Data type	Cardinality	Description
layerProtocol	Enum (inlined)	1	The identifier of layer(s) and protocol(s) associated to the network address information. Permitted values: IP_OVER_ETHERNET. See note.
ipOverEthernet	IpOverEthernetAddressInfo	0..1	IP addresses over Ethernet to assign to the extCP instance. Shall be present if layerProtocol is equal to "IP_OVER_ETHERNET", and shall be absent 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.			

5.5.3.16 Type: IpOverEthernetAddressInfo

This type represents information about a network address that has been assigned. It shall comply with the provisions defined in table 5.5.3.16-1.

Table 5.5.3.16-1: Definition of the IpOverEthernetAddressInfo data type

Attribute name	Data type	Cardinality	Description
macAddress	MacAddress	0..1	MAC address if assigned. 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 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.
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.			

5.5.3.17 Type: MonitoringParameter

This type represents a monitoring parameter that is tracked by the VNFM, e.g. for auto-scaling purposes. It shall comply with the provisions defined in table 5.5.3.17-1.

Valid monitoring parameters of a VNF are defined in the VNFD.

NOTE: ETSI GS NFV-SOL 001 [i.3] specifies the structure and format of the VNFD based on TOSCA specifications.

Table 5.5.3.17-1: Definition of the MonitoringParameter 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 [5].

5.5.3.18 Type: LifecycleChangeNotificationsFilter

This type represents a subscription filter related to notifications about VNF lifecycle changes. It shall comply with the provisions defined in table 5.5.3.18-1.

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 5.5.3.18-1: Definition of the LifecycleChangeNotificationsFilter data type

Attribute name	Data type	Cardinality	Description
vnfInstanceSubscriptionFilter	VnfInstanceSubscriptionFilter	0..1	Filter criteria to select VNF instances about which to notify.
notificationTypes	Enum (inlined)	0..N	Match particular notification types. Permitted values: • VnfLcmOperationOccurrenceNotification • VnfIdentifierCreationNotification • VnfIdentifierDeletionNotification See note.
operationTypes	LcmOperationType	0..N	Match particular VNF lifecycle operation types for the notification of type VnfLcmOperationOccurrenceNotification. May be present if the "notificationTypes" attribute contains the value "VnfLcmOperationOccurrenceNotification", and shall be absent otherwise.
operationStates	LcmOperationStateType	0..N	Match particular LCM operation state values as reported in notifications of type VnfLcmOperationOccurrenceNotification. May be present if the "notificationTypes" attribute contains the value "VnfLcmOperationOccurrenceNotification", 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.			

5.5.3.19 Type: AffectedVnfc

This type provides information about added, deleted, modified and temporary VNFCs. It shall comply with the provisions in table 5.5.3.19-1.

Table 5.5.3.19-1: Definition of the AffectedVnfc data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of the Vnfc instance, identifying the applicable "vnfcResourceInfo" entry in the "VnfInstance" data type (see clause 5.5.2.2).
vduId	IdentifierInVnfd	1	Identifier of the related VDU in the VNFD.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case of a "change current VNF Package" to identify whether the affected VNFC instance is associated to a VDU which is referred from the source or destination VNFD.
changeType	Enum (inlined)	1	Signals the type of change. Permitted values: • ADDED • REMOVED • MODIFIED • TEMPORARY For a temporary resource, an AffectedVnfc structure exists as long as the temporary resource exists.

Attribute name	Data type	Cardinality	Description
computeResource	ResourceHandle	1	Reference to the VirtualCompute resource. Detailed information is (for new and modified resources) or has been (for removed resources) available from the VIM.
metadata	KeyValuePairs	0..1	Metadata about this resource. The content of this attribute shall be a copy of the content of the "metadata" attribute of the VnfcResourceInfo structure.
affectedVnfcCpIds	IdentifierInVnf	0..N	Identifiers of CP(s) of the VNFC instance that were affected by the change.
addedStorageResourceIds	IdentifierInVnf	0..N	References to VirtualStorage resources that have been added. Each value refers to a VirtualStorageResourceInfo item in the VnfInstance that was added to the VNFC. It shall be provided if at least one storage resource was added to the VNFC.
removedStorageResourceIds	IdentifierInVnf	0..N	References to VirtualStorage resources that have been removed. The value contains the identifier of a VirtualStorageResourceInfo item that has been removed from the VNFC, and might no longer exist in the VnfInstance. It shall be provided if at least one storage resource was removed from the VNFC.

5.5.3.20 Type: AffectedVirtualLink

This type provides information about added, deleted, modified and temporary VLs, and added or removed VNF link ports. It shall comply with the provisions in table 5.5.3.20-1.

Table 5.5.3.20-1: Definition of the AffectedVirtualLink data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of the virtual link instance, identifying the applicable "vnfVirtualLinkResourceInfo" or "extManagedVirtualLinkInfo" entry in the "VnfInstance" data type (see clause 5.5.2.2).
vnfVirtualLinkDescId	IdentifierInVnfd	1	Identifier of the related VLD in the VNFD.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case of a "change current VNF Package" to identify whether the affected VL instance is associated to a VLD which is referred from the source or destination VNFD.
changeType	Enum (inlined)	1	Signals the type of change. Permitted values: • ADDED • REMOVED • MODIFIED • TEMPORARY • LINK_PORT_ADDED • LINK_PORT_REMOVED For a temporary resource, an AffectedVirtualLink structure exists as long as the temporary resource exists. See note.
networkResource	ResourceHandle	1	Reference to the VirtualNetwork resource. Detailed information is (for new and modified resources) or has been (for removed resources) available from the VIM. See note.

Attribute name	Data type	Cardinality	Description
vnfLinkPortIds	IdentifierInVnf	0..N	Identifiers of the link ports of the affected VL related to the change. Each identifier references a "VnfLinkPortInfo" structure. Shall be set when changeType is equal to "LINK_PORT_ADDED" or "LINK_PORT_REMOVED", and the related "VnfLinkPortInfo" structures are present (case "added") or have been present (case "removed") in the "VnfVirtualLinkResourceInfo" or "ExtManagedVirtualLinkInfo" structures that are represented by the "vnfVirtualLinkResourceInfo" or "extManagedVirtualLinkInfo" attribute in the "VnfInstance" structure. See note.
metadata	KeyValuePairs	0..1	Metadata about this resource. The content of this attribute shall be a copy of the content of the "metadata" attribute of the applicable "VnfVirtualLinkResourceInfo" structure if such structure is referenced by the "id" attribute and it has metadata.
NOTE: When signalling the addition (LINK_PORT_ADDED) or removal (LINK_PORT_REMOVED) of VNF link ports, the "networkResource" attribute refers to the affected virtual link instance, not the link port instance. The resource handles of the affected VNF link ports can be found by dereferencing the identifiers in the "vnfLinkPortIds" attribute.			

5.5.3.20a Type: AffectedExtLinkPort

This type provides information about added and deleted external link ports (link ports attached to external virtual links). It shall comply with the provisions in table 5.5.3.20a-1.

Table 5.5.3.20a-1: Definition of the AffectedExtLinkPort data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of the link port, identifying the applicable "extLinkPorts" entry in the "ExtVirtualLinkInfo" data type (see clause 5.5.3.3).
changeType	Enum (inlined)	1	Signals the type of change. Permitted values: • ADDED • MODIFIED • REMOVED
extCpInstanceId	IdentifierInVnf	1	Identifier of the related external CP instance.
resourceHandle	ResourceHandle	1	Reference to the link port resource. Detailed information is (for added resources) or has been (for removed resources) available from the VIM.

5.5.3.20b Type: AffectedVipCp

This type provides information about added, deleted and modified virtual IP CP instances. It shall comply with the provisions in table 5.5.3.20b-1.

Table 5.5.3.20b-1: Definition of the AffectedVipCp data type

Attribute name	Data type	Cardinality	Description
cpInstanceId	IdentifierInVnf	1	Identifier of the virtual IP CP instance and the related "VipCpInfo" structure in "VnfInstance".
cpdId	IdentifierInVnfd	1	Identifier of the VipCpd in the VNFD.
vnfdId	Identifier	0..1	Reference to the VNFD. Shall be present in case of a "change current VNF Package" to identify whether the affected virtual CP instance is associated to a VipCpd which is referred from the source or destination VNFD.
changeType	Enum (inlined)	1	Signals the type of change. Permitted values: • ADDED • REMOVED • MODIFIED

5.5.3.21 Type: AffectedVirtualStorage

This type provides information about added, deleted, modified and temporary virtual storage resources. It shall comply with the provisions in table 5.5.3.21-1.

Table 5.5.3.21-1: Definition of the AffectedVirtualStorage data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of the storage instance, identifying the applicable "virtualStorageResourceInfo" entry in the "VnfInstance" data type (see clause 5.5.2.2).
virtualStorageDescId	IdentifierInVnfd	1	Identifier of the related VirtualStorage descriptor in the VNFD.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case of a "change current VNF Package" to identify whether the affected virtual storage instance is associated to a VirtualStorage descriptor which is referred from the source or destination VNFD.
changeType	Enum (inlined)	1	Signals the type of change. Permitted values: • ADDED • REMOVED • MODIFIED • TEMPORARY For a temporary resource, an AffectedVirtualStorage structure exists as long as the temporary resource exists.
storageResource	ResourceHandle	1	Reference to the VirtualStorage resource. Detailed information is (for new and modified resources) or has been (for removed resources) available from the VIM.
metadata	KeyValuePairs	0..1	Metadata about this resource. The content of this attribute shall be a copy of the content of the "metadata" attribute of the VirtualStorageResourceInfo structure.

5.5.3.22 Type: LccnLinks

This type represents the links to resources that a notification can contain. It shall comply with the provisions defined in table 5.5.3.22-1.

Table 5.5.3.22-1: Definition of the LccnLinks data type

Attribute name	Data type	Cardinality	Description
vnfInstance	NotificationLink	1	Link to the resource representing the VNF instance to which the notified change applies.
subscription	NotificationLink	1	Link to the related subscription.
vnfLcmOpOcc	NotificationLink	0..1	Link to the VNF lifecycle management operation occurrence that this notification is related to. Shall be present if there is a related lifecycle operation occurrence.

5.5.3.23 Type: VnfcInfo

This type represents the information about a VNFC instance that is part of a VNF instance. It shall comply with the provisions defined in table 5.5.3.23-1.

Table 5.5.3.23-1: Definition of the VnfcInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of the VNFC instance.
vduId	IdentifierInVnfd	1	Reference to the applicable VDU information element in the VNFD.
vnfcResourceInfoId	IdentifierInVnf	0..1	Identifier of the VnfcResourceInfo instance representing the virtualised resources used by this VNFC instance. Shall be present in case a corresponding VnfcResourceInfo instance exists. See note.
vnfcState	Enum (inlined)	1	State of the VNFC instance. Permitted values: • STARTED: The VNFC instance is up and running. • STOPPED: The VNFC instance has been shut down.
vnfcConfigurableProperties	KeyValuePairs	0..1	Current values of the configurable properties of the VNFC instance. Configurable properties referred in this attribute are declared in the VNFD. This attribute can be modified with the PATCH method. In addition, the provisions in clause 5.7 shall apply.
NOTE: This allows to represent the error condition that a VNFC instance has lost its resources.			

5.5.3.24 Type: VnfcInfoModifications

This type represents modifications of an entry in an array of "VnfcInfo" objects. It shall comply with the provisions defined in table 5.5.3.24-1.

Table 5.5.3.24-1: Definition of the VnfcInfoModifications data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of the VNFC instance of which the information is to be modified. The identifier references the "id" attribute in a "VnfcInfo" structure. See note.
vnfcConfigurableProperties	KeyValuePairs	1	Changes of the configurable properties of the VNFC instance. When this structure is part of a request, the modifications signalled in this attribute shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [3]). In addition, the provisions in clause 5.7 shall apply.
NOTE: The attribute "id" in this data type represents the same identifier as the attribute "vnfcInstanceId" in other related data types in the present document. For reasons of backward compatibility, this misalignment is not corrected.			

5.5.3.25 Type: VnfExtCpInfo

This type represents information about an external CP of a VNF. It shall comply with the provisions defined in table 5.5.3.25-1.

Table 5.5.3.25-1: Definition of the VnfExtCpInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of the external CP instance and the related information instance.
cpdId	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.
metadata	KeyValuePairs	0..1	Metadata about this external CP.
associatedVnfcCpId	IdentifierInVnf	0..1	Identifier of the "vnfcCpInfo" 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.
associatedVipCpId	IdentifierInVnf	0..1	Identifier of the VIP CP instance that is exposed as this VnfExtCp instance, either directly or via a floating IP address, and the related "VipCpInfo" structure in "VnfInstance". Shall be present if the cpdId of this VnfExtCp has a vipCpd attribute. See note 1.
associatedVnfVirtual LinkId	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 "associatedVnfcCpId", "associatedVipCpId" 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 5.5.3.2.			

5.5.3.26 Type: VnfcSnapshotInfo

This type represents a VNFC snapshot. It shall comply with the provisions defined in table 5.5.3.26-1.

Table 5.5.3.26-1: 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. The identifier references the "id" attribute in a "VnfcInfo" structure.
creationStartedAt	DateTime	1	Timestamp indicating when the VNF snapshot creation has been started by the VNFM.
creationFinishedAt	DateTime	0..1	Timestamp indicating when the VNFC snapshot has been completed. Shall be present once the VNFC snapshot creation has been completed by the VNFM.
vnfcResourceInfoId	IdentifierInVnf	1	Reference to the "VnfcResourceInfo" structure in the "VnfInstance" 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.
>storageResourceId	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.
>storageSnapshotResource	ResourceHandle	0..1	Reference to a storage snapshot resource. See note 2.
userDefinedData	KeyValuePairs	0..1	User defined data for the VNF 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".

5.5.3.27 Type: ModificationsTriggeredByVnfPkgChange

This type represents attribute modifications that were performed on an "Individual VNF instance" resource when changing the current VNF package. The attributes that can be included consist of those requested to be modified explicitly in the "ChangeCurrentVnfPkgRequest" data structure, and additional attributes of the "VnfInstance" data structure that were modified implicitly during the operation.

The "ModificationsTriggeredByVnfPkgChange" data type shall comply with the provisions defined in table 5.5.3.27-1.

Table 5.5.3.27-1: 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 5.7 shall apply.

Attribute name	Data type	Cardinality	Description
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 5.7 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 [3], 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.			

5.5.3.28 Type: VipCpInfo

Thi information element provides information related to virtual IP (VIP) CP. It shall comply with the provisions defined in table 5.5.3.28-1.

Table 5.5.3.28-1: 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.
cpdId	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). See note 2.
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 addresse allocated to the VIP CP instance. See note 1.
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	KeyValuePairs	0..N	Metadata about this VIP CP.

NOTE 1: It is possible that there is no associated VnfcCp because the VIP CP is available but not associated yet.
NOTE 2: If only the value or the presence of this attribute is changed in the "VipCpInfo" structure by an LCM operation occurrence, this does not represent a change that requires including a related "AffectedVipCp" structure in the VNF LCM operation occurrence notifications or the "VnfLcmOpOcc" structure related to this LCM operation occurrence.

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.

5.5.4.3 Enumeration: VnfOperationalStateType

The enumeration VnfOperationalStateType shall comply with the provisions defined in table 5.5.4.3-1.

Table 5.5.4.3-1: Enumeration VnfOperationalStateType

Enumeration value	Description
STARTED	The VNF instance is up and running.
STOPPED	The VNF instance has been shut down.

5.5.4.4 Enumeration: StopType

The enumeration StopType shall comply with the provisions defined in table 5.5.4.4-1.

Table 5.5.4.4-1: Enumeration StopType

Enumeration value	Description
FORCEFUL	The VNFM will stop the VNF instance or VNFC instance(s) immediately after accepting the request.
GRACEFUL	The VNFM will first arrange to take the VNF instance or VNFC instance(s) 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 instance or VNFC instance(s).

5.5.4.5 Enumeration: LcmOperationType

The enumeration LcmOpType defines the permitted values to represent VNF lifecycle operation types in VNF lifecycle management operation occurrence resources and VNF lifecycle management operation occurrence notifications. It shall comply with the provisions defined in table 5.5.4.5-1.

Table 5.5.4.5-1: Enumeration LcmOperationType

Enumeration value	Description
INSTANTIATE	Represents the "Instantiate VNF" LCM operation.
SCALE	Represents the "Scale VNF" LCM operation.
SCALE_TO_LEVEL	Represents the "Scale VNF to Level" LCM operation.
CHANGE_FLAVOUR	Represents the "Change VNF Flavour" LCM operation.
TERMINATE	Represents the "Terminate VNF" LCM operation.
HEAL	Represents the "Heal VNF" LCM operation.
OPERATE	Represents the "Operate VNF" LCM operation.
CHANGE_EXT_CONN	Represents the "Change external VNF connectivity" LCM operation.
MODIFY_INFO	Represents the "Modify VNF Information" LCM operation.
CREATE_SNAPSHOT	Represents the "Create VNF Snapshot" LCM operation.
REVERT_TO_SNAPSHOT	Represents the "Revert-To VNF Snapshot" LCM operation.
CHANGE_VNFPKG	Represents the "Change current VNF package" LCM operation.

5.5.4.6 Enumeration: LcmOperationStateType

The enumeration LcmOperationStateType shall comply with the provisions defined in table 5.5.4.6-1. More information of the meaning of the states can be found in clause 5.6.2.2.

Table 5.5.4.6-1: Enumeration LcmOperationStateType

Enumeration value	Description
STARTING	The LCM operation is starting.
PROCESSING	The LCM operation is currently in execution.
COMPLETED	The LCM operation has been completed successfully.
FAILED_TEMP	The LCM operation has failed and execution has stopped, but the execution of the operation is not considered to be closed.
FAILED	The 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 LCM operation is currently being rolled back.
ROLLED_BACK	The LCM operation has been successfully rolled back, i.e. The state of the VNF prior to the original operation invocation has been restored as closely as possible.

5.5.4.7 Enumeration: CancelModeType

The enumeration CancelModeType defines the valid modes of cancelling a VNF LCM operation occurrence. It shall comply with the provisions defined in table 5.5.4.7-1.

Table 5.5.4.7-1: Enumeration CancelModeType

Enumeration value	Description
GRACEFUL	If the VNF LCM operation occurrence is in "PROCESSING" or "ROLLING_BACK" state, the VNFM shall not start any new resource management operation and shall wait for the ongoing resource management operations in the underlying system, typically the VIM, to finish execution or to time out. After that, the VNFM shall put the operation occurrence into the FAILED_TEMP state. If the VNF LCM operation occurrence is in "STARTING" state, the VNFM shall not start any resource management operation and shall wait for the granting request to finish execution or time out. After that, the VNFM shall put the operation occurrence into the ROLLED_BACK state.
FORCEFUL	If the VNF LCM operation occurrence is in "PROCESSING" or "ROLLING_BACK" state, the VNFM shall not start any new resource management operation, shall cancel the ongoing resource management operations in the underlying system, typically the VIM, and shall wait for the cancellation to finish or to time out. After that, the VNFM shall put the operation occurrence into the FAILED_TEMP state. If the VNF LCM operation occurrence is in "STARTING" state, the VNFM shall not start any resource management operation and put the operation occurrence into the ROLLED_BACK state.

5.5.4.8 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 5.5.4.8-1.

Table 5.5.4.8-1: Enumeration VnfOperationalStateType

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.

5.6 Success and error states of VNF lifecycle management operations

5.6.1 Basic concepts for error handling (informative)

5.6.1.1 Motivation

VNF 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.

5.6.1.2 Failure resolution strategies: Retry and Rollback

Most transient errors are handled best with a retry mechanism. Retry might happen automatically at the point of failure within the same 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 resolution of the inconsistent state can be attempted by requesting to roll back the changes made by the operation. Therefore, rollback as an error handling strategy is 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 rollback if the retry has failed. Idempotent retry is meaningful and useful for all operation types, but 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 InstantiateVnf or scale out, while ill-suited for subtractive ones such as scale in or TerminateVnf, or for HealVnf.

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 EM, NFVO or human operator.

5.6.1.3 Error handling at VNFM and EM

If the VNFM executes an LCM workflow and encounters a problem, the following options are possible:

- Stop on first error:
  - Once the VNFM encounters an error, the normal execution of the LCM workflow is interrupted, and an error handling procedure is triggered (automatic retry, automatic rollback, automatic fail, escalate). See the paragraphs below for description of error handling procedures.
  - It is assumed that all VNFs and all VNFMs support "stop on first error".

EXAMPLE 1: EM is attempting to instantiate a VNF with 100 VNFCs. The first 97 VNFCs are instantiated successfully, however, an error occurs when attempting to instantiate VNFC #98. The VNFM 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 VNFM encounters an error, it is decided whether the execution of a part or all of the remaining steps of the LCM workflow is performed, or whether the execution is interrupted and an error handling procedure is triggered (automatic retry, automatic rollback, automatic fail, escalate). See the paragraphs below for description of error handling procedures.
  - Support of "best effort" requires a suitable workflow design.
  - It is therefore assumed that not all VNFs and not all VNFMs support "best effort".

EXAMPLE 2: Same example as above. After the error occurs attempting to instantiate VNFC #98, the VNFM continues by creating #99 and #100, and then chooses which error handling options it invokes.

The VNFM has the following error handling procedures to react to errors (see clause 5.6.1.2 for general elaboration regarding retry and rollback):

- Automatic Retry: The VNFM 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 LCM operation to be reported as successful. Failed retry is typically escalated.
- Automatic Rollback: The VNFM rolls back the VNF to the state prior to starting the LCM operation without involving an external entity. Rollback can eventually succeed or can fail, preventing the VNF from reaching that previous state. Successful rollback leads to the LCM operation to be reported as rolled back. Failed rollback is typically escalated.
- Escalate: After failed automatic retry/retries, automatic rollback is typically not the first option in most situations, but the error is preferably reported to the EM for further resolution. The same applies if no automatic error resolution was attempted by the VNFM, or if automated rollback has failed. This is done by sending a VNF LCM operation occurrence notification.

- Unresolvable Error: The VNFM determines that the operation has failed and definitely cannot be recovered (e.g. if no retry and no rollback is possible), and that escalating the error will have no chance to lead to a resolution either. In this case, the VNFM would report that the operation has terminally failed. After that, other means of resolution can be attempted, such as the invocation of HealVnf, or manual procedures using the GUI of the VNFM or VIM to release stranded resources.

The EM has the following error handling procedures to react to error reports from the VNFM:

- On-demand retry: After the VNFM has reported the error to the EM, the EM 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 VNFM by the EM via the VNF LCM interface.
- On-demand rollback: After the VNFM has reported the error to the EM, and after the EM or the human operator has decided to roll back the operation, the rollback of the operation is triggered towards the VNFM by the EM via the VNF LCM interface.
- Fail: After the VNFM has reported the error to the EM, and after the EM or the human operator has determined that neither on-demand retry nor on-demand rollback will fix the error, the LCM operation can be declared as terminally failed towards the VNFM via the VNF LCM interface. After that, other means of resolution can be attempted, such as the invocation of HealVnf, or manual procedures using the GUI of the VNFM or VIM to release stranded resources.

NOTE: Error handling by the EM can involve escalations to the OSS/BSS, or to the NFVO via the OSS/BSS.

5.6.2 States and state transitions of a VNF lifecycle management operation occurrence

5.6.2.1 General

A VNF lifecycle management operation occurrence supports a number of states and error handling operations. The states and state transitions that shall be supported by the VNFM are shown in figure 5.6.2.1-1. Transitions labelled with underlined text represent error handling operations; other transitions represent conditions.

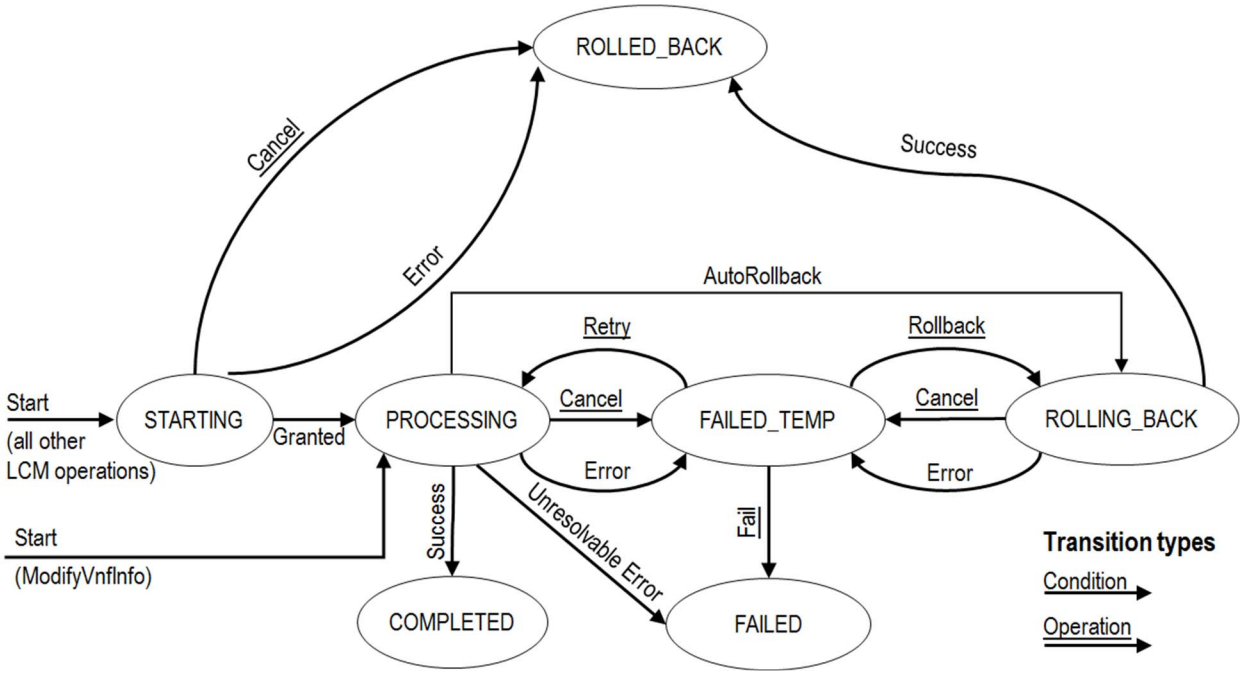

Figure 5.6.2.1-1: States of a VNF lifecycle management operation occurrence

5.6.2.2 States of a VNF lifecycle management operation occurrence

At each time, a VNF lifecycle management operation occurrence is in one of the following states. There are transient 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 a VNF lifecycle management operation occurrence in a terminal state cannot change anymore).

STARTING: The operation is starting. This state represents the preparation phase of the operation, including invoking Grant Lifecycle Operation. This state has the following characteristics:

- This is the initial state for any LCM operation except ModifyVnfInformation.
- This is a transient state.
- This state may block other LCM operations from being executed on the same VNF instance (up to VNF and VNFM implementation).
- In this state, the VNF lifecycle management operation occurrence does not perform any changes to the VNF instance or to resources.
- Once the VNF lifecycle operation has been granted, the VNF lifecycle management operation occurrence shall transit into the PROCESSING state.
- If the LCM operation is cancelled in the "STARTING" state, the VNF lifecycle management operation occurrence shall transit to the "ROLLED_BACK" state. The EM/VNF shall be prepared to receive the notification about the cancellation of the operation before and after having provided the grant. This is necessary to address possible race conditions.
- If an error occurs before the VNFM receives the grant response, or the grant is rejected, as no changes to the underlying VNF or resources were done, the VNF lifecycle management operation occurrence shall transit into the "ROLLED_BACK" state.

COMPLETED: The operation has completed successfully. This is a terminal state.

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.
- The grant received for the operation is still valid, and the granted resource changes are still foreseen for the VNF.
- This state may block other LCM operations from being executed on the same VNF instance (enforced by the VNFM, and up to VNF and VNFM capabilities).
- Retry or rollback or fail may be invoked for the operation.
- If the VNF LCM operation is retried, the VNF lifecycle management operation occurrence shall transit into the "PROCESSING" state.
- If the VNF LCM operation is rolled back, the VNF lifecycle management operation occurrence shall transit into the "ROLLING_BACK" state.
- If the VNF LCM operation is marked as "failed", the VNF 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 or rolled back, 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 (EM/VNF) or its human operator that an operation in FAILED_TEMP state cannot be retried or rolled back ("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 nor rolled back ("Unresolvable Error").

NOTE 1: The direct transition from "PROCESSING" into "FAILED" state is deprecated and only provided for backward compatibility with legacy; implementations need to be aware that support can be removed in subsequent versions of the present document.

- The result of the LCM operation (the actual resource changes) can show an inconsistent state of the VNF, and can reflect partial resource changes compared to the granted changes. Nevertheless, these resource changes, as known by the VNFM shall be synchronized between the VNFM and EM/VNF (by reporting them in the LCCN, and by allowing the EM/VNF to obtain them on request) in order for other VNF LCM operations (e.g. Heal, Terminate) to be guaranteed to work on resources that are known to the EM/VNF.

NOTE 2: In certain error cases during a procedure that requires interactions with the VIM, the information about VIM resources known by the VNFM might not be accurate.

- The fact that an LCM operation is in "FAILED" state shall not block other operations from execution on the VNF instance by the VNFM. However, the VNF instance may itself be in a state that disallows certain operations.

ROLLED_BACK: The state of the VNF 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.

PROCESSING: The LCM operation is currently in execution. This state has the following characteristics:

- This is the initial state for the "ModifyVnfInformation" operation.

- This is a transient state.

- This state may block other LCM operations from being executed on the same VNF instance (up to VNF and VNFM implementation).

- The operations "Retry" and "Rollback" shall not be permitted to be invoked for an operation that is in this state.

- All failures of procedures executed by the VNFM as part of the LCM operation while in "PROCESSING" state shall result in transiting to "FAILED_TEMP", with the following two exceptions:

  - If a failure occurs in the "PROCESSING" state from which the VNFM knows that the VNF instance can be brought into a consistent state by immediately rolling back the operation, the VNF lifecycle management operation occurrence may transit directly into the "ROLLING_BACK" state ("AutoRollback"). For the "ModifyVnfInformation" operation, AutoRollback is the typical error handling method.

  - If a failure occurs in the "PROCESSING" state from which the VNFM knows that it can neither be fixed by retrying nor be rolled back, the VNF lifecycle management operation occurrence may transit directly into the "FAILED" state ("Unresolvable Error").

NOTE 3: The direct transition from "PROCESSING" into "FAILED" state is deprecated and only provided for backward compatibility with legacy; implementations need to be aware that support can be removed in subsequent versions of the present document.

- 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 VNF lifecycle management operation occurrence shall transit into the "FAILED_TEMP" state.

ROLLING_BACK: The LCM operation is currently being rolled back. This state has the following characteristics:

- This is a transient state.
- This state may block other LCM operations from being executed on the same VNF instance (up to VNF and VNFM implementation).
- The operations "Retry" 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 VNF 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 VNF lifecycle management operation occurrence shall transit into the "ROLLED_BACK" state.

The following provisions apply to the sending of VNF lifecycle management operation occurrence notifications by the VNFM:

- The "start" notification (i.e. notificationStatus="START") shall be sent each time when the operation enters one of states "STARTING", "PROCESSING" and "ROLLING_BACK" from another state, indicating the state entered in the "operationState" attribute.
- The "result" notification (i.e. notificationStatus="RESULT") shall be sent each time when the VNF LCM operation occurrence enters one of the error states "FAILED_TEMP", "FAILED", "ROLLED_BACK", indicating the state entered in the "operationState" attribute, as well as the error cause and the changes to the VNF'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", indicating the state entered in the "operationState" attribute, as well as the changes to the VNF's resources.

The following provisions apply to the sending of notifications related to VNF lifecycle changes (VNF LCM operation Occurrence Notifications, VNF identifier creation and VNF identifier deletion notifications):

- The processing of a VNF LCM operation occurrence shall not wait for the acknowledgement of the delivery of the triggered notifications.
- Invoking a subsequent LCM operation on the same VNF 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 VNF instance.

Such a notification scheme allows the EM/VNF to keep in sync with changes to the VNF's resources by an ongoing 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 VNF's resources will be performed by the related VNF lifecycle management operation occurrence. In order to avoid inconsistent information about the state and result of the VNF lifecycle management operation by the EM/VNF, which can impact the error handling procedure, the state of the VNF lifecycle management operation shall be synchronized between the VNFM and EM/VNF. The EM/VNF can use the information in the notification to synchronize its internal state with the current state and result of the LCM operation. In case of loss of notifications, the EM/VNF can read the resource that represents the VNF lifecycle management operation occurrence to obtain the same information.

5.6.2.3 Error handling operations that change the state of a VNF lifecycle management operation occurrence

Retry: This operation retries a VNF lifecycle operation. It has the following characteristics:

- Execution of "Retry" for an actual LCM operation on a particular VNF may be supported, depending on characteristics of the VNF and the LCM operation.

- "Retry" shall operate within the bounds of the Grant for the LCM operation.
- The operation may be invoked via an interface, or the VNFM may invoke the operation per its own decision.

Rollback: This operation rolls back a VNF lifecycle operation. It has the following characteristics:

- Execution of "Rollback" for an actual LCM operation on a particular VNF may be supported, depending on characteristics of the VNF and the LCM operation.
- "Rollback" shall operate within the bounds of the Grant for the LCM operation, an additionally may execute the inverse of granted LCM operations (e.g. if a resource deletion was granted, rollback might re-create the deleted resource or a similar resource).
- The operation may be invoked via an interface, or the VNFM may invoke the operation per its own decision.

Fail: This operation transits the VNF lifecycle management operation occurrence into the terminal "FAILED" state. It has the following characteristics:

- Execution of "Fail" shall be supported for an LCM operation on a particular VNF if at least one of Retry, Rollback, Cancel is supported for this operation.
- The operation may be invoked via an interface, or the VNFM may invoke the operation per its own decision.

Cancel: This operation cancels an ongoing VNF lifecycle management operation, its Retry or Rollback. It has the following characteristics:

- Execution of Cancel for an actual LCM operation on a particular VNF may be supported, depending on characteristics of the VNF and the 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 VNFM will not initiate any new operation towards the underlying systems, will wait until the currently executed operations finish, fail or time out in the VNFM, and will then put the VNF lifecycle management operation occurrence into the "FAILED_TEMP" state.
  - When executing the *forceful* "Cancel" operation, the VNFM 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, fail or time out in the VNFM, and will then put the VNF lifecycle management operation occurrence into the "FAILED_TEMP" state.

  NOTE: In both modes, the time-out is determined by means outside the scope of the present document.

  - In "STARTING" state, there is no difference between the graceful and the forceful cancellation mode.
- Executing "Cancel" can lead to inconsistencies between the information that the VNFM has about the state of the resources of the VNF, and their actual state. The probability of such inconsistencies is bigger when using the *forceful* cancellation mode.

5.6.3 Detailed flows for error handling

5.6.3.1 Immediate failure

If the VNF LCM operation fails immediately, i.e. it returns an HTTP error, then the operation has not started, and no "Individual VNF LCM operation occurrence" resource has been created. Also, neither a "start" VNF lifecycle management operation occurrence notification nor a Grant request has been sent. The operation cannot be retried, but the same operation may be invoked again from the API. The VNF instance is not changed by a synchronous failure, so no special error handling is required.

Figure 5.6.3.1-1 illustrates the flow.

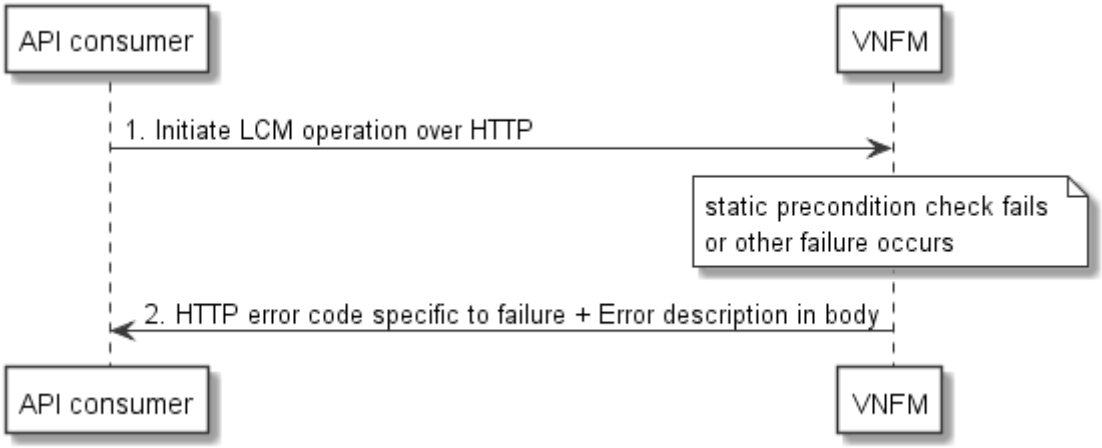

Figure 5.6.3.1-1: Immediate failure of a VNF LCM operation

5.6.3.2 Failure in "STARTING" state

This error scenario assumes that the "Individual VNF LCM operation occurrence" resource has been created and the "start" VNF lifecycle management operation occurrence notification has been sent.

If the operation fails before the VNFM receives the Grant response, or the Grant is rejected, persistent change to the state of the VNF cannot have happened. Therefore, it is assumed that this operation enters the ROLLED_BACK state immediately. Figure 5.6.3.2-1 illustrates the flow.

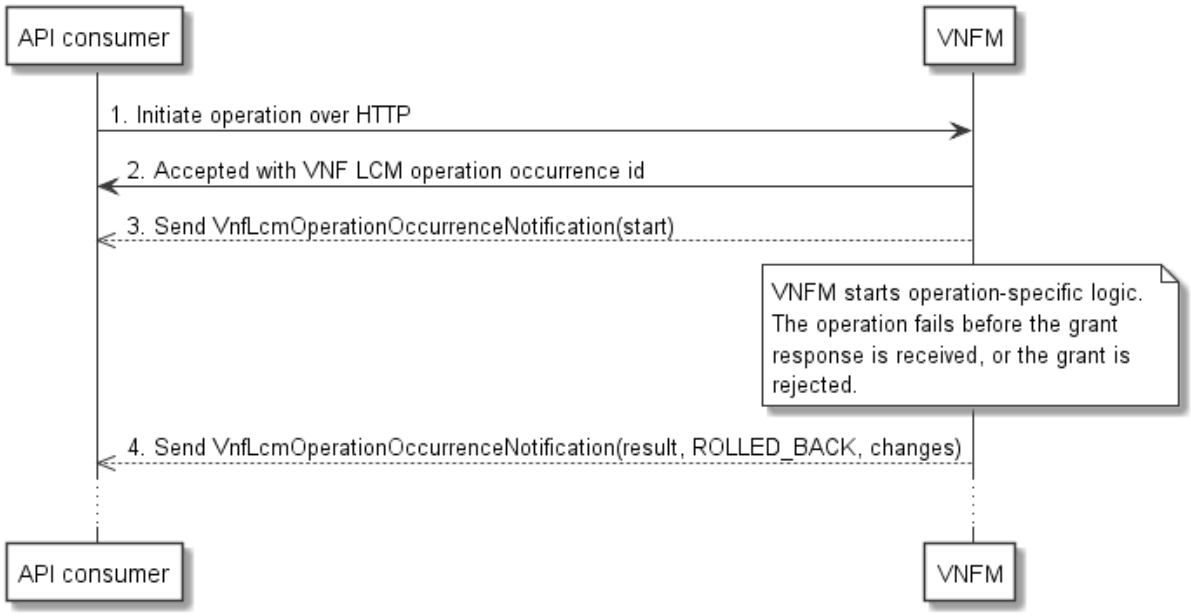

Figure 5.6.3.2-1: Failure of a VNF LCM operation before applying any change to the VNF instance

5.6.3.3 Failure during actual LCM operation execution

After a failed resource management operation, automatic retry can be invoked by the VNFM itself. These invocations are not visible outside of the VNFM, as the VNF 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 LCM operation is set into a temporary failure state, and the EM 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 5.6.3.3-1 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.

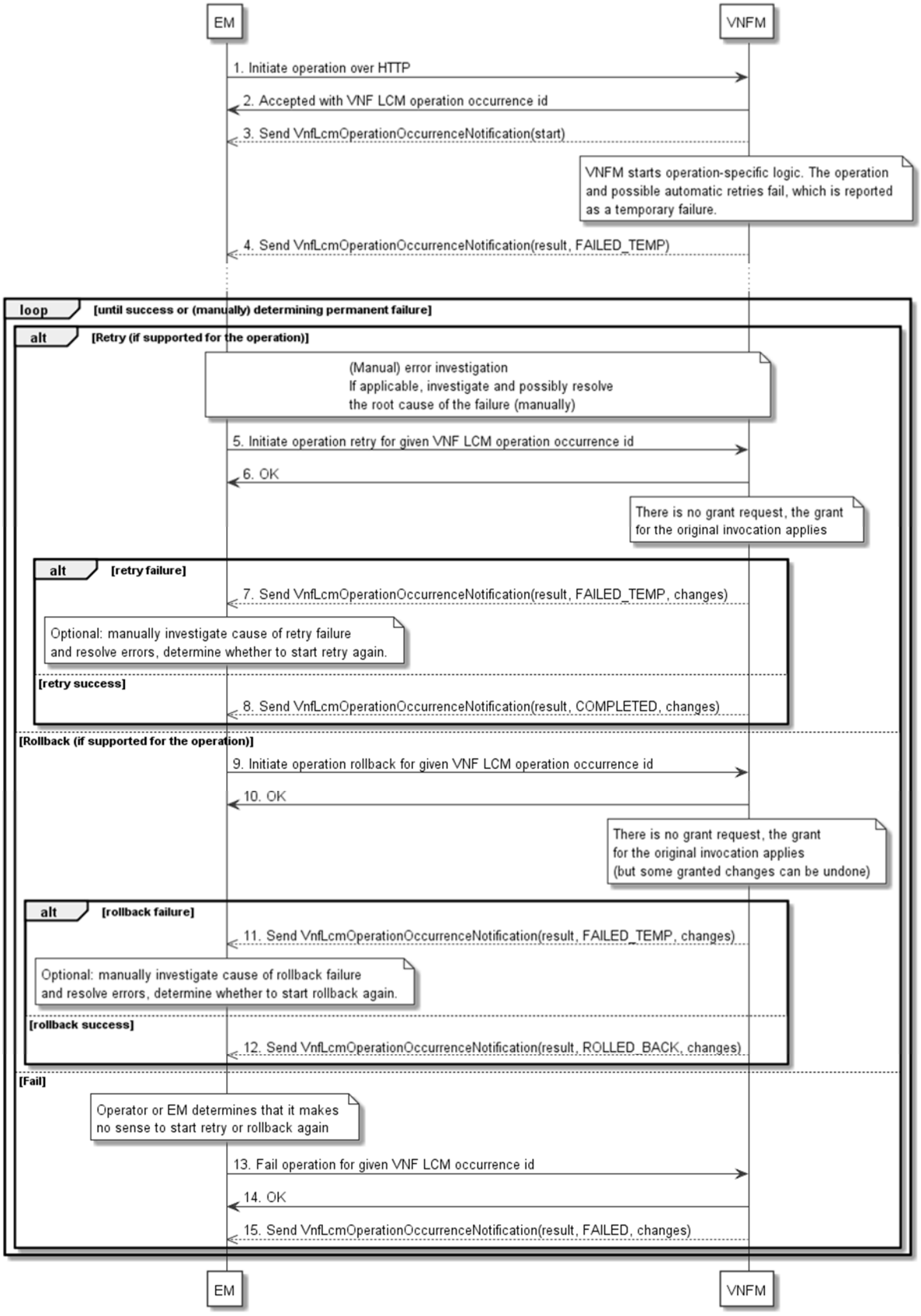

Figure 5.6.3.3-1: Handling failures during the actual execution of a VNF LCM operation

5.6.3.4 LCM operation cancellation

The cancellation of an LCM operation that is in PROCESSING or ROLLING_BACK state is handled like any other error that leads to stopping the execution of the VNF LCM workflow before it can be successfully completed. The VNF 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.

The cancellation of an operation in STARTING state (i.e. until the Grant is received) transits the operation into the ROLLED_BACK state, as no changes to the resources or VNF instance have been performed.

5.7 Handling of security-sensitive attributes

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" which 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.

6 VNF Performance Management interface

6.1 Description

This interface allows providing performance management (measurement results collection and notifications) related to VNFs. Performance information on a given VNF/VNFC instance results from performance information of the virtualised resources that is collected from the VIM and mapped to this VNF/VNFC instance. 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

6.1a API version

For the VNF performance management interface version as specified in the present document, the MAJOR version field shall be 2 and the MINOR version field shall be 1, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [6] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v2".

NOTE: In the present document, there were no changes to the clauses defining the VNF performance management interface that are visible at interface level compared to the previous version of the present document; hence, the MAJOR/MINOR/PATCH version fields are kept the same.

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 [6].

The string "vnfpm" 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 performance management API.

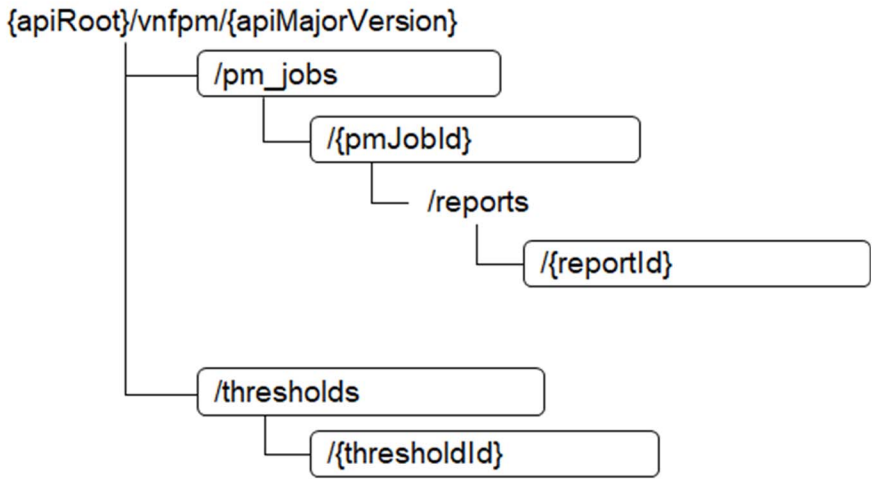

Figure 6.2-1: Resource URI structure of the VNF Performance Management interface

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

The VNFM 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 VNFM shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [6].

Table 6.2-1: Resources and methods overview of the VNF 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.
		GET	See note	Test the notification endpoint.
NOTE: The VNFM shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the EM or VNF. If the EM or VNF supports invoking the POST method on the "PM jobs" or "Thresholds" resource towards the VNFM, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.				

6.3 Sequence diagrams (informative)

6.3.1 Flow of creating a PM job

This clause describes a sequence for creating a performance management job.

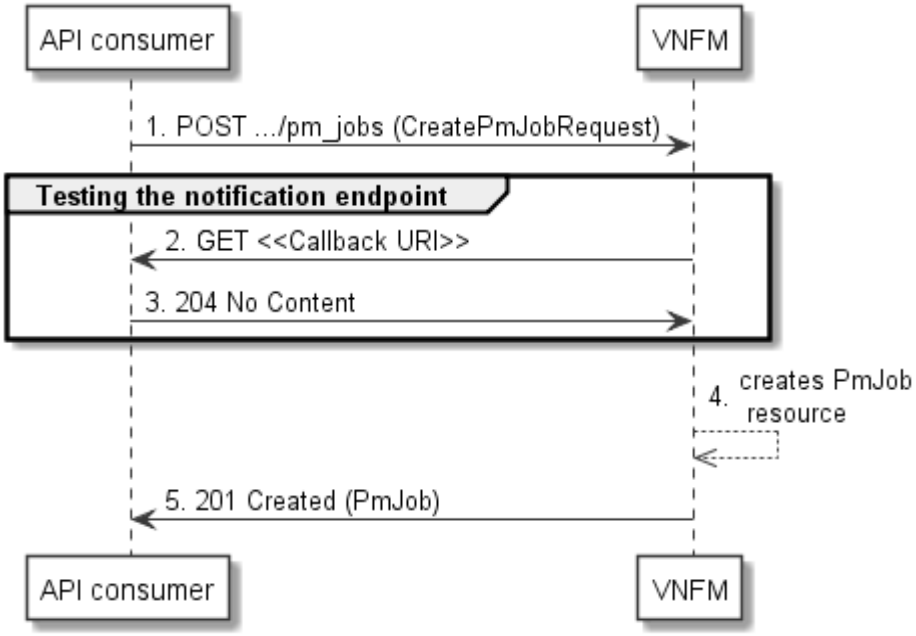

Figure 6.3.1-1: Flow of PM job creation

PM job creation, as illustrated in figure 6.3.1-1, consists of the following steps:

1. If the API consumer 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 API consumer during PM job creation, the VNFM sends a GET request to the notification endpoint URI.
3. The API consumer returns a "204 No Content" response to indicate success.
4. The VNFM creates a PM job instance.
5. The VNFM returns a "201 Created" response to the API consumer, 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.

6.3.1a Flow of updating the callback URI of a PM job

This clause describes a sequence for updating the callback URI in a PM job.

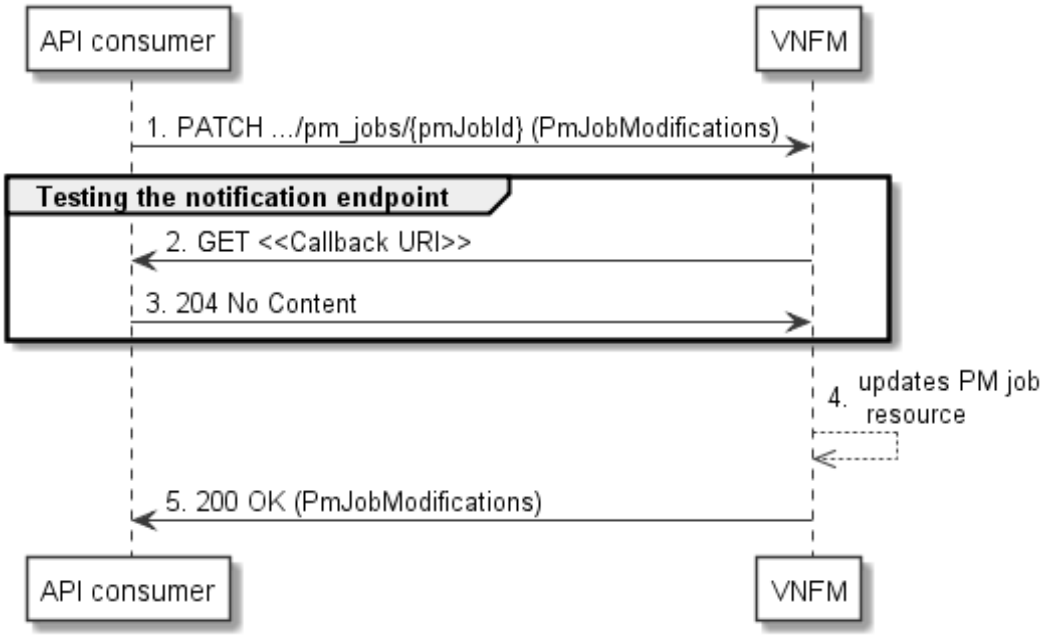

Figure 6.3.1a-1: Flow of PM job callback URI update

PM job callback URI update, as illustrated in figure 6.3.1a-1, consists of the following steps:

1. If the API consumer 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 VNFM sends a GET request to the notification endpoint URI.
3. The API consumer returns a "204 No Content" response to indicate success.
4. The VNFM updates the callback URI of the "Individual PM job" resource.
5. The VNFM returns a "200 OK" response to the API consumer 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.

6.3.2 Flow of querying/reading PM jobs

This clause describes a sequence for querying/reading performance management jobs.

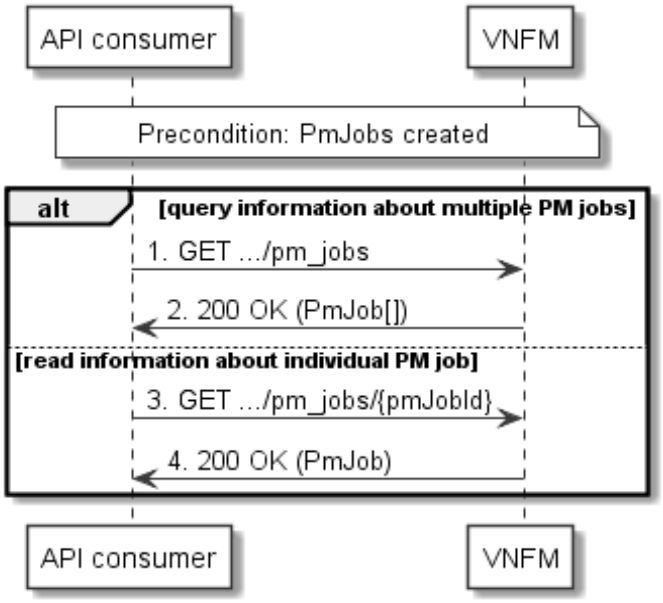

Figure 6.3.2-1: Flow of PM jobs query/read

PM jobs query/read, as illustrated in figure 6.3.2-1, consists of the following steps:

1. If the API consumer intends to query all PM jobs, it sends a GET request to the "PM jobs" resource.
2. The VNFM returns a "200 OK" response to the API consumer, and includes zero or more data structures of type "PmJob" in the payload body.
3. If the API consumer 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 VNFM returns a "200 OK" response to the API consumer, 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.

6.3.3 Flow of deleting a PM job

This clause describes a sequence for deleting a performance management job.

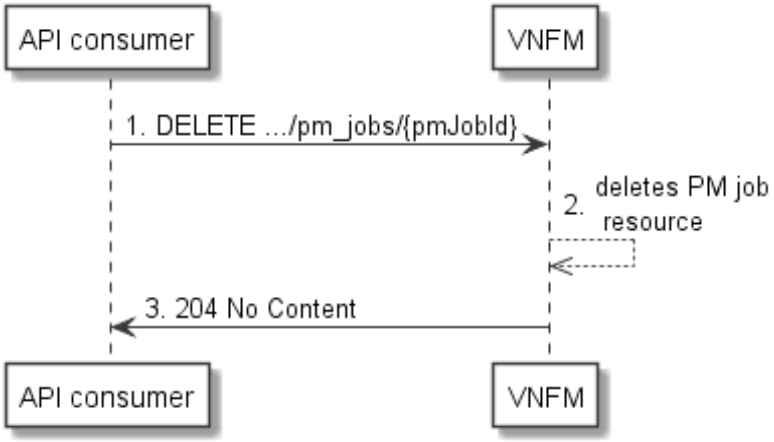

Figure 6.3.3-1: Flow of PM job deletion

PM job deletion, as illustrated in figure 6.3.3-1, consists of the following steps:

1. If the API consumer 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 VNFM deletes the "Individual PM job" resource.
3. The VNFM returns a response with a "204 No Content" response code and an empty payload body to the API consumer.

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

6.3.4 Flow of obtaining performance reports

This clause describes a sequence for obtaining performance reports.

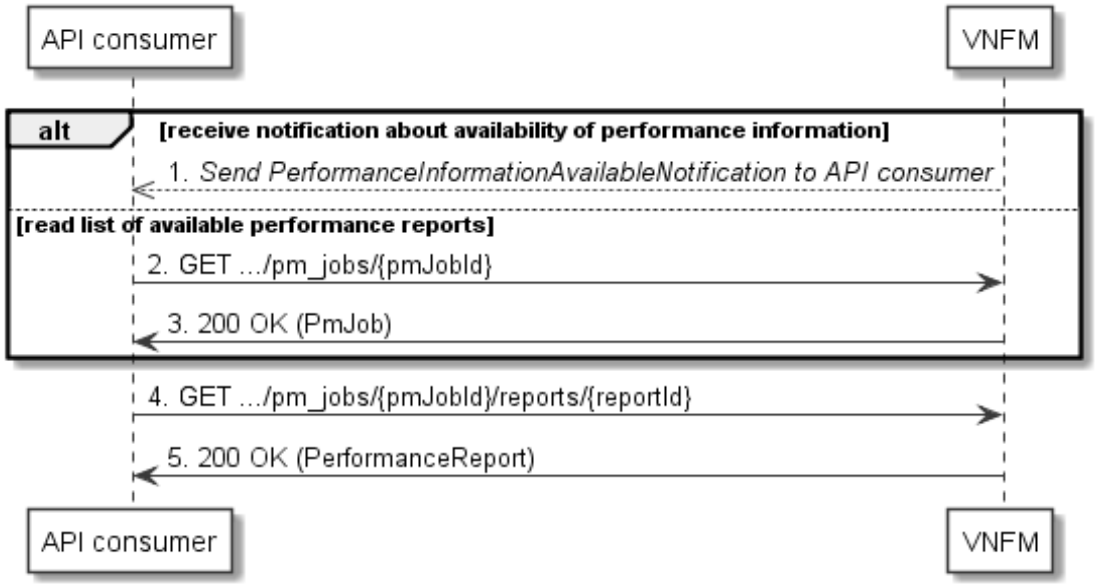

Figure 6.3.4-1: Flow of obtaining performance reports

Obtaining a performance report, as illustrated in figure 6.3.4-1, consists of the following steps:

1. The VNFM sends to the API consumer a PerformanceInformationAvailableNotification (see clause 6.3.9) that indicates the availability of a new performance report, including a link from which the report can be obtained.
2. Alternatively, the API consumer sends a GET request to the "Individual PM job" resource, to obtain a representation of the resource including information about performance reports that are available for this PM job, including their URIs.
3. In that case, the VNFM returns a "200 OK" response to the API consumer, and includes a data structure of type "PmJob" in the payload body.
4. The API consumer sends to the VNFM 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 VNFM returns a "200 OK" response to the API consumer, and includes a data structure of type "PerformanceReport" in the payload body.

6.3.5 Flow of creating a threshold

This clause describes a sequence for creating a performance management threshold.

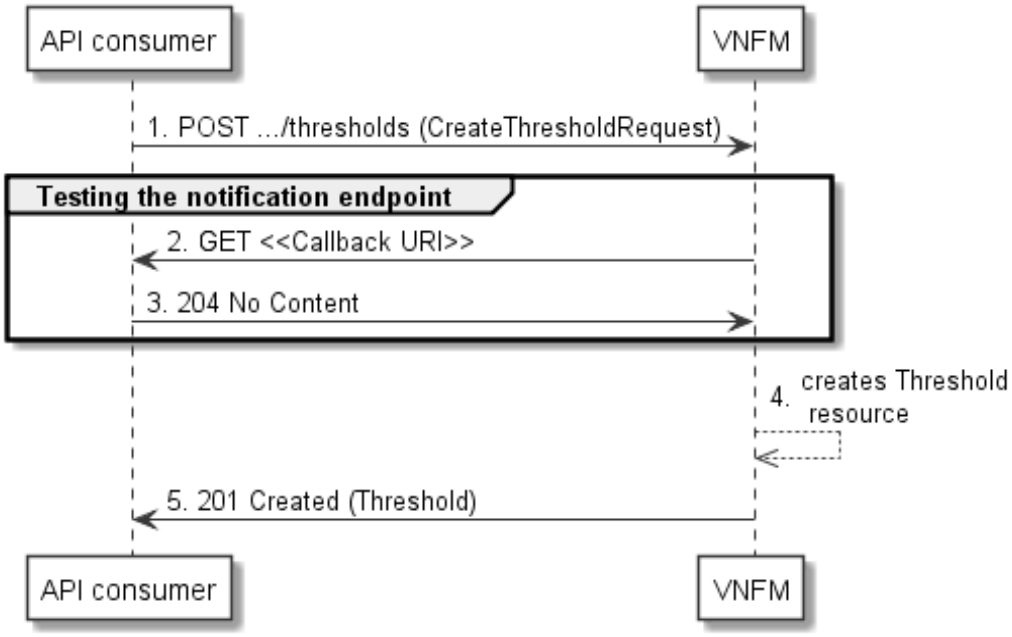

Figure 6.3.5-1: Flow of threshold creation

Threshold creation, as illustrated in figure 6.3.5-1, consists of the following steps:

1. If the API consumer 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 API consumer during threshold creation, the VNFM sends a GET request to the notification endpoint URI.
3. The API consumer returns a "204 No Content" response to indicate success.
4. The VNFM creates a threshold instance.
5. The VNFM returns a "201 Created" response to the API consumer, 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.

6.3.5a Flow of updating the callback URI of a threshold

This clause describes a sequence for updating the callback URI in a performance management threshold.

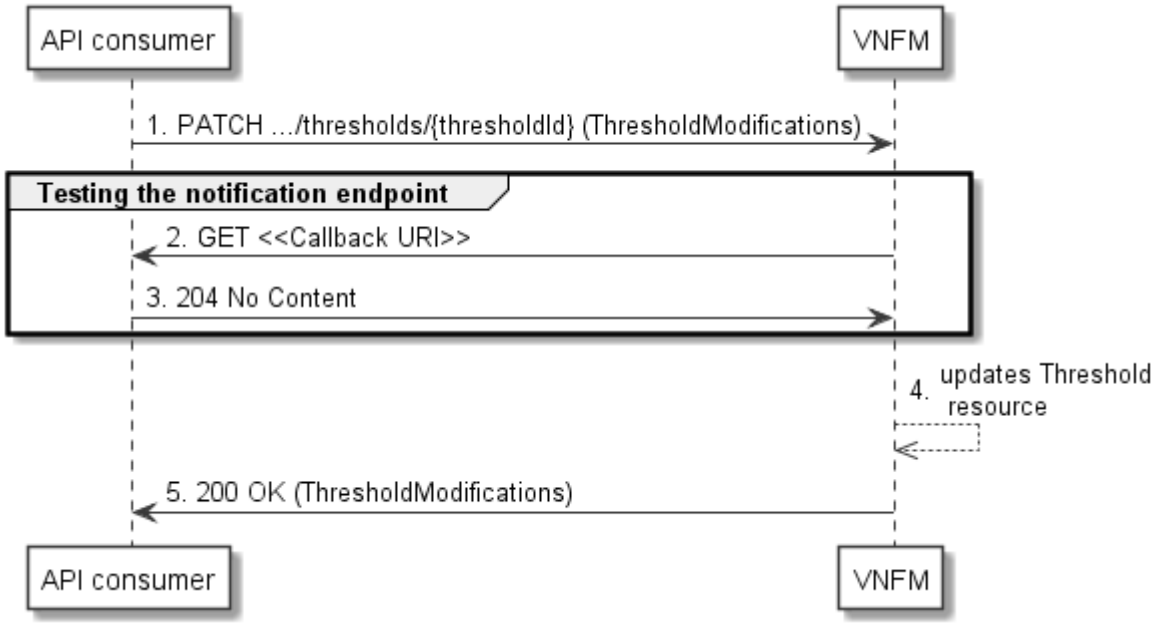

Figure 6.3.5a-1: Flow of threshold callback URI update

Threshold callback URI update, as illustrated in figure 6.3.5a-1, consists of the following steps:

1. If the API consumer 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 VNFM sends a GET request to the notification endpoint URI.
3. The API consumer returns a "204 No Content" response to indicate success.
4. The VNFM updates the callback URI of the "Individual threshold" resource.
5. The VNFM returns a "200 OK" response to the API consumer 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.

6.3.6 Flow of querying/reading thresholds

This clause describes a sequence for querying/reading performance management thresholds.

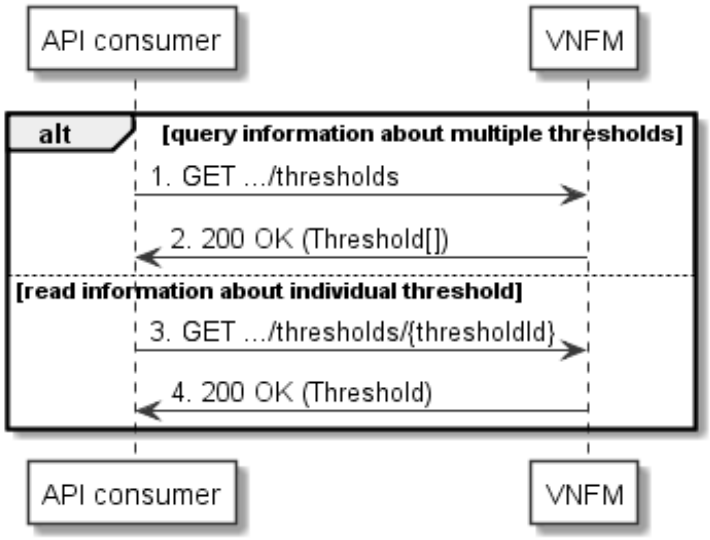

Figure 6.3.6-1: Flow of thresholds query/read

Threshold query/read, as illustrated in figure 6.3.6-1, consists of the following steps:

1. If the API consumer intends to query all thresholds, it sends a GET request to the "Thresholds" resource.
2. The VNFM returns a "200 OK" response to the API consumer, and includes zero or more data structures of type "Threshold" in the payload body.
3. If the API consumer intends to read information about a particular threshold, it sends a GET request to the "Individual threshold" resource, addressed by the appropriate threshold identifier in its resource URI.
4. The VNFM returns a "200 OK" response to the API consumer, 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.

6.3.7 Flow of deleting thresholds

This clause describes a sequence for deleting performance management thresholds.

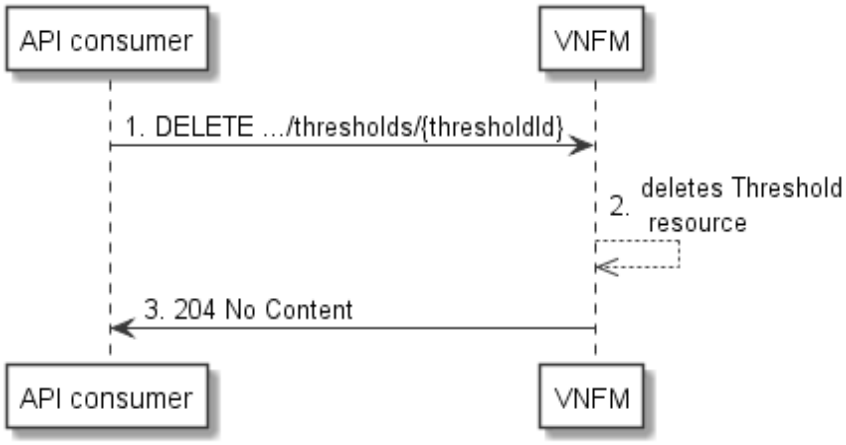

Figure 6.3.7-1: Flow of threshold deletion

Threshold deletion, as illustrated in figure 6.3.7-1, consists of the following steps:

1. If the API consumer 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 VNFM deletes the "Individual threshold" resource.

3. The VNFM returns a "204 No Content" response code to the API consumer. The response body shall be empty.

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

6.3.8 Void

6.3.9 Flow of sending notifications

This clause describes the procedure for sending notifications related to VNF performance management between the API consumer and the VNFM.

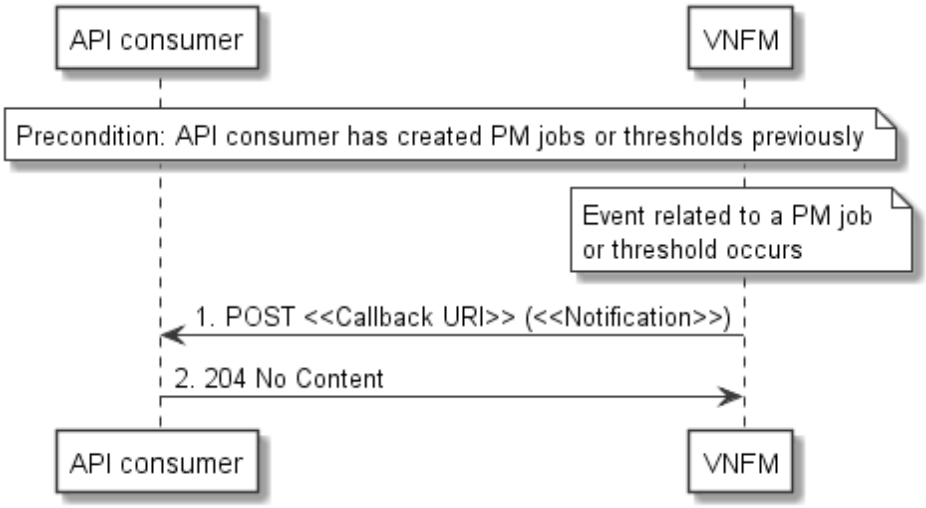

Figure 6.3.9-1: Flow of sending notifications

Precondition: The API consumer has previously created thresholds and/or PM jobs which trigger notifications related to VNF performance management.

The procedure consists of the following steps as illustrated in figure 6.3.9-1:

1. If an event occurs that indicates a threshold crossing or availability of performance information in a PM job, the VNFM generates a notification that includes information about the event, and sends it in the body of a POST request to the URI which the API consumer 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.

2. The API consumer acknowledges the successful delivery of the notification by returning a "204 No Content" response.

Error handling: If the VNFM does not receive the "204 No Content" response from the API consumer, it can retry sending the notification.

6.4 Resources

6.4.1 Introduction

This clause defines all the resources and methods provided by the performance management interface.

6.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [6] are part of the VNF performance management interface.

6.4.2 Resource: PM jobs

6.4.2.1 Description

This resource represents PM jobs. The API consumer can use this resource to create and query PM jobs.

6.4.2.2 Resource definition

The resource URI is:

{apiRoot}/vnfpm/{apiMajorVersion}/pm_jobs

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 [6].
apiMajorVersion	See clause 6.1a.

6.4.2.3 Resource methods

6.4.2.3.1 POST

The POST method creates a PM job.

This method shall follow the provisions specified in 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 PM job" resource as defined in clause 6.4.3 shall have been created.

Table 6.4.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.2.3.1-2: 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	Shall be returned when the PM job has been created successfully. The response body shall contain a representation of the created "Individual PM job" resource, as defined in clause 6.5.2.7. The HTTP response shall include a "Location" HTTP header that points to the created "Individual PM job" resource.
	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 [6], 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 VNFM has tested the Notification endpoint as described in clause 6.4.9.3.2 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 [6]	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 [6] may be returned.

6.4.2.3.2 GET

The API consumer can use this method to retrieve information about PM jobs.

This method shall follow the provisions specified in 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	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [6]. The VNFM shall support receiving this parameter as part of the URI query string. The API consumer may supply this parameter. All attribute names that appear in the PmJob and in data types referenced from it shall be supported by the VNFM 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 [6] for details. The VNFM 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 [6] for details. The VNFM 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 [6] for details. The VNFM 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 [6] for details. The VNFM 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 VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource.

Table 6.4.2.3.2-2: 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 6.5.2.7. 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 [6], respectively. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] 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 [6].
	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. In the returned ProblemDetails structure, 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 VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [6].

	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

6.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.3 Resource: Individual PM job

6.4.3.1 Description

This resource represents an individual PM job. The API consumer can use this resource to delete and read the underlying PM job.

6.4.3.2 Resource definition

The resource URI is:

{apiRoot}/vnfpm/{apiMajorVersion}/pm_jobs/{pmJobId}

This resource shall support the resource URI variables 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 [6].
apiMajorVersion	See clause 6.1a.
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.	

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 VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.3.3.2 GET

The API consumer can use this method for reading an individual PM job.

This method shall follow the provisions specified in 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	Description
none supported		

Table 6.4.3.3.2-2: 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 6.5.2.7.
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

6.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.3.3.4 PATCH

This method allows to modify an "individual PM job" resource.

This method shall follow the provisions specified in tables 6.4.3.3.4-1 and 6.4.3.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.3.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.3.3.4-2: Details of the PATCH request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	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 [3].
Response body	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 [6], 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 VNFM has tested the Notification endpoint as described in clause 6.4.9.3.2 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 [6]	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 [6] may be returned.

6.4.3.3.5 DELETE

This method terminates an individual PM job.

This method shall follow the provisions specified in 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 PM job" resource shall not exist any longer.

Table 6.4.3.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.3.3.5-2: 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 [6]	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 [6] may be returned.

6.4.4 Resource: Individual performance report

6.4.4.1 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 6.5.2.5) 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.

6.4.4.2 Resource definition

The resource URI is:

{apiRoot}/vnfpm/{apiMajorVersion}/pm_jobs/{pmJobId}/reports/{reportId}

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 [6].
apiMajorVersion	See clause 6.1a.
pmJobId	Identifier of the PM job.
reportId	Identifier of the performance report.

6.4.4.3 Resource methods

6.4.4.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.4.3.2 GET

The API consumer can use this method for reading an individual performance report.

This method shall follow the provisions specified in tables 6.4.4.3.2-1 and 6.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.4.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.4.3.2-2: 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 6.5.2.10.
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

6.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.5 Resource: Thresholds

6.4.5.1 Description

This resource represents thresholds. The API consumer can use this resource to create and query thresholds.

6.4.5.2 Resource definition

The resource URI is:

{apiRoot}/vnfpm/{apiMajorVersion}/thresholds

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 [6].
apiMajorVersion	See clause 6.1a.

6.4.5.3 Resource methods

6.4.5.3.1 POST

The POST method can be used by the API consumer to create a threshold.

This method shall follow the provisions specified in 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.

As the result of successfully executing this method, a new "Individual threshold" resource as defined in clause 6.4.6 shall have been created.

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

	Data type	Cardinality	Response Codes	Description
Request body	CreateThresholdRequest	1		Request parameters to create a new "Individual threshold" resource.
Response body	Threshold	1	201 Created	Shall be returned when a threshold has been created successfully. The response body shall contain a representation of the created "Individual threshold" resource, as defined in clause 6.5.2.9. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created resource.
	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 [6], 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 VNFM has tested the Notification endpoint as described in clause 6.4.9.3.2 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 [6]	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 [6] may be returned.

6.4.5.3.2 GET

The API consumer can use this method to query information about thresholds.

This method shall follow the provisions specified in tables 6.4.5.3.2-1 and 6.4.5.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.5.3.2-1: 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 [6]. The VNFM shall support receiving this parameter as part of the URI query string. The API consumer 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 VNFM in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] 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 6.4.5.3.2-2: 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 [6]. The response body shall contain in an array the representations of zero or more thresholds, as defined in clause 6.5.2.9. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] 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 [6].
	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 VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [6].
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

6.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.6 Resource: Individual threshold

6.4.6.1 Description

This resource represents an individual threshold.

6.4.6.2 Resource definition

The resource URI is:

{apiRoot}/vnfpm/{apiMajorVersion}/thresholds/{thresholdId}

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 [6].
apiMajorVersion	See clause 6.1a.
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.	

6.4.6.3 Resource methods

6.4.6.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.6.3.2 GET

The API consumer can use this method for reading an individual threshold.

This method shall follow the provisions specified in tables 6.4.6.3.2-1 and 6.4.6.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.6.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.6.3.2-2: 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 6.5.2.9.
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

6.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.6.3.4 PATCH

This method allows to modify an "Individual threshold" resource.

This method shall follow the provisions specified in tables 6.4.6.3.4-1 and 6.4.6.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.6.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.6.3.4-2: Details of the PATCH request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	ThresholdModifications	1		Parameters for the threshold modification. The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [3].
Response body	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	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 [6], 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 VNFM has tested the Notification endpoint as described in clause 6.4.9.3.2 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 [6]	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 [6] may be returned.

6.4.6.3.5 DELETE

This method allows to delete a threshold.

This method shall follow the provisions specified in tables 6.4.6.3.5-1 and 6.4.6.3.5-2 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 6.4.6.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.6.3.5-2: 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 threshold has been deleted successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

6.4.7 Void

6.4.8 Void

6.4.9 Resource: Notification endpoint

6.4.9.1 Description

This resource represents a notification endpoint for VNF 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.

6.4.9.2 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 6.4.9.2-1.

Table 6.4.9.2-1: Resource URI variables for this resource

Name	Definition
none supported	

6.4.9.3 Resource methods

6.4.9.3.1 POST

The POST method delivers a notification regarding a performance management event from 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 tables 6.4.9.3.1-1 and 6.4.9.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.9.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.9.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	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 [6]	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 [6] may be returned.

6.4.9.3.2 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 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
none supported		

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**
	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 [6]	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 [6] may be returned.

6.4.9.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.9.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.4.9.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

6.5 Data Model

6.5.1 Introduction

This clause defines the request and response data structures of the VNF 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.

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 Void

6.5.2.3 Void

6.5.2.4 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 6.5.2.4-1.

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 VNFM when a threshold has been crossed.

Table 6.5.2.4-1: 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 7.2 of ETSI GS NFV-IFA 027 [5].
objectInstanceId	Identifier	1	Identifier of the measured object instance as per clause 6.2 of ETSI GS NFV-IFA 027 [5].
subObjectInstanceId	IdentifierInVnf	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 [5] 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 7.2 of ETSI GS NFV-IFA 027 [5].
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 7.2 of ETSI GS NFV-IFA 027 [5].
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 7.2 of ETSI GS NFV-IFA 027 [5].
_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.			

6.5.2.5 Type: PerformanceInformationAvailableNotification

This notification informs the receiver that performance information is available. It shall comply with the provisions defined in table 6.5.2.5-1.

The notification shall be triggered by the VNFM 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 6.5.3.3.

Table 6.5.2.5-1: 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 7.2 of ETSI GS NFV-IFA 027 [5].
objectInstanceId	Identifier	1	Identifier of the measured object instance as per clause 6.2 of ETSI GS NFV-IFA 027 [5].
subObjectInstanceIds	IdentifierInVnf	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 [5] 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 6.5.2.10) can be obtained. This link should point to an "Individual performance report" resource as defined in clause 6.4.4.

6.5.2.6 Type: CreatePmJobRequest

This type represents a request to create a PM job. It shall comply with the provisions defined in table 6.5.2.6-1.

Table 6.5.2.6-1: 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 7.2 of ETSI GS NFV-IFA 027 [5].
objectInstanceIds	Identifier	1..N	Identifiers of the measured object instances for which performance information is requested to be collected.

Attribute name	Data type	Cardinality	Description
subObjectInstanceIds	IdentifierInVnf	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 [5] for the related measured object type If this attribute is present, the cardinality of the "objectInstanceIds" 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 [5] 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 [6]. This attribute shall only be present if the API consumer requires authorization of notifications.

6.5.2.7 Type: PmJob

This type represents a PM job. It shall comply with the provisions defined in table 6.5.2.7-1.

Table 6.5.2.7-1: 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 7.2 of ETSI GS NFV-IFA 027 [5].
objectInstanceIds	Identifier	1..N	Identifiers of the measured object instances for which performance information is collected.
subObjectInstanceIds	IdentifierInVnf	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 [5] for the related measured object type If this attribute is present, the cardinality of the "objectInstanceIds" 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 [5] 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	UnsigendInt	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.

Attribute name	Data type	Cardinality	Description
>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.

6.5.2.8 Type: CreateThresholdRequest

This type represents a request to create a threshold. It shall comply with the provisions defined in table 6.5.2.8-1.

Table 6.5.2.8-1: 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 7.2 of ETSI GS NFV-IFA 027 [5].
objectInstanceId	Identifier	1	Identifier of the measured object instance associated with this threshold.
subObjectInstanceIds	IdentifierInVnf	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 [5] 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 [5] 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 [6]. This attribute shall only be present if the API consumer requires authorization of notifications.

6.5.2.9 Type: Threshold

This type represents a threshold. It shall comply with the provisions defined in table 6.5.2.9-1.

Table 6.5.2.9-1: 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 7.2 of ETSI GS NFV-IFA 027 [5].
objectInstanceId	Identifier	1	Identifier of the measured object (i.e. VNF instance) associated with the threshold.
subObjectInstanceIds	IdentifierInVnf	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 [5] 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 [5] 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.
_links	Structure (inlined)	1	Links for this resource.

Attribute name	Data type	Cardinality	Description
>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.

6.5.2.10 Type: PerformanceReport

This type defines the format of a performance report provided by the VNFM to the API consumer as a result of collecting performance information as part of a PM job. The type shall comply with the provisions defined in table 6.5.2.10-1.

Table 6.5.2.10-1: 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. VNF 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 7.2 of ETSI GS NFV-IFA 027 [5].
>objectInstanceId	Identifier	1	Identifier of the measured object instance for which the performance metric is reported.
>subObjectInstanceId	IdentifierInVnf	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 [5] 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 7.2 of ETSI GS NFV-IFA 027 [5].
>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 7.2 of ETSI GS NFV-IFA 027 [5].
>>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 7.2 of ETSI GS NFV-IFA 027 [5].

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: VnfInstanceXYZ
Sub-object: VnfcInstance1
Measurement: vCPU_utilization
Sub-counters: vCPU utilization of each of the vCPUs of VnfcInstance1 (vCPU_utilization.vCPU1, vCPU_utilization.vCPU2, etc.).

6.5.2.11 Type: ThresholdModifications

This type represents modifications to a threshold. It shall comply with the provisions defined in table 6.5.2.11-1.

Table 6.5.2.11-1: 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 [3]). 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.			

6.5.2.12 Type: PmJobModifications

This type represents modifications to a PM job. It shall comply with the provisions defined in table 6.5.2.12-1.

Table 6.5.2.12-1: 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 [3]). 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.			

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 Void

6.5.3.3 Type: PmJobCriteria

This type represents collection criteria for PM jobs. It shall comply with the provisions defined in table 6.5.3.3-1.

Table 6.5.3.3-1: 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 7.2 of ETSI GS NFV-IFA 027 [5]). At least one of the two attributes (performance metric or group) shall be present.

Attribute name	Data type	Cardinality	Description
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 7.2 of ETSI GS NFV-IFA 027 [5]). 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.
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.			

6.5.3.4 Type: ThresholdCriteria

This type represents criteria that define a threshold. It shall comply with the provisions defined in table 6.5.3.4-1.

Table 6.5.3.4-1: 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 [5].
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).			

6.5.4 Referenced simple data types and enumerations

6.5.4.1 Introduction

This clause defines simple data types and enumerations 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.

6.5.4.3 Enumeration: CrossingDirectionType

The enumeration CrossingDirectionType shall comply with the provisions defined in table 6.5.4.3-1.

Table 6.5.4.3-1: Enumeration CrossingDirectionType

Enumeration value	Description
UP	The threshold was crossed in upward direction.
DOWN	The threshold was crossed in downward direction.

7 VNF Fault Management interface

7.1 Description

This interface allows the EM/VNF to subscribe to notifications regarding VNF alarms provided by the VNFM, and API version information retrieval.

Virtualised resource alarms collected by the VNFM are filtered, correlated and modified by the VNFM and mapped to the corresponding VNF instance, resulting in alarms on that VNF instance which contain information on the VNFC(s) affected by the fault.

Reasons for creating alarms include the following:

- faults detected by the VNFM;
- faults generated by the VNFM due to changes in the state of virtualised resources used by the VNFs and their constituent VNFC instances managed by the VNFM, including changes in the state of the virtualised resources due to upcoming NFVI operation and maintenance; and
- faults generated by the VIM on virtualised resources used by the VNFs and their constituent VNFC instances managed by the VNFM.

NOTE 1: The present document specifies values of the attribute "faultType" and other attributes only for the case of faults generated by the VNFM because of changes in the state of virtualised resources due to upcoming NFVI operation and maintenance. The other cases listed above are not specified in the present document.

NOTE 2: The activities of NFVI operation and maintenance have impact on VNF instances running on top of those NFVI resources to be changed. Annex C of ETSI GS NFV-IFA 008 [1] provides NFVI operation and maintenance policies to minimize the impact on the service continuity of the VNF instances.

The operations provided through this interface are:

- Get Alarm List
- Acknowledge Alarm
- Subscribe

- Query Subscription Information
- Terminate Subscription
- Notify
- Escalate perceived severity

7.1a API version

For the VNF fault management interface version as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 4, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [6] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v1".

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 [6]. The string "vnffm" 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 VNF fault management interface.

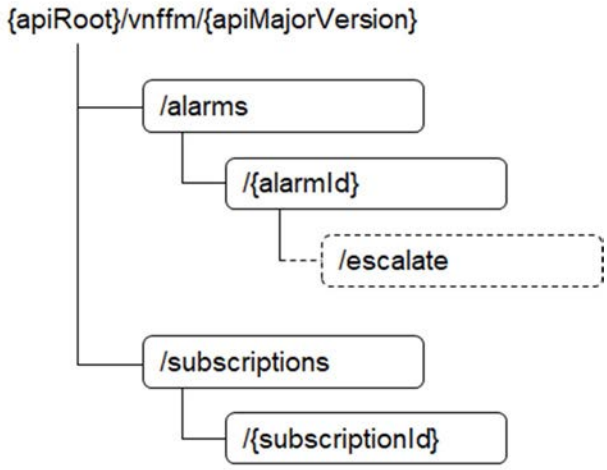

Figure 7.2-1: Resource URI structure of the VNF Fault Management interface

Table 7.2-1 lists the individual resources defined, and the applicable HTTP methods.

The VNFM 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 VNFM shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [6].

Table 7.2-1: Resources and methods overview of the VNF Fault Management interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
Alarms	/alarms	GET	M	Query alarms related to VNF instances.
Individual alarm	/alarms/{alarmId}	GET	M	Read individual alarm.
		PATCH	M	Acknowledge individual alarm.
Escalate perceived severity task	/alarms/{alarmId}/escalate	POST	M	Escalate the API consumer's view of perceived severity.
Subscriptions	/subscriptions	POST	M	Subscribe to VNF alarms.
		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 VNF alarms.
		GET	See note	Test the notification endpoint.
NOTE: The VNFM shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the EM or VNF. If the EM or VNF supports invoking the POST method on the "Subscription" resource towards the VNFM, 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 the Get Alarm List operation

This clause describes a sequence flow for querying one or multiple alarms.

NOTE: The API consumer can be either EM or VNF depending on the operations.

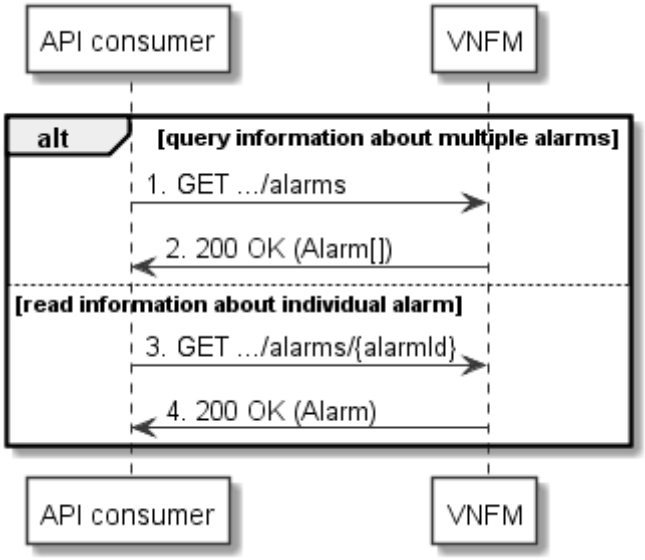

Figure 7.3.1-1: Flow of alarm query/read

Alarm query, as illustrated in figure 7.3.1-1, consists of the following steps:

1. If the API consumer intends to query all alarms, it sends a GET request to the "Alarms" resource.
2. The VNFM returns a "200 OK" response to the API consumer, and includes zero or more data structures of type "Alarm" in the payload body.
3. If the API consumer 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 VNFM returns a "200 OK" response to the API consumer, 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.

7.3.2 Escalate perceived severity task flow

This clause describes the procedure to escalate the API consumer's view of the perceived severity for an alarm to the VNFM.

NOTE 1: The API consumer can be either the EM or VNF.

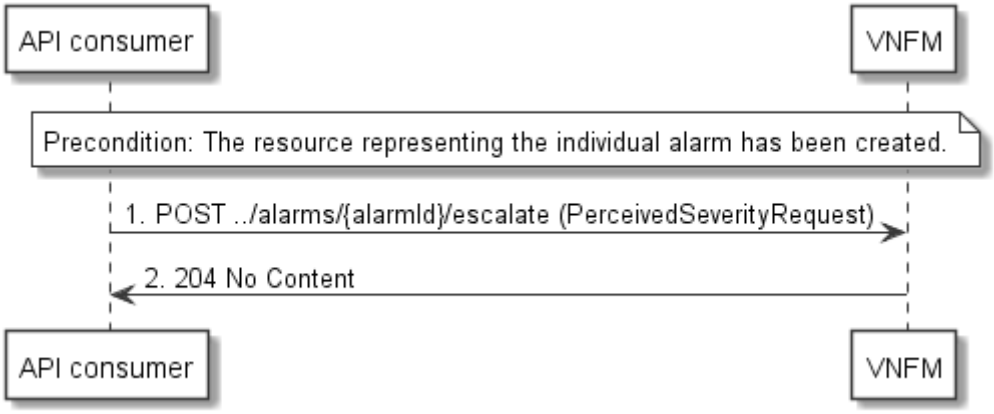

Figure 7.3.2-1: Escalate perceived severity task flow

Precondition: The resource representing the individual alarm has been created.

The escalate perceived severity task flow, as illustrated in figure 7.3.2-1, consists of the following steps:

1. The API consumer sends a POST request to the "Escalate Perceived Severity" task resource of an individual alarm, which is identified by the "alarmId" in the resource URI.

2. The VNFM returns a "204 No Content" response to the API consumer.

NOTE 2: If the value of the perceived severity is changed after this operation is finished, the VNFM will send an AlarmNotification to each subscribed API consumer.

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

7.3.3 Flow of acknowledging alarm

This clause describes the procedure to acknowledge an individual alarm.

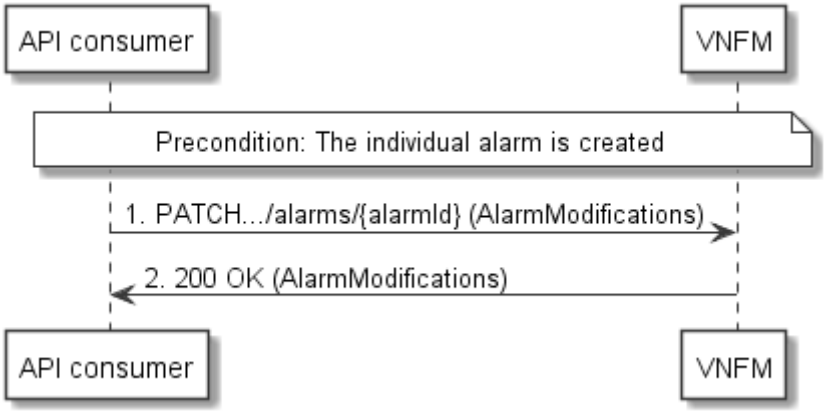

Figure 7.3.3-1: Flow of acknowledging alarm

Precondition: The resource representing the individual alarm has been created.

Acknowledge alarm, as illustrated in figure 7.3.3-1, consists of the following steps:

1. The API consumer sends a PATCH request to the individual alarm.
2. The VNFM returns a "200 OK" response to the API consumer, 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.

7.3.4 Flow of managing subscriptions

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

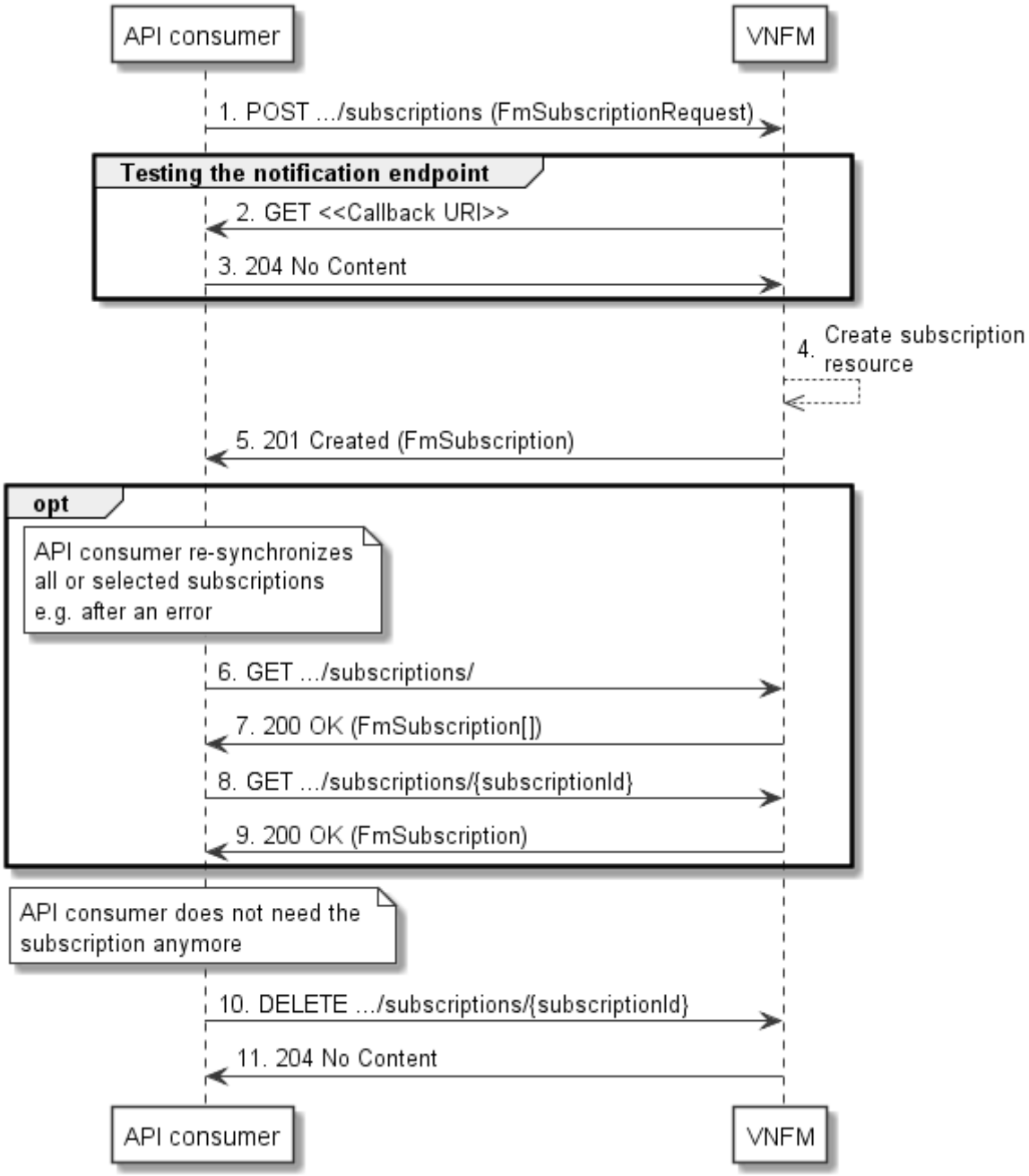

Figure 7.3.4-1: Flow of managing subscriptions

The procedure consists of the following steps as illustrated in figure 7.3.4-1:

1. The API consumer 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 VNFM will subsequently send notifications about events that match the filter.

2. To test the notification endpoint that has been registered by the API consumer as part of the subscription, the VNFM sends a GET request to the notification endpoint URI.

3. The API consumer returns a "204 No Content" response to indicate success.

4. The VNFM creates a new subscription for notifications related to VNF fault management, and a resource that represents this subscription.

5. The VNFM returns a "201 Created" response containing a data structure of type "FmSubscription", representing the "Individual subscription" resource just created by the VNFM, 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 API consumer can query information about its subscriptions by sending a GET request to the "Subscriptions" resource.

7. In that case, the VNFM returns a "200 OK" response that contains the list of representations of all existing subscriptions that were created by the API consumer.

8. If desired, e.g. to recover from an error situation, the API consumer can read information about a particular subscription by sending a GET request to the resource representing that individual subscription.

9. In that case, the VNFM returns a "200 OK" response that contains a representation of that individual subscription.

10. When the API consumer 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 VNFM acknowledges the successful termination of the subscription by returning a "204 No Content" response.

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

7.3.5 Flow of sending notifications

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

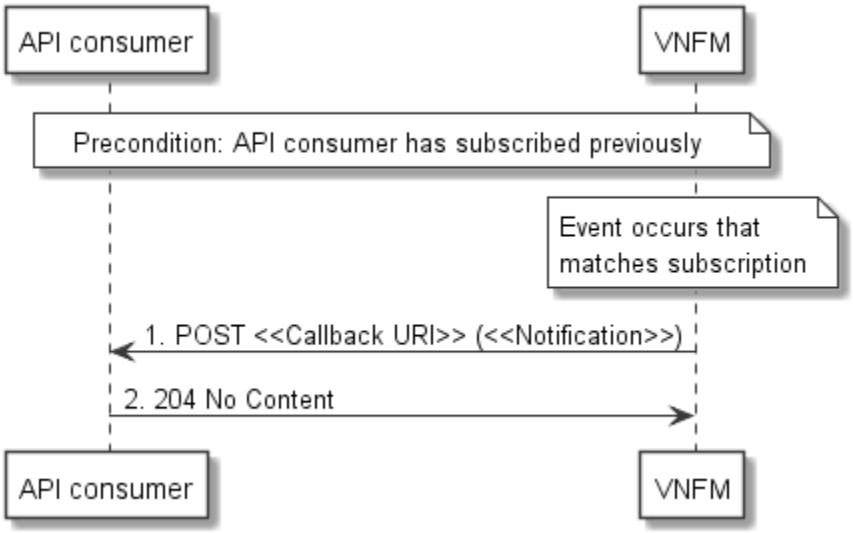

Figure 7.3.5-1: Flow of sending notifications

Precondition: The API consumer has subscribed previously for notifications related to VNF fault management.

The procedure consists of the following steps as illustrated in figure 7.3.5-1:

1. If an event occurs that matches the filtering criteria which are part of the subscription, the VNFM generates a notification that includes information about the event, and sends it in the body of a POST request to the URI which the API consumer 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 7.5.2.5, 7.5.2.6 and 7.5.2.7).

2. The API consumer acknowledges the successful delivery of the notification by returning a "204 No Content" response.

Error handling: If the VNFM does not receive the "204 No Content" response from the API consumer, it can retry sending the notification.

7.4 Resources

7.4.1 Introduction

This clause defines all the resources and methods provided by the VNF fault 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 [6] are part of the VNF fault management interface.

7.4.2 Resource: Alarms

7.4.2.1 Description

This resource represents a list of alarms related to VNF instances.

7.4.2.2 Resource definition

The resource URI is:

{apiRoot}/vnffm/{apiMajorVersion}/alarms

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 [6].
apiMajorVersion	See clause 7.1a.

7.4.2.3 Resource methods

7.4.2.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.2.3.2 GET

The API consumer can use this method to retrieve information about the alarm list.

This method shall follow the provisions specified in 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 [6]. The VNFM shall support receiving this parameter as part of the URI query string. The EM may supply this parameter. The VNF may supply its instance Id as an attribute filter. The following attribute names shall be supported in the filter expression: id, managedObjectId, vnfcInstanceIds, rootCauseFaultyResource.faultyResourceType, eventType, perceivedSeverity, probableCause. If the vnfcInstanceIds parameter is provided, exactly one value for the managedObjectId attribute shall be provided.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] 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 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**
	Alarm	0..N	200 OK	Shall be returned when information about zero or more alarms has been queried successfully. The response body shall contain in an array the representations of zero or more alarms as defined in clause 7.5.2.4. 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 [6]. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] 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 [6].
	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 VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [6].
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

7.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.3 Resource: Individual alarm

7.4.3.1 Description

This resource represents an individual alarm.

7.4.3.2 Resource definition

The resource URI is:

{apiRoot}/vnffm/{apiMajorVersion}/alarms/{alarmId}

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 [6].
apiMajorVersion	See clause 7.1a.
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.	

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 VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.3.3.2 GET

The API consumer can use this method to read an individual alarm.

This method shall follow the provisions specified in 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**
	Alarm	1	200	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 [6]	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 [6] may be returned.

7.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.3.3.4 PATCH

This method modifies an individual alarm resource.

This method shall follow the provisions specified in 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

	Data type	Cardinality	Response Codes	Description
Request body	AlarmModifications	1		The parameter for the alarm modification, as defined in clause 7.5.2.9. The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [3].
Response body	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 7.5.2.4).
	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 [6]	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 [6] may be returned.

7.4.3.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.4 Resource: Escalate Perceived Severity task

7.4.4.1 Description

This task resource represents the "Escalate Perceived Severity" operation. The API consumer can use this resource to escalate the perceived severity of an alarm with the VNFM. This operation does not directly modify the value of perceived severity attribute in the alarm information element within the VNFM. VNFM implementation (e.g. controlled by operator configuration) will determine how it should act upon receipt of the requested change in perceived severity. Some requests from the EM/VNF may be respected and applied directly by the VNFM, while others may be ignored by the VNFM.

7.4.4.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/alarms/{alarmId}/escalate

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 [6].
apiMajorVersion	See clause 7.1a.
alarmId	Identifier of the alarm.

7.4.4.3 Resource Methods

7.4.4.3.1 POST

The POST method enables the API consumer to escalate the perceived severity of an alarm that is represented by an individual alarm resource.

This method shall follow the provisions specified in tables 7.4.4.3.1-1 and 7.4.4.3.1-2 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 7.4.5 shall have been created. This method shall not trigger any notification.

Table 7.4.4.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.4.3.1-2: Details of the POST request/response on this resource

Request body	**Data type**	**Cardinality**	**Description**	
	PerceivedSeverityRequest	1	The proposed "escalated perceived severity" value, as defined in clause 7.5.2.7.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a	1	204 No Content	Shall be returned when the VNFM has received the proposed "escalated perceived severity" value successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [6]	4xx/5xx	Any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6] may be returned.

7.4.4.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.5 Resource: Subscriptions

7.4.5.1 Description

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

7.4.5.2 Resource definition

The resource URI is:

{apiRoot}/vnffm/{apiMajorVersion}/subscriptions

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 [6].
apiMajorVersion	See clause 7.1a.

7.4.5.3 Resource methods

7.4.5.3.1 POST

The POST method creates a new subscription.

This method shall follow the provisions specified in 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 subscription" resource as defined in clause 7.4.5 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 EM or VNF, and might make sense only in very rare use cases. Consequently, the VNFM 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 7.4.5.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Remarks
none supported		

Table 7.4.5.3.1-2: 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 7.5.2.2.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	FmSubscription	1	201 Created	Shall be returned when the subscription has been created successfully. The response body shall contain a representation of the created "Individual subscription" resource. The HTTP response shall include a "Location:" HTTP header that points to the created "Individual subscription" resource.
	n/a		303 See Other	Shall be returned when a subscription with the same callback URI and the same filter already exists and the policy of the VNFM is to not create redundant subscriptions. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource. The response body shall be empty.
	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 [6], 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 VNFM has tested the Notification endpoint as described in clause 7.4.7.3.2 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 [6]	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 [6] may be returned.

7.4.5.3.2 GET

The API consumer can use this method to retrieve the list of active subscriptions for VNF alarms subscribed by the API consumer. It can be used e.g. for resynchronization after error situations.

This method shall follow the provisions specified in 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	Remarks
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [6]. The VNFM shall support receiving this parameter as part of the URI query string. The EM may supply this parameter. The VNF may supply its instance Id as an attribute filter. All attribute names that appear in the FmSubscription and in data types referenced from it shall be supported by the VNFM in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource.

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**
	FmSubscription	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 FM subscriptions as defined in clause 7.5.2.3. 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 [6]. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] 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 [6].
	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 VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [6].
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

7.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.6 Resource: Individual subscription

7.4.6.1 Description

This resource represents an individual subscription for VNF alarms. The API consumer can use this resource to read and to terminate a subscription to notifications related to VNF fault management.

7.4.6.2 Resource definition

The resource URI is:

{apiRoot}/vnffm/{apiMajorVersion}/subscriptions/{subscriptionId}

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 [6].
apiMajorVersion	See clause 7.1a.
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.	

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 VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.6.3.2 GET

The API consumer can use this method for reading an individual subscription for VNF alarms subscribed by the API consumer.

This method shall follow the provisions specified in 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**
	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 [6]	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 [6] may be returned.

7.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.6.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.6.3.5 DELETE

This method terminates an individual subscription.

This method shall follow the provisions specified in 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 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 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	Shall be returned when the "Individual subscription" resource has been deleted successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

7.4.7 Resource: Notification endpoint

7.4.7.1 Description

This resource represents a notification endpoint for VNF alarms. The API producer can use this resource to send notifications related to VNF alarms or about a rebuilt alarm list to a subscribed API consumer, which has provided the URI of this resource during the subscription process.

7.4.7.2 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 7.4.7.2-1.

Table 7.4.7.2-1: Resource URI variables for this resource

Name	Definition
n/a	

7.4.7.3 Resource methods

7.4.7.3.1 POST

The POST method notifies a VNF alarm 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 tables 7.4.7.3.1-1 and 7.4.7.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.7.3.1-1: 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 7.4.7.3.1-2.

Table 7.4.7.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	AlarmNotification	1		Information of a VNF alarm.
	AlarmClearedNotification	1		Information of the clearance of a VNF alarm.
	AlarmListRebuiltNotification	1		Information that the alarm list has been rebuilt by the VNFM.
Response body	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 [6]	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 [6] may be returned.

7.4.7.3.2 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 tables 7.4.7.3.2-1 and 7.4.7.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.7.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.7.3.2-2: Details of the GET request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	n/a			
Response body	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 [6]	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 [6] may be returned.

7.4.7.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.7.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.4.7.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

7.5 Data Model

7.5.1 Introduction

This clause defines the request and response data structures of the VNF 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.

7.5.2 Resource and notification data types

7.5.2.1 Introduction

This clause defines the data structures to be used in the resource representations and notifications for the VNF fault management interface.

7.5.2.2 Type: FmSubscriptionRequest

This type represents a subscription request related to notifications about VNF faults. It shall comply with the provisions defined in table 7.5.2.2-1.

Table 7.5.2.2-1: 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 [6]. This attribute shall only be present if the subscriber requires authorization of notifications.

7.5.2.3 Type: FmSubscription

This type represents a subscription related to notifications about VNF faults. It shall comply with the provisions defined in table 7.5.2.3-1.

Table 7.5.2.3-1: 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.

7.5.2.4 Type: Alarm

The alarm data type encapsulates information about an alarm. It shall comply with the provisions defined in table 7.5.2.4-1.

Table 7.5.2.4-1: 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 VNF instance.
vnfcInstanceIds	IdentifierInVnf	0..N	Identifiers of the affected VNFC instances. Each identifier references the "id" attribute in a "VnfcInfo" structure. Shall be present if the alarm affects at least one VNFC instance.
rootCauseFaultyResource	FaultyResourceInfo	0..1	The virtualised resources that are causing the VNF fault. Shall be present if the alarm affects virtualised resources. See note 1.
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.
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. See note 2.
eventType	EventType	1	Type of event.
faultType	String	0..1	Additional information to clarify the type of the fault. If the alarm is related to changes in the state of virtualised resources due to NFVI operation and maintenance, this attribute shall be set to "NFVI_OAM_VIRTUALISED_RESOURCE_STATE_CHANGE".

Attribute name	Data type	Cardinality	Description
probableCause	String	1	Information about the probable cause of the fault. If the attribute "faultType" has the value "NFVI_OAM_VIRTUALISED_RESOURCE_STATE_CHANGE", the permitted values are: • "NFVI_COMPONENT_MAINTENANCE": Maintenance of NFVI components, e.g. physical maintenance/repair, hypervisor software updates, etc.; • "NFVI_COMPONENT_EVACUATION": Evacuation of physical hosts; • "NFVI_COMPONENT_OPTIMIZATION": Operation and management of NFVI resources, e.g. to support energy efficiency or resource usage optimization.
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. See notes 1 and 2.
_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 VNF instance to which the notified alarm is correlated. Shall be present if the VNF instance information is accessible as a resource.
NOTE 1: For an alarm about upcoming impact due to NFVI operation and maintenance (i.e. the attribute "faultType" has the value "NFVI_OAM_VIRTUALISED_RESOURCE_STATE_CHANGE"), the attribute "rootCauseFaultyResource" indicates a resource to be impacted. Further information on the upcoming impact (e.g. group of impacted resources, time of impact) is provided in the attribute "faultDetails". NOTE 2: When alarms are due to upcoming NFVI operation and maintenance (i.e. the attribute "faultType" has the value "NFVI_OAM_VIRTUALISED_RESOURCE_STATE_CHANGE"), the attribute "faultDetails" shall include information about the anticipated time of the maintenance. See provisions under the present table.			

If the attribute "faultType" has the value "NFVI_OAM_VIRTUALISED_RESOURCE_STATE_CHANGE", the following provisions apply for the values of the attribute "faultDetails" related to changes in the state of virtualised resources:

- One of the entries in the array shall provide information about the anticipated time of maintenance in the following format: "anticipatedTime=$time", wherein "$time" shall be formatted as a "DateTime", as specified in ETSI GS NFV-SOL 013 [6].

- One of the entries in the array shall provide identification information about the affinity/anti-affinity group defined in the VNFD that is associated to the affected virtualised resource indicated by "rootCauseFaultyResource" in the following format: "affinityOrAntiAffinityGroupId=$group", wherein "$group" shall be equal to the "affinityOrAntiAffinityGroupId" value in the corresponding "VduProfile" (for a VNFC/COMPUTE affected resource) or "VirtualLinkProfile" for a VL/NETWORK affected resource) in the VNFD, which is mapped by the VNFM to the virtualised resource group identifier in the virtualised resource change notification received by the VNFM from the VIM.

7.5.2.5 Type: AlarmNotification

This type represents an alarm notification about VNF faults. It shall comply with the provisions defined in table 7.5.2.5-1.

This notification shall be triggered by the VNFM when:

- An alarm has been created.

- An alarm has been updated, e.g. the severity of the alarm has changed.

Table 7.5.2.5-1: 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 VNF identifier, and FaultDetails.
_links	Structure (inlined)	1	Links to resources related to this notification.
>subscription	NotificationLink	1	Link to the related subscription.

7.5.2.6 Type: AlarmClearedNotification

This type represents an alarm cleared notification about VNF faults. It shall comply with the provisions defined in table 7.5.2.6-1.

The notification shall be triggered by the VNFM when an alarm has been cleared.

Table 7.5.2.6-1: 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.

7.5.2.7 Type: PerceivedSeverityRequest

This type represents the escalated value of the perceived severity for an alarm. It shall comply with the provisions defined in table 7.5.2.7-1.

Table 7.5.2.7-1: Definition of the PerceivedSeverityRequest data type

Attribute name	Data type	Cardinality	Description
proposedPerceivedSeverity	PerceivedSeverityType	1	Indicates the proposed escalated perceived severity for an alarm.

7.5.2.8 Type: AlarmListRebuiltNotification

This type represents a notification that the alarm list has been rebuilt, e.g. if the VNFM detects its storage holding the alarm list is corrupted. It shall comply with the provisions defined in table 7.5.2.8-1.

The notification shall be triggered by the VNFM when the alarm list has been rebuilt, e.g. because the VNFM has detected that its storage holding the alarm list was corrupted.

Table 7.5.2.8-1: 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.

7.5.2.9 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 7.5.2.4 are included in the "AlarmModifications" data type.

The "AlarmModifications" data type shall comply with the provisions defined in table 7.5.2.9-1.

Table 7.5.2.9-1: 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.

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 Type: FmNotificationsFilter

This type represents a subscription filter related to notifications about VNF faults. It shall comply with the provisions defined in table 7.5.3.2-1.

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 7.5.3.2-1: Definition of the FmNotificationsFilter data type

Attribute name	Data type	Cardinality	Description
vnfInstanceSubscriptionFilter	VnfInstanceSubscriptionFilter	0..1	Filter criteria to select VNF 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 VNF alarms with a faulty resource type listed in this attribute.
perceivedSeverities	PerceivedSeverityType	0..N	Match VNF alarms with a perceived severity listed in this attribute.
eventTypes	EventType	0..N	Match VNF alarms with an event type listed in this attribute.
probableCauses	String	0..N	Match VNF alarms 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.			

7.5.3.3 Type: FaultyResourceInfo

This type represents the faulty virtual resources that have a negative impact on a VNF. It shall comply with the provisions defined in table 7.5.3.3-1.

Table 7.5.3.3-1: 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.

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.

7.5.4.3 Enumeration: PerceivedSeverityType

The enumeration PerceivedSeverityType shall comply with the provisions defined in table 7.5.4.3-1. It indicates the relative level of urgency for operator attention.

Table 7.5.4.3-1: 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 [4]).
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 [4]).
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 [4]).
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 [4]).
INDETERMINATE	The Indeterminate severity level indicates that the severity level cannot be determined (Recommendation ITU-T X.733 [4]).
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 [4]).

7.5.4.4 Enumeration: EventType

The enumeration EventType represents those types of events that trigger an alarm. It shall comply with the provisions defined in table 7.5.4.4-1.

Table 7.5.4.4-1: 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 [4]).
PROCESSING_ERROR_ALARM	An alarm of this type is associated with a software or processing fault (Recommendation ITU-T X.733 [4]).
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 [4]).
QOS_ALARM	An alarm of this type is associated with degradation in the quality of a service (Recommendation ITU-T X.733 [4]).
EQUIPMENT_ALARM	An alarm of this type is associated with an equipment fault (Recommendation ITU-T X.733 [4]).

7.5.4.5 Enumeration: FaultyResourceType

The enumeration FaultyResourceType represents those types of faulty resource. It shall comply with the provisions defined in table 7.5.4.5-1.

Table 7.5.4.5-1: Enumeration FaultyResourceType

Enumeration value	Description
COMPUTE	Virtual compute resource
STORAGE	Virtual storage resource
NETWORK	Virtual network resource

8 VNF Indicator interface

8.1 Description

This interface allows the EM/VNF to provide information on value changes of VNF related indicators. VNF related indicators are declared in the VNFD. Further, this interface allows API version information retrieval.

The support of the VNF Indicator interface depends on the VNF capabilities. If at least one VNF indicator is declared by the VNF provider in the VNFD for a particular VNF, support of this interface by the VNF or the corresponding EM is defined as follows:

- An EM shall support this interface if it is capable of managing at least one VNF for which at least one VNF indicator is declared in the related VNFD with the source defined to be "EM" or "both EM and VNF".
- A VNF shall support this interface if at least one VNF indicator is declared in the related VNFD with the source defined to be "VNF" or "both EM and VNF".

Otherwise, support for this interface is optional for the VNF and corresponding EM.

The operations provided through this interface are:

- Get Indicator Value
- Subscribe
- Query Subscription Information
- Terminate Subscription
- Notify

8.1a API version

For the VNF indicator interface version as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 3 and the PATCH version field shall be 1 (see clause 9.1 of ETSI GS NFV-SOL 013 [6] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v1".

NOTE: In the present document, there were no changes to the clauses defining the VNF indicator interface that are visible at interface level compared to the previous version of the present document; hence, the MAJOR/MINOR/PATCH version fields are kept the same.

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 [6]. The string "vnfind" 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 VNF Indicator interface exposed by the EM.

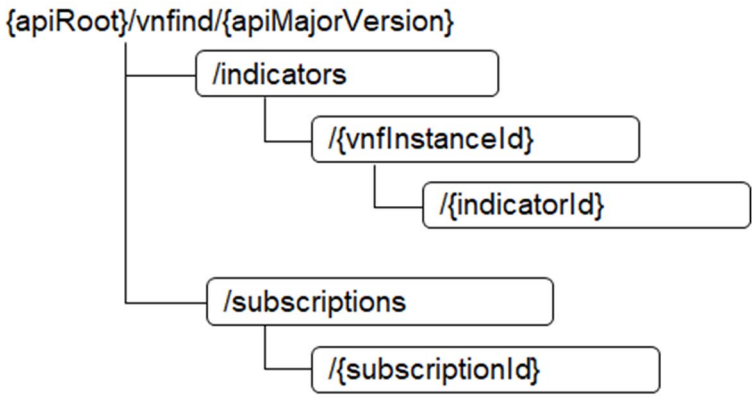

Figure 8.2-1: Resource URI structure of the VNF Indicator Interface exposed by the EM

Figure 8.2-2 shows the overall resource URI structure defined for the VNF Indicator interface exposed by the VNF.

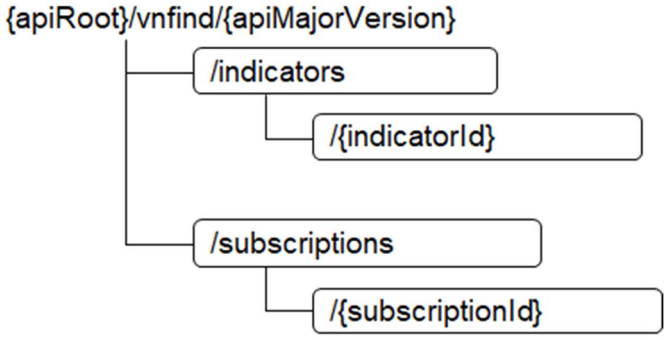

Figure 8.2-2: Resource URI structure of the VNF Indicator Interface exposed by the VNF

Table 8.2-1 lists the individual resources defined, and the applicable HTTP methods.

If the EM supports the VNF Indicator interface, the EM shall support responding to requests for all HTTP methods on the resources in table 8.2-1 that are marked as "M" (mandatory) or "M1" (mandatory for EM) in the "Cat" column.

If the VNF supports the VNF Indicator interface, the VNF shall support responding to requests for all HTTP methods on the resources in table 8.2-1 that are marked as "M" (mandatory) or "M2" (mandatory for VNF) in the "Cat" column.

The EM and VNF shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [6] if they support the VNF Indicator interface.

Table 8.2-1: Resources and methods overview of the VNF Indicator interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
VNF indicators	/indicators	GET	M	Query multiple VNF indicators. See note 1.
VNF indicators related to a VNF instance	/indicators/{vnfInstanceId}	GET	M1	Query multiple VNF indicators related to one VNF instance. See note 1.
Individual VNF indicator	/indicators/{vnfInstanceId}/{indicatorId}	GET	M1	Read an individual VNF indicator.
Individual VNF indicator	/indicators/{indicatorId}	GET	M2	Read an individual VNF indicator.
Subscriptions	/subscriptions	POST	M	Subscribe to VNF indicator change notifications.
		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 2	Notify about VNF indicator change.
		GET	See note 2.	Test the notification endpoint.
NOTE 1: This resource allows to query all VNF indicators that are known to the API producer. NOTE 2: The EM and VNF shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the VNFM. If the VNFM supports invoking the POST method on the "Subscription" resource towards the EM or VNF, 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 querying VNF indicators

This clause describes a sequence for querying VNF indicators from the API producer (EM or VNF).

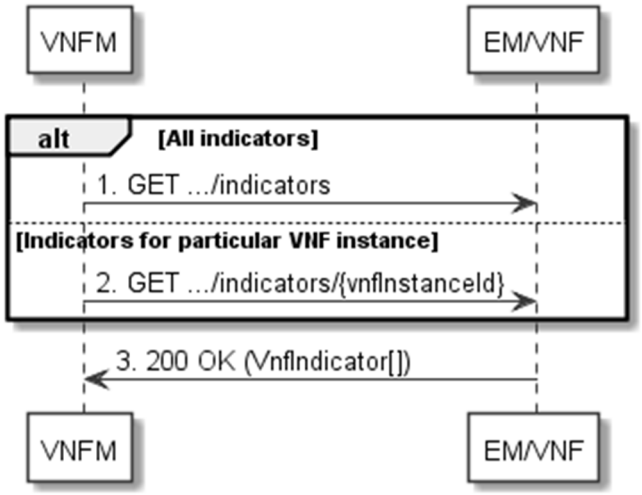

Figure 8.3.1-1: Flow of querying VNF indicators

VNF indicator query, as illustrated in figure 8.3.1-1, consists of the following steps:

1. If the VNFM intends to query all VNF indicators, it sends a GET request to the "VNF indicators" resource exposed by the EM or the VNF.

2. If the VNFM intends to query the VNF indicators of a particular VNF instance, it sends a GET request to the "VNF indicators related to a VNF instance" resource exposed by the EM.

3. The EM/VNF returns a "200 OK" response to the VNFM, and includes zero or more data structures of type "VnfIndicator" in the payload body.

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

8.3.2 Flow of reading a VNF indicator

This clause describes a sequence for reading a VNF indicator, i.e. for getting the indicator value, from the API producer (EM or VNF).

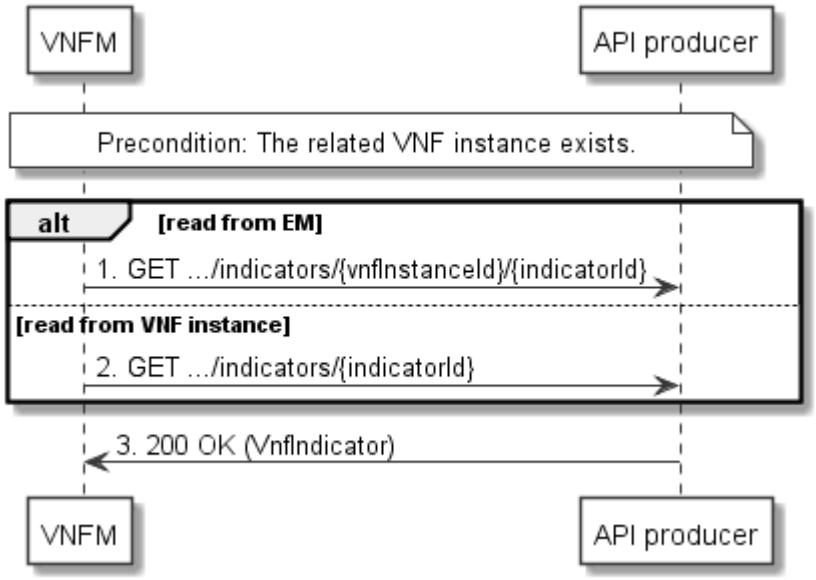

Figure 8.3.2-1: Flow of reading a VNF indicator

Precondition: The related VNF instance exists.

Reading a VNF indicator, as illustrated in figure 8.3.2-1, consists of the following steps:

1. If the VNFM wants to retrieve from the EM, an indicator value for a particular VNF instance, it sends a GET request to the "Individual VNF indicator" resource that is to be read in the EM. The URI contains the VNF instance identifier.

2. If the VNFM wants to retrieve an indicator value directly from a particular VNF instance, it sends a GET request to the corresponding "Individual VNF indicator" resource exposed by this VNF. The URI does not contain the VNF instance identifier.

3. The EM/VNF returns a "200 OK" response to the VNFM, and includes a data structure of type "VnfIndicator" 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, querying/reading and terminating subscriptions to notifications related to VNF indicator value changes.

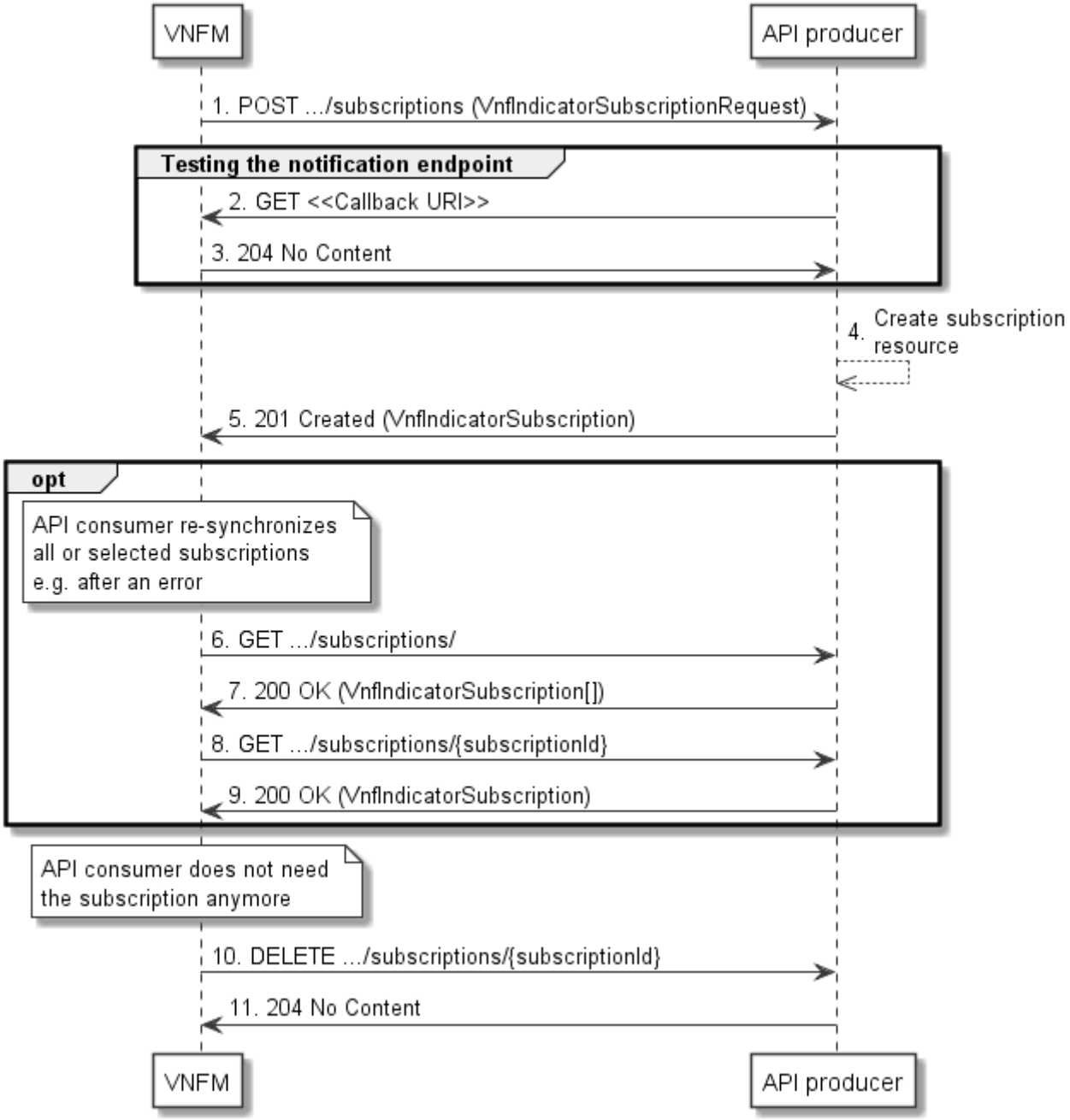

Figure 8.3.3-1: Flow of managing subscriptions

The procedure consists of the following steps as illustrated in figure 8.3.3-1:

1. The VNFM sends a POST request to the "Subscriptions" resource including in the payload body a data structure of type "VnfIndicatorSubscriptionRequest". That data structure contains filtering criteria and a callback URI to which the API producer will subsequently send notifications about events that match the filter.

2. To test the notification endpoint that has been registered by the VNFM as part of the subscription, the API producer sends a GET request to the notification endpoint URI.

3. The API producer returns a "204 No Content" response to indicate success.

4. The API producer creates a new subscription to notifications related to VNF indicator value changes, and a resource that represents this subscription.

5. The API producer returns a 201 Created response containing a data structure of type "VnfIndicatorSubscription" representing the "Individual subscription" resource just created by the API producer, 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 VNFM can query information about its subscriptions by sending a GET request to the resource representing the subscriptions.

7. In that case, the API producer returns a "200 OK" response that contains the list of representations of all existing subscriptions that were created by the VNFM.

8. If desired, e.g. to recover from an error situation, the VNFM can read information about a particular subscription by sending a GET request to the resource representing that individual subscription.

9. In that case, the API producer returns a "200 OK" response that contains a representation of that individual subscription.

10. If the VNFM 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 API producer acknowledges the successful termination of the subscription by returning a "204 No Content" response.

Error handling: The API producer 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.

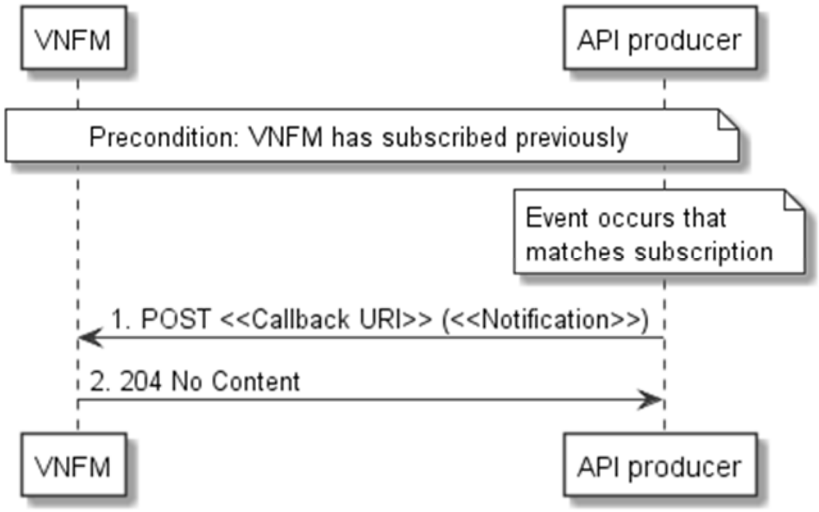

Figure 8.3.4-1: Flow of sending notifications

The procedure consists of the following steps as illustrated in figure 8.3.4-1:

Precondition: The VNFM has subscribed previously to notifications related to VNF indicator value changes.

1. If an event occurs that matches the filtering criteria which are part of the subscription, the API producer generates a notification that includes information about the event, and sends it in the body of a POST request to the callback URI which the VNFM 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 and 8.5.2.6).

2. The VNFM acknowledges the successful delivery of the notification by returning a "204 No Content" response.

Error handling: If the API producer does not receive the "204 No Content" response from the VNFM, it can retry sending the notification.

8.4 Resources

8.4.1 Introduction

This clause defines all the resources and methods provided by the VNF Indicator interface.

8.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [6] are part of the VNF indicator interface.

8.4.2 Resource: VNF indicators

8.4.2.1 Description

This resource represents VNF indicators. The API consumer can use this resource to query multiple VNF indicators.

8.4.2.2 Resource definition

The resource URI is:

{apiRoot}/vnfind/{apiMajorVersion}/indicators

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 [6].
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 API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.2.3.2 GET

The GET method queries multiple VNF indicators.

This method shall follow the provisions specified in 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	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [6]. The API producer shall support receiving this parameter as part of the URI query string. The VNFM may supply this parameter. All attribute names that appear in the VnfIndicator data type and in data types referenced from it shall be supported by the API producer in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the API producer if the API producer supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource.

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**
	VnfIndicator	0..N	200 OK	Shall be returned when information about zero or more VNF indicators has been queried successfully. The response body shall contain in an array the representations of all VNF indicators that match the attribute filter, i.e. zero or more representations of VNF indicators as defined in clause 8.5.2.2. 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 [6]. If the API producer supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] 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 [6].
	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 API producer supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [6].
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

8.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.3 Resource: VNF indicators related to a VNF instance

8.4.3.1 Description

This resource represents VNF indicators related to a VNF instance. The API consumer can use this resource to query multiple VNF indicators that are related to a particular VNF instance.

8.4.3.2 Resource definition

The resource URI is:

{apiRoot}/vnfind/{apiMajorVersion}/indicators/{vnfInstanceId}

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 [6].
apiMajorVersion	See clause 8.1a.
vnfInstanceId	Identifier of the VNF instance to which the VNF indicator applies. 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 instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

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 API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.3.3.2 GET

The GET method queries multiple VNF indicators related to a VNF instance.

This method shall follow the provisions specified in 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
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [6]. The API producer shall support receiving this parameter as part of the URI query string. The VNFM may supply this parameter. All attribute names that appear in the VnfIndicator data type and in data types referenced from it shall be supported by the API producer in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the API producer if the API producer supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource.

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**
	VnfIndicator	0..N	200 OK	Shall be returned when information about zero or more VNF indicators has been queried successfully. The response body shall contain in an array the representations of all VNF indicators that are related to the particular VNF instance and that match the attribute filter, i.e. zero or more representations of VNF indicators as defined in clause 8.5.2.2. 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 [6]. If the API producer supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] 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 [6].
	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 API producer supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [6].
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

8.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.3.3.4 PATCH

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.3.3.5 DELETE

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.4 Resource: Individual VNF indicator

8.4.4.1 Description

This resource represents an individual VNF indicator. The API consumer can use this resource to read an individual VNF indicator.

8.4.4.2 Resource definition

When the resource is exposed by the EM, the resource URI is:

{apiRoot}/vnfind/{apiMajorVersion}/indicators/{vnfInstanceId}/{indicatorId}

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 when exposed by the EM

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 8.1a.
vnfInstanceId	Identifier of the VNF instance to which the VNF indicator applies. See note 1.
indicatorId	Identifier of the VNF indicator. See note 2.
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 "Individual VNF instance" 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 resource referenced by the payload body in the response to a POST request creating a new "Individual VNF instance" resource.	

When the resource is exposed by the VNF instance, the resource URI is:

{apiRoot}/vnfind/{apiMajorVersion}/indicators/{indicatorId}

This resource shall support the resource URI variables defined in table 8.4.4.2-2.

Table 8.4.4.2-2: Resource URI variables for this resource when exposed by a VNF instance

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
apiMajorVersion	See clause 8.1a.
indicatorId	Identifier of the VNF indicator.

8.4.4.3 Resource methods

8.4.4.3.1 POST

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.4.3.2 GET

The GET method reads a VNF indicator.

This method shall follow the provisions specified in 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	Description
none supported		

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**
	VnfIndicator	1	200 OK	Shall be returned when the VNF indicator has been read successfully. The response body shall contain the representation of the VNF indicator.
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

8.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.5 Resource: Subscriptions

8.4.5.1 Description

This resource represents subscriptions. The API consumer can use this resource to subscribe to notifications related to VNF indicator value changes, and to query its subscriptions.

8.4.5.2 Resource definition

The resource URI is:

{apiRoot}/vnfind/{apiMajorVersion}/subscriptions

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 [6].
apiMajorVersion	See clause 8.1a.

8.4.5.3 Resource methods

8.4.5.3.1 POST

The POST method creates a new subscription.

As the result of successfully executing this method, a new "Individual subscription" resource as defined in clause 8.4.6 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 VNFM, and might make sense only in very rare use cases. Consequently, the API producer 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).

This method shall follow the provisions specified in tables 8.4.5.3.1-1 and 8.4.5.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.5.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 8.4.5.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	VnfIndicatorSubscription Request	1	Details of the subscription to be created.	
Response body	VnfIndicatorSubscription	1	201 Created	Shall be returned when the subscription has been created successfully. The response body shall contain a representation of the created "Individual subscription" resource. The HTTP response shall include a "Location" HTTP header that points to the created resource.
	n/a		303 See Other	Shall be returned when a subscription with the same callback URI and the same filter already exists and the policy of the API producer is to not create redundant subscriptions. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource. The response body shall be empty.
	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 [6], 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 VNFM has tested the Notification endpoint as described in clause 8.4.7.3.2 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 [6]	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 [6] may be returned.

8.4.5.3.2 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 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
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [6]. The EM shall and the VNF may support receiving this parameter as part of the URI query string. The VNFM may supply this parameter. All attribute names that appear in the VnfIndicatorSubscription data type and in data types referenced from it shall be supported in the filter expression. If receiving, this parameter is not supported, a 400 Bad Request response shall be returned (see table 8.4.5.3.2-2).
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the API producer if the API producer supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource.

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**
	VnfIndicatorSubscription	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 which match the attribute filter, i.e. zero or more representations of VNF indicator subscriptions as defined in clause 8.5.2.4. 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 [6]. If the API producer supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] 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 [6].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression or "filter" URI query parameter not supported. 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 API producer supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [6] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [6].
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

8.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.6 Resource: Individual subscription

8.4.6.1 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 VNF indicator value changes.

8.4.6.2 Resource definition

The resource URI is:

{apiRoot}/vnfind/{apiMajorVersion}/subscriptions/{subscriptionId}

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
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [6].
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.	

8.4.6.3 Resource methods

8.4.6.3.1 POST

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.6.3.2 GET

The GET method reads an individual subscription.

This method shall follow the provisions specified in 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**
	VnfIndicatorSubscription	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 [6]	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 [6] may be returned.

8.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.6.3.4 PATCH

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.6.3.5 DELETE

The DELETE method terminates an individual subscription.

This method shall follow the provisions specified in tables 8.4.6.3.5-1 and 8.4.6.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual 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 8.4.6.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 8.4.6.3.5-2: 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 "Individual subscription" resource has been deleted successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

8.4.7 Resource: Notification endpoint

8.4.7.1 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.

8.4.7.2 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 8.4.7.2-1.

Table 8.4.7.2-1: Resource URI variables for this resource

Name	Definition
none supported	

8.4.7.3 Resource methods

8.4.7.3.1 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 tables 8.4.7.3.1-1 and 8.4.7.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.7.3.1-1: 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 8.4.7.3.1-2.

Table 8.4.7.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	VnfIndicatorValueChangeNotification	1	A notification about VNF indicator value changes.	
	SupportedIndicatorsChangeNotification	1	A notification about changes of the set of supported indicators.	
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 [6]	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 [6] may be returned.

8.4.7.3.2 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 tables 8.4.7.3.2-1 and 8.4.7.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.7.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 8.4.7.3.2-2: 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 [6]	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 [6] may be returned.

8.4.7.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.7.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.4.7.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

8.5 Data model

8.5.1 Introduction

This clause defines the request and response data structures of the VNF Indicator 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 resource representations and notifications.

8.5.2.2 Type: VnfIndicator

This type represents a VNF indicator value. It shall comply with the provisions defined in table 8.5.2.2-1.

Table 8.5.2.2-1: Definition of the VnfIndicator data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnfd	1	Identifier of this VNF indicator.
name	String	0..1	Human readable name of the indicator. Shall be present if defined in the VNFD.
value	Object	1	Provides the value of the indicator. The value format is defined in the VNFD. See note.
vnfInstanceId	Identifier	1	Identifier of the VNF instance which provides the indicator value.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.
>vnfInstance	Link	1	Link to the related "Individual VNF instance" resource.
NOTE: ETSI GS NFV-SOL 001 [i.3] specifies the structure and format of the VNFD based on TOSCA specifications.			

8.5.2.3 Type: VnfIndicatorSubscriptionRequest

This type represents a subscription request related to VNF indicator value change notifications. It shall comply with the provisions defined in table 8.5.2.3-1.

Table 8.5.2.3-1: Definition of the VnfIndicatorSubscriptionRequest data type

Attribute name	Data type	Cardinality	Description
filter	VnfIndicatorNotificationsFilter	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 [6]. This attribute shall only be present if the subscriber requires authorization of notifications.

8.5.2.4 Type: VnfIndicatorSubscription

This type represents a subscription related to notifications about VNF indicator value changes. It shall comply with the provisions defined in table 8.5.2.4-1.

Table 8.5.2.4-1: Definition of the VnfIndicatorSubscription data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this "Individual subscription" resource.
filter	VnfIndicatorNotificationsFilter	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.

8.5.2.5 Type: VnfIndicatorValueChangeNotification

This type represents a VNF indicator value change notification. It shall comply with the provisions defined in table 8.5.2.5-1.

The notification shall be triggered by the API consumer when the value of an indicator has changed.

Table 8.5.2.5-1: Definition of the VnfIndicatorValueChangeNotification 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 "VnfIndicatorValueChangeNotification" 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.
vnfIndicatorId	IdentifierInVnfd	1	Identifier of the VNF indicator whose value has changed.
name	String	0..1	Human readable name of the VNF indicator. Shall be present if defined in the VNFD.
value	Object	1	Provides the value of the VNF indicator. The value format is defined in the VNFD. See note.
vnfInstanceId	Identifier	1	Identifier of the VNF instance which provides the indicator value.
_links	Structure (inlined)	1	Links for this resource.
>vnfInstance	NotificationLink	1	Link to the related "Individual VNF instance" resource. Shall be present if the VNF instance information is accessible as a resource.
>subscription	NotificationLink	1	Link to the related subscription.
NOTE: ETSI GS NFV-SOL 001 [i.3] specifies the structure and format of the VNFD based on TOSCA specifications.			

8.5.2.6 Type: SupportedIndicatorsChangeNotification

This type represents a notification to inform the receiver that the set of indicators supported by a VNF instance has changed. It shall comply with the provisions defined in table 8.5.2.6-1.

The notification shall be triggered by the API producer when the set of supported VNF indicators has changed as a side effect of the "Change current VNF package" operation. It may be triggered by the API producer when a VNF has been instantiated.

Table 8.5.2.6-1: Definition of the SupportedIndicatorsChangeNotification 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 "SupportedIndicatorsChangeNotification" 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.
vnfInstanceId	Identifier	1	Identifier of the VNF instance which provides the indicator value.
supportedIndicators	Structure (inlined)	0..N	Set of VNF indicators supported by the VNF instance.
>vnfIndicatorId	IdentifierInVnfd	1	Identifier of the VNF indicator whose value has changed.
>name	String	0..1	Human readable name of the VNF indicator. Shall be present if defined in the VNFD. See note.
_links	Structure (inlined)	1	Links for this resource.
>vnfInstance	NotificationLink	0..1	Link to the related "Individual VNF instance" resource. Shall be present if the VNF instance information is accessible as a resource.
>subscription	NotificationLink	1	Link to the related subscription.
NOTE: ETSI GS NFV-SOL 001 [i.3] specifies the structure and format of the VNFD based on TOSCA specifications.			

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: VnfIndicatorNotificationsFilter

This type represents a subscription filter for notifications related to VNF indicators. 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 VnfIndicatorNotificationsFilter data type

Attribute name	Data type	Cardinality	Description
vnfInstanceSubscriptionFilter	VnfInstanceSubscriptionFilter	0..1	Filter criteria to select VNF instances about which to notify. See note 1.
notificationTypes	Enum (inlined)	0..N	Match particular notification types. Permitted values: • VnfIndicatorValueChangeNotification • SupportedIndicatorsChangeNotification See note 2.
indicatorIds	IdentifierInVnfd	0..N	Match particular VNF indicator identifiers.
NOTE 1: This attribute shall not be included when the VNFM sends a subscription request to a particular VNF instance. NOTE 2: The permitted values of the "notificationTypes" attribute are spelled exactly as the names of the notification types to facilitate automated code generation systems.			

8.5.4 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.

9 VNF Configuration interface

9.1 Description

This interface allows the VNFM to set configuration of a VNF instance and/or its VNFC instance(s).

Further, this interface allows API version information retrieval.

The support of the VNF Configuration interface is optional.

The operation provided through this interface is:

- Set Configuration

9.1a API version

For the VNF configuration interface version as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 2, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [6] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v1".

NOTE: In the present document, there were no changes to the clauses defining the VNF configuration management interface that are visible at interface level compared to the previous version of the present document; hence, the MAJOR/MINOR/PATCH version fields are kept the same.

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 [6]. The string "vnfconfig" 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 configuration interface.

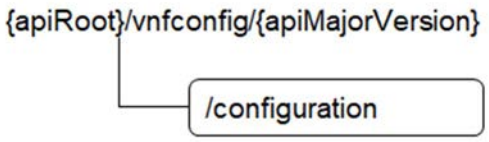

Figure 9.2-1: Resource URI structure of VNF configuration Interface

Table 9.2-1 lists the individual resources defined, and the applicable HTTP methods.

If the VNF supports the VNF configuration interface, the VNF 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 VNF shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [6].

Table 9.2-1: Resources and methods overview of the VNF configuration interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
Configuration	/configuration	PATCH	M	Set configuration data of a VNF instance and/or its VNFC instances.
		GET	M	Read configuration data of a VNF instance and its VNFC instances.

9.3 Sequence diagrams (informative)

9.3.1 Flow of setting the VNF configuration

This clause describes the procedure for setting the configuration of a VNF instance and/or its VNFC instances.

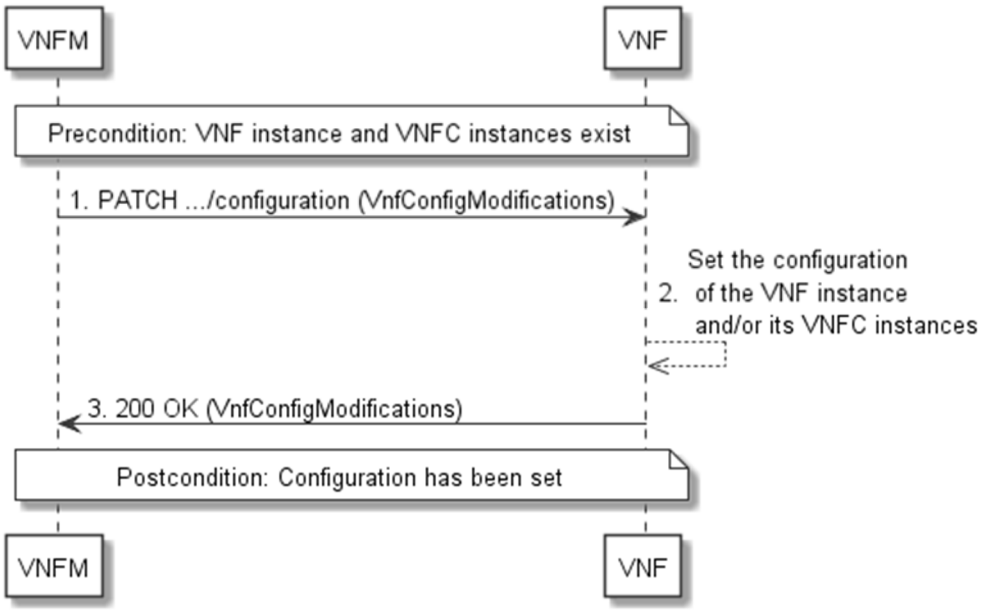

Figure 9.3.1-1: Flow of setting the configuration of a VNF instance and/or its VNFC instances

The procedure consists of the following steps as illustrated in figure 9.3.1-1.

Precondition: A VNF instance and its VNFC instances exist:

1. The VNFM sends a PATCH request to the "configuration" resource including in the payload body a data structure of type "VnfConfigModifications".
2. The VNF sets the configuration of the VNF instance and/or its VNFC instances.
3. The VNF returns a "200 OK" response to the VNFM, and includes a data structure of type "VnfConfigModifications" in the payload body.

Postcondition: Configuration of the VNF instance and/or its VNFC instances has been set.

9.4 Resources

9.4.1 Introduction

This clause defines all the resources and methods provided by the VNF configuration interface.

9.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [6] are part of the VNF configuration interface.

9.4.2 Resource: Configuration

9.4.2.1 Description

This resource represents the configuration of a VNF instance and its VNFC instances. The API consumer can use this resource to set and read the configuration of a VNF instance and its VNFC instances.

9.4.2.2 Resource definition

The resource URI is:

{apiRoot}/vnfconfig/{apiMajorVersion}/configuration

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 [6].
apiMajorVersion	See clause 9.1a.

9.4.2.3 Resource methods

9.4.2.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

9.4.2.3.2 GET

The API consumer can use this method to read configuration information about a VNF instance and/or its VNFC instances.

This method shall follow the provisions specified in 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
none supported		

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**
	VnfConfiguration	1	200 OK	Shall be returned when configuration information about a VNF instance has been read successfully. The response body shall contain a representation of the configuration resource, as defined in clause 9.5.3.2.
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

9.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

9.4.2.3.4 PATCH

This method sets or modifies a configuration resource.

This method shall follow the provisions specified in tables 9.4.2.3.4-1 and 9.4.2.3.4-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the configuration of the VNF instance and/or its VNFC instances shall have been changed as requested.

Table 9.4.2.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.2.3.4-2: Details of the PATCH request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	VnfConfigModifications	1	The parameter for the configuration modification, as defined in clause 9.5.2.2.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfConfigModifications	1	200 OK	Shall be returned when the request has been accepted and completed. The response body shall contain the parameters of the configuration modification that was applied to the configuration resource (see clause 9.5.2.2).
	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 [6]	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 [6] may be returned.

9.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

9.5 Data model

9.5.1 Introduction

This clause defines the request and response data structures of the VNF Configuration 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 the data structures to be used in resource representations and notifications.

9.5.2.2 Type: VnfConfigModifications

This type represents request parameters for the "Set Configuration" operation. It shall comply with the provisions defined in table 9.5.2.2-1.

Table 9.5.2.2-1: Definition of the VnfConfigModifications data type

Attribute name	Data type	Cardinality	Description
vnfConfigurationData	VnfConfigurationData	0..1	Modifications to configuration data for the VNF instance. See note 1. If present, the modifications of the "vnfConfigurationData" attribute shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [3]).
vnfcConfigurationData	VnfcConfigurationData	0..N	Modifications to configuration data for certain VNFC instances. See note 1 and note 2. If present, the modifications of the "vnfcConfigurationData" attribute shall follow the provisions defined below this table.
vnfcConfigurationData DeleteIds	Identifier	0..N	List of identifiers entries to be deleted from the "vnfcConfigurationData" attribute array to be used as "deleteIdList" as defined below this table.
NOTE 1: At least one of "vnfConfigurationData" and "vnfcConfigurationData" shall be present. NOTE 2: The VnfcConfiguration data type can only be used to modify the configuration of existing VNFC instances.			

The following provisions shall apply when modifying an attribute that is an array of objects of type "VnfcConfigurationData".

Assumptions:

1) "oldList" is the "VnfcConfigurationData" array to be modified, "newList" is the "VnfcConfigurationData" array that contains the changes and "deleteIdList" is the array that contains the identifiers of those "oldList" entries to be deleted.

2) "oldEntry" is an entry in "oldList" and "newEntry" is an entry in "newList".

3) A "newEntry" has a "corresponding entry" if there exists an "oldEntry" that has the same content of the "vnfcInstanceId" attribute as the "newEntry"; a "newEntry" has no corresponding entry if no such "oldEntry" exists.

4) In any array of "VnfcConfigurationData" structures, the content of "vnfcInstanceId" is unique (i.e. there shall be no two entries with the same content of "vnfcInstanceId").

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".

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 can neither be resource representations nor bound to any subscribe/notify mechanism.

9.5.3.2 Type: VnfConfiguration

This type represents configuration parameters of a VNF instance and its VNFC instances. It shall comply with the provisions defined in table 9.5.3.2-1.

Table 9.5.3.2-1: Definition of the VnfConfiguration data type

Attribute name	Data type	Cardinality	Description
vnfConfigurationData	VnfConfigurationData	1	Configuration parameters of the VNF instance
vnfcConfigurationData	VnfcConfigurationData	0..N	Configuration parameters of the VNFC instances

9.5.3.3 Type: VnfConfigurationData

This type represents configuration parameters of a VNF instance. It shall comply with the provisions defined in table 9.5.3.3-1.

Table 9.5.3.3-1: Definition of the VnfConfigurationData data type

Attribute name	Data type	Cardinality	Description
extCpConfig	CpConfiguration	0..N	Configuration parameters for the external CPs of the VNF instance.
dhcpServer	IpAddress	0..1	IP address of the DHCP server that the VNF instance can use to obtain IP addresses to be assigned to its external CPs.
vnfSpecificData	KeyValuePairs	0..1	Additional configurable properties of the VNF instance declared in the VNFD as "VnfConfigurableProperties". See note.
NOTE: ETSI GS NFV-SOL 001 [i.3] specifies the structure and format of the VNFD based on TOSCA specifications.			

9.5.3.4 Type: VnfcConfigurationData

This type represents configuration parameters of a VNFC instance. It shall comply with the provisions defined in table 9.5.3.4-1.

Table 9.5.3.4-1: Definition of the VnfcConfigurationData data type

Attribute name	Data type	Cardinality	Description
vnfcInstanceId	IdentifierInVnf	1	Identifier of a VNFC instance to which this set of configuration data applies. The identifier references the "id" attribute in a "VnfcInfo" structure.
intCpConfig	CpConfiguration	0..N	Configuration parameters for the internal CPs of the VNFC instance.
dhcpServer	IpAddress	0..1	IP address of the DHCP server that the VNF can use to obtain IP addresses to be assigned to its CPs.
vnfcSpecificData	KeyValuePairs	0..1	Additional configurable properties of the VNFC instance declared in the VNFD as "VnfcConfigurableProperties". See note.
NOTE: ETSI GS NFV-SOL 001 [i.3] specifies the structure and format of the VNFD based on TOSCA specifications.			

9.5.3.5 Type: CpConfiguration

This type represents configuration parameters of a CP instance. It shall comply with the provisions defined in table 9.5.3.5-1.

Table 9.5.3.5-1: Definition of the CpConfiguration data type

Attribute name	Data type	Cardinality	Description
cpId	IdentifierInVnf	1	Identifier of a CP instance within the namespace of a specific VNF instance or a VNFC instance
cpdId	IdentifierInVnfd	1	Identifier of the CPD in the VNFD
addresses	CpAddress	1..N	Network address and port assigned to the CP

9.5.3.6 Type: CpAddress

This type represents configuration parameters of a CP instance address. It shall comply with the provisions defined in table 9.5.3.6-1.

Table 9.5.3.6-1: Definition of the CpAddress data type

Attribute name	Data type	Cardinality	Description
address	Structure (inlined)	0..1	Network address that has been configured on the CP. See note 1.
>macAddress	MacAddress	0..1	Mac address. See note 2.
>ipAddress	IpAddress	0..1	IP address. See note 2.
useDynamicAddress	Boolean	0..1	Set to true if an address shall be assigned dynamically. Otherwise set to false. The default value shall be false. See note 1.
port	UnsignedInt	0..1	The port assigned to the CP instance (e.g. IP port number, Ethernet port number, etc.).
NOTE 1: Either "address" or "useDynamicAddress" shall be present. NOTE 2: At least one of "macAddress" and "ipAddress" shall be present.			

9.5.4 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.

10 VNF LCM Coordination interface

10.1 Description

This interface allows the VNFM to request a VNF instance or EM to perform coordinative actions during LCM operations. Further, this interface allows API version information retrieval.

The operation provided through this interface is:

- CoordinateLcmOperation

The support of the VNF LCM Coordination interface by the VNF and the EM is optional and is declared in the VNFD.

NOTE: This interface follows the information model and requirements defined in ETSI GS NFV-IFA 008 [1] and supports a synchronous and an asynchronous mode as described in clause 10.3.1 "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.

10.1a API version

For the VNF LCM coordination interface as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 0 and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV SOL 013 [6] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v1".

NOTE: In the present document, there were no changes to the clauses defining the VNF LCM coordination interface that are visible at interface level compared to the previous version of the present document; hence, the MAJOR/MINOR/PATCH version fields are kept the same.

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 [6]. 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 10.2-1 shows the overall resource URI structure defined for the VNF LCM coordination interface.

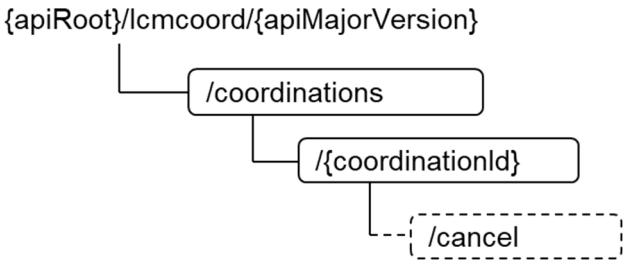

Figure 10.2-1: Resource URI structure of the VNF LCM coordination interface

Table 10.2-1 lists the individual resources defined, and the applicable HTTP methods.

If the VNF/EM supports the VNF LCM coordination interface, the EM/VNF 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 EM/VNF shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [6].

Table 10.2-1: Resources and methods overview of the VNF 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

10.3 Sequence diagrams (informative)

10.3.1 Flow of LCM coordination

This clause describes a sequence for an LCM coordination between the VNFM and the EM/VNF.

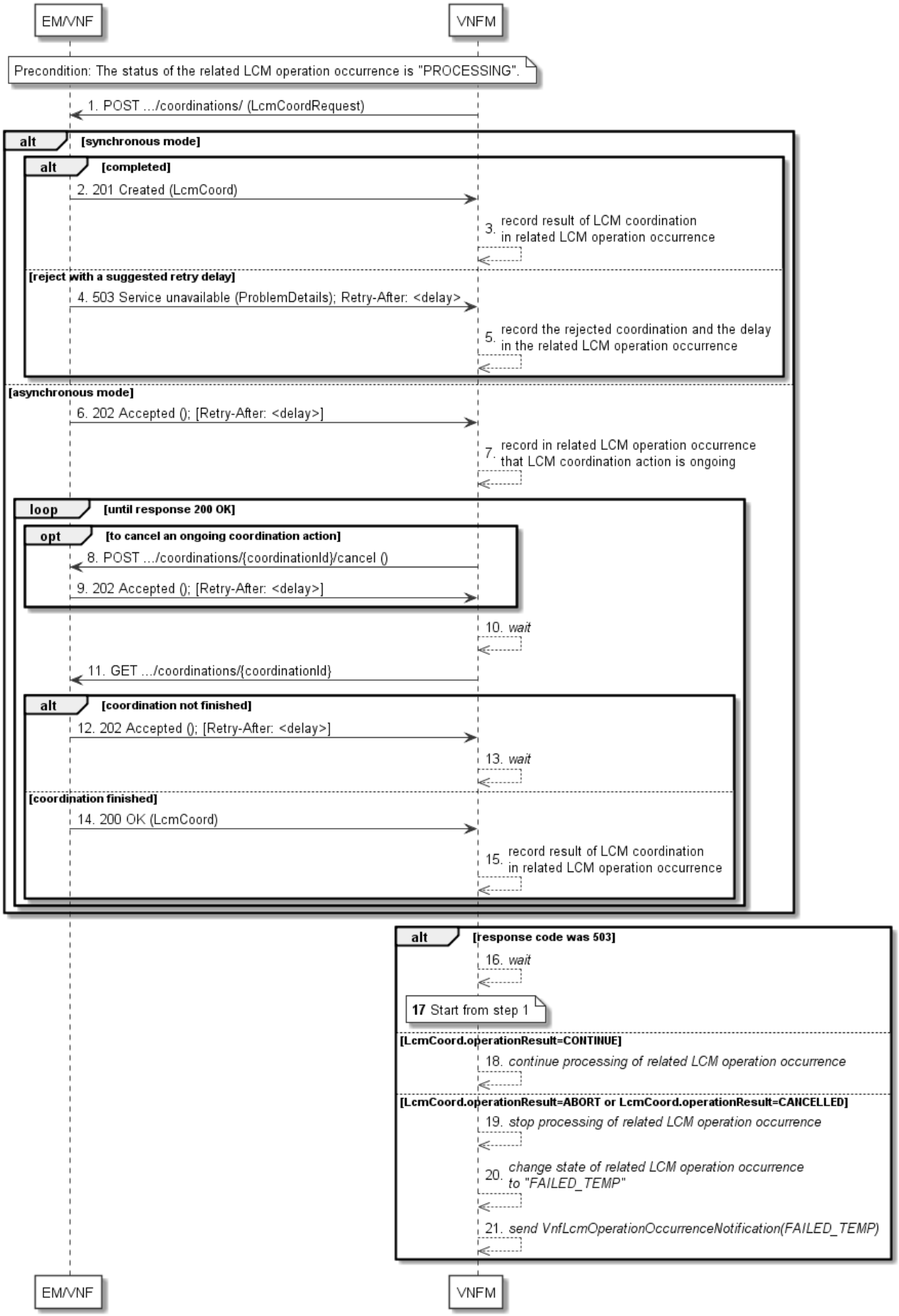

Figure 10.3.1-1: Flow of LCM coordination

An LCM coordination occurs always in the context of an LCM operation occurrence which is represent as a resource in the VNF lifecycle management API (see clause 5.4.13). 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".

LCM coordination, as illustrated in figure 10.3.1-1, consists of the following steps:

1. The VNFM 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 EM/VNF returns to the VNFM 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 VNFM 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 EM/VNF returns to the VNFM a "503 Service unavailable" response with an "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 VNFM 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 VNFM starts again at step 1.

Asynchronous mode:

6. The EM/VNF returns to the VNFM 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 VNFM 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 VNFM sends a POST request to the "Cancel coordination action task" resource with an empty request body.

9. In that case, the EM/VNF starts the cancellation and returns to the VNFM 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 VNFM 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 VNFM polls the status of the coordination by sending a GET request to the EM/VNF, using the URI that was returned in step 6. in the "Location" header.

12. If the coordination action is ongoing at the EM/VNF and consequently the "Individual coordination action" resource is still in the process of being created, the EM/VNF 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 VNFM 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 EM/VNF returns a "200 OK" response with an "LcmCoord" data structure in the body.

15. In this case, the VNFM 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 VNFM waits for the time interval indicated in the Retry-After header.

17. In that case, after waiting, the VNFM 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 VNFM continues the processing of the LCM operation.

19. If the coordinationResult resulting from the coordination was "ABORT" or CANCELLED, the VNFM stops the processing of the LCM operation.

20. In that case, the VNFM sets the state of the related LCM operation occurrence resource to "FAILED_TEMP".

21. Further in that case, the VNFM 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, LcmOperationOccurrenceNotifications have been sent to subscribed entities.

10.4 Resources

10.4.1 Introduction

This clause defines all the resources and methods provided by the VNF LCM coordination interface.

10.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [6] are part of the VNF LCM coordination interface.

10.4.2 Resource: Coordinations

10.4.2.1 Description

This resource represents LCM coordination actions. The VNFM 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.

10.4.2.2 Resource definition

The resource URI is:

{apiRoot}/lcmcoord/{apiMajorVersion}/coordinations

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 [6].
apiMajorVersion	See clause 10.1a.

10.4.2.3 Resource methods

10.4.2.3.1 POST

This 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 tables 10.4.2.3.1-1 and 10.4.2.3.1-2 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 VNFM shall send the coordination request again with the same parameters after the signalled time interval has passed, unless the VNFM 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 VNFM should not send the subsequent GET request before the signalled time interval has passed.

Table 10.4.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 10.4.2.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	LcmCoordRequest	1		Parameters for the coordination action as defined in clause 10.5.2.2.
Response body	LcmCoord	1	201 Created	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. The response body shall contain an LcmCoord data structure that represents the result of the coordination action. 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.

	n/a		202 Accepted	Shall be returned when the API producer has chosen the asynchronous mode and the request has been accepted for processing. The response body shall be empty. 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 VNFM shall record the signalled delay value in the "delay" attribute of the applicable entry in the "lcmCoordinations" array in the "VnfLcmOpOcc" structure.
	ProblemDetails	1	403 Forbidden	Shall be returned upon the following error: The starting of the coordination operation has been rejected. No "individual coordination action" resource shall be created. A ProblemDetails structure shall be included in the response to provide more details about the rejection in the "details" attribute.
	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 "Coordinations" resource. 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. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	503 Service unavailable	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. No "individual coordination action" resource shall be created. A ProblemDetails structure shall be included in the response to provide more details about the rejection in the "details" attribute. 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 VNFM shall record the signalled delay value in the "delay" attribute of the applicable entry in the "rejectedLcmCoordinations" array in the "VnfLcmOpOcc" structure.
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

10.4.2.3.2 GET

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

10.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

10.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

10.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

10.4.3 Resource: Individual coordination action

10.4.3.1 Description

This resource represents an individual coordination action. The VNFM 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 VNFM will wait for the API producer to provide a response.

10.4.3.2 Resource definition

The resource URI is:

{apiRoot}/lcmcoord/{apiMajorVersion}/coordinations/{coordinationId}

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 [6].
apiMajorVersion	See clause 10.1a.
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.	

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 API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

10.4.3.3.2 GET

The GET method reads a coordination result.

This method shall follow the provisions specified in 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
none supported		

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**
	LcmCoord	1	200 OK	Shall be returned when the coordination is finished and the coordination result has been read successfully. A representation of the "Individual coordination action" resource shall be returned in the response body.
	n/a		202 Accepted	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. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [6]	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 [6] may be returned.

10.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

10.4.3.3.4 PATCH

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

10.4.3.3.5 DELETE

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

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.

10.4.4 Resource: Cancel coordination action task

10.4.4.1 Description

This task resource represents the "cancel" operation related to an individual coordination action. The VNFM can use this resource to request the cancellation of an ongoing individual coordination action.

10.4.4.2 Resource definition

The resource URI is:

{apiRoot}/lcmcoord/{apiMajorVersion}/coordinations/{coordinationId}/cancel

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 [6].
apiMajorVersion	See clause 10.1a.
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.	

10.4.4.3 Resource methods

10.4.4.3.1 POST

The POST method initiates the cancellation of an ongoing coordination action.

This method shall follow the provisions specified in 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**	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the cancellation request has been accepted for processing. The response shall have an empty payload body.
	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 coordination action" resource. Typically, this is due to the fact that the coordination action has finished processing. 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 [6]	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 [6] may be returned.

10.4.4.3.2 GET

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

10.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

10.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

10.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [6].

10.5 Data model

10.5.1 Introduction

This clause defines the request and response data structures of the VNF 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 VNF and the LCM operation. They are declared in the VNFD.

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: LcmCoordRequest

This type represents request parameters for the "Set Configuration" operation. It shall comply with the provisions defined in table 10.5.2.2-1.

Table 10.5.2.2-1: Definition of the LcmCoordRequest data type

Attribute name	Data type	Cardinality	Description
vnfInstanceId	Identifier	1	Identifier of the VNF instance which this coordination request is related to.
vnfLcmOpOccId	Identifier	1	The identifier of the VNF 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 "operation" attribute in the LcmOpOcc structure that is referenced by the "vnfLcmOpOccId".
coordinationActionName	IdentifierInVnfd	1	Indicates the LCM coordination action. The coordination actions that a VNF supports are declared in the VNFD.
inputParams	KeyValuePairs	0..1	Additional parameters passed as input to the coordination action.
_links	Structure (inlined)	1	Links to resources related to this request.
>vnfLcmOpOcc	Link	1	Related lifecycle management operation occurrence.
>vnfInstance	Link	1	Related VNF instance.

10.5.2.3 Type: LcmCoord

This type represents an LCM coordination result. It shall comply with the provisions defined in table 10.5.2.3-1.

Table 10.5.2.3-1: 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 VNFM as the result of this coordination.
vnfInstanceId	Identifier	1	Identifier of the VNF instance which this coordination request is related to.
vnfLcmOpOccId	Identifier	1	The identifier of the VNF 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 "operation" attribute in the LcmOpOcc structure that is referenced by the "vnfLcmOpOccId".
coordinationActionName	String	1	Indicates the actual LCM coordination action. The coordination actions that a VNF supports are declared in the VNFD.
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 VnfLcmOpOcc data structure.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource
>vnfLcmOpOcc	Link	1	Related lifecycle management operation occurrence.
>vnfInstance	Link	1	Related VNF instance.

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 are not resource representations. In the present version of the present document, no such types are defined.

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.

10.5.4.3 Enumeration: LcmOperationForCoordType

The enumeration LcmOperationForCoordType defines the permitted values to represent VNF lifecycle operation types in VNF LCM operation coordination actions. It shall comply with the provisions defined in table 10.5.4.3-1.

Table 10.5.4.3-1: Enumeration LcmOperationForCoordType

Enumeration value	Description
INSTANTIATE	Represents the "Instantiate VNF" LCM operation.
SCALE	Represents the "Scale VNF" LCM operation.
SCALE_TO_LEVEL	Represents the "Scale VNF to Level" LCM operation.
CHANGE_FLAVOUR	Represents the "Change VNF Flavour" LCM operation.
TERMINATE	Represents the "Terminate VNF" LCM operation.
HEAL	Represents the "Heal VNF" LCM operation.
OPERATE	Represents the "Operate VNF" LCM operation.
CHANGE_EXT_CONN	Represents the "Change external VNF connectivity" LCM operation.
MODIFY_INFO	Represents the "Modify VNF Information" LCM operation.
CREATE_SNAPSHOT	Represents the "Create VNF Snapshot" LCM operation.
REVERT_TO_SNAPSHOT	Represents the "Revert To VNF Snapshot" LCM operation.
CHANGE_VNFPKG	Represents the "Change current VNF package" LCM operation.

10.6 Standardized coordination actions

10.6.1 Introduction

The following clauses define standardized VNF LCM coordination actions. A VNF or its operation supporting management systems (e.g. EM) may support zero or more of these actions.

For each coordination action, a standardized value for the coordination action name is defined, and valid values for the operation stage are specified. The meaning of "operationStage" is defined in ETSI GS NFV-IFA011 [7]. Further, for each coordination action, it is defined whether input or output parameters are applicable as defined in clauses 10.5.2.2 and 10.5.2.3, and their data types are specified.

10.6.2 Taking a VNF instance out of service

This coordination action allows the VNFM to request taking a VNF instance out of service. This coordination can be used in the context of the graceful modes of terminating (TerminateVnf) and stopping (OperateVnf) a VNF instance.

The coordination action shall follow the provisions defined in tables 10.6.2-1, 10.6.2-2 and 10.6.2-3.

Table 10.6.2-1: Definition of values

Attribute name	Definition
coordinationActionName	"urn:etsi:nfv:coord:take-vnf-out-of-service"
operationStage	"START" shall be the only allowed value.

Table 10.6.2-2: Data type of the "inputParams" attribute in "LcmCoordRequest"

Attribute name	Data type	Cardinality	Description
(none)			

Table 10.6.2-3: Data type of the "outputParams" attribute in "LcmCoord"

Attribute name	Data type	Cardinality	Description
(none)			

10.6.3 Taking VNFC instances of a VNF instance out of service

This coordination action allows the VNFM to request taking a set of individual VNFC instances of a VNF instance out of service. The coordination action shall follow the provisions defined in tables 10.6.3-1, 10.6.3-2 and 10.6.3-3.

This coordination can be used in the context of the graceful mode of stopping (OperateVnf) one or more VNFC instances of a VNF instance.

Table 10.6.3-1: Definition of values

Attribute name	Definition
coordinationActionName	"urn:etsi:nfv:coord:take-vnfcs-out-of-service"
operationStage	"START" shall be the only allowed value.

Table 10.6.3-2: Data type of the "inputParams" attribute in "LcmCoordRequest"

Attribute	Cardinality	Content	Description
vnfcInstanceIds	1..N	Identifier	List of identifiers of VNFC instances to be taken out of service.

Table 10.6.3-3: Data type of the "outputParams" attribute in "LcmCoord"

Attribute	Cardinality	Content	Description
(none)			

10.7 Conventions for coordination action names

To distinguish between public and private LCM coordination actions and to avoid collisions, this clause defines namespaces and conventions for the values of the coordination action name. The following naming conventions for defining coordination action name strings apply:

1) The name of a public coordination action (i.e. one that is defined in a public document) shall be represented by a URN (see IETF RFC 8141 [8]) where the namespace identifier (NID) of the URN is registered to the organization that issues the public document and where the namespace specific string (NSS) indicates the name of the coordination action unique within the scope defined by the NID.

2) The name of a private coordination action (i.e. one that is not defined in a public document or where the issuing organization does not have a registered URN namespace identifier) shall start with a prefix consisting of the string "prv.", followed by the name of the organization, followed by a dot ".", which shall be followed by the name of the coordination action unique within the scope indicated by the prefix.

3) Only alphanumeric characters and ".", "-", "_" should be used in the part of a coordination action name following the NID or prefix.

4) A coordination action name string defined by ETSI shall be prefixed by "urn:etsi:", followed by an NSS-root registered in https://portal.etsi.org/PNNS/Generic-Allocation/ETSI-URN-Namespace, followed by a string documented in an ETSI specification.

5) A coordination action name string defined by ETSI NFV shall be prefixed by "urn:etsi:nfv:coord:", followed by a string documented in an ETSI NFV specification.

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 008 [1] and the corresponding resources and HTTP methods defined in the present document.

A.2 VNF Lifecycle Management interface

ETSI GS NFV-IFA 008 [1] operation	HTTP method	Resource	Direction
Create VNF identifier	POST	/vnf_instances	EM → VNFM
Instantiate VNF	POST	/vnf_instances/{vnfInstanceId}/instantiate	EM → VNFM
Scale VNF	POST	/vnf_instances/{vnfInstanceId}/scale	EM → VNFM VNF → VNFM
Scale VNF to Level	POST	/vnf_instances/{vnfInstanceId}/scale_to_level	EM → VNFM VNF → VNFM
Change VNF Flavour	POST	/vnf_instances/{vnfInstanceId}/change_flavour	EM → VNFM
Terminate VNF	POST	/vnf_instances/{vnfInstanceId}/terminate	EM → VNFM
Delete VNF identifier	DELETE	/vnf_instances/{vnfInstanceId}	EM → VNFM
Query VNF	GET	/vnf_instances/{vnfInstanceId}	EM → VNFM VNF → VNFM
	GET	/vnf_instances	EM → VNFM VNF → VNFM
Heal VNF	POST	/vnf_instances/{vnfInstanceId}/heal	EM → VNFM VNF → VNFM
Operate VNF	POST	/vnf_instances/{vnfInstanceId}/operate	EM → VNFM
Change external VNF Connectivity	POST	/vnf_instances/{vnfInstanceId}/change_ext_conn	EM → VNFM
Modify VNF information	PATCH	/vnf_instances/{vnfInstanceId}	EM → VNFM
Change current VNF package	POST	/vnf_instances/{vnfInstanceId}/change_vnfpkg	EM → VNFM
Create VNF/VNFC Snapshot	POST	/vnf_instances/{vnfInstanceId}/create_snapshot	EM → VNFM
Revert to VNF/VNFC Snapshot	POST	/vnf_instances/{vnfInstanceId}/revert_to_snapshot	EM → VNFM
Delete VNF/VNFC Snapshot information	DELETE	/vnf_snapshots/{vnfSnapshotInfoId}	EM → VNFM
Query VNF/VNFC Snapshot information	GET	/vnf_snapshots/{vnfSnapshotInfoId}	EM → VNFM
	GET	/vnf_snapshots	EM → VNFM
Get Operation Status	GET	/vnf_lcm_op_occs	EM → VNFM VNF → VNFM
	GET	/vnf_lcm_op_occs/{vnfLcmOpOccId}	EM → VNFM VNF→ VNFM
Subscribe	POST	/subscriptions	EM → VNFM VNF → VNFM
Query Subscription Information	GET	/subscriptions	EM → VNFM VNF → VNFM
	GET	/subscriptions/{subscriptionId}	EM → VNFM VNF → VNFM
Terminate subscription	DELETE	/subscriptions/{subscriptionId}	EM → VNFM VNF → VNFM
Notify	POST	(provided by API consumer)	VNFM → EM VNFM → VNF

A.3 VNF Performance Management interface

ETSI GS NFV-IFA 008 [1] operation	HTTP method	Resource	Direction
Create PM Job	POST	/pm_jobs	EM → VNFM
Delete PM Job	DELETE	/pm_jobs/{pmJobId}	EM → VNFM
Query PM Job	GET	/pm_jobs	EM → VNFM
	GET	/pm_jobs/{pmJobId}	EM → VNFM
Create Threshold	POST	/thresholds	EM → VNFM
Delete Thresholds	DELETE	/thresholds/{thresholdId}	EM → VNFM
Query Threshold	GET	/thresholds	EM → VNFM
	GET	/thresholds/{thresholdId}	EM → VNFM
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)	VNFM → EM VNFM → VNF
NOTE: In the VNF 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.4 VNF Fault Management interface

ETSI GS NFV-IFA 008 [1] operation	HTTP method	Resource	Direction
Get Alarm List	GET	/alarms	EM → VNFM VNF → VNFM
Acknowledge Alarm	PATCH	/alarms/{alarmId}	EM → VNFM VNF → VNFM
Escalate Perceived Severity	POST	/alarms/{alarmId}/escalate	EM → VNFM VNF → VNFM
Subscribe	POST	/subscriptions	EM → VNFM VNF → VNFM
Query Subscription Information	GET	/subscriptions	EM → VNFM VNF → VNFM
	GET	/subscriptions/{subscriptionId}	EM → VNFM VNF → VNFM
Terminate subscription	DELETE	/subscriptions/{subscriptionId}	EM → VNFM VNF → VNFM
Notify	POST	(provided by API consumer)	VNFM → EM VNFM → VNF

A.5 VNF Indicator interface

ETSI GS NFV-IFA 008 [1] operation	HTTP method	Resource	Direction
Get Indicator Value	GET	/indicators	VNFM → EM VNFM → VNF
	GET	/indicators/{vnfInstanceId}	VNFM → EM
	GET	/indicators/{vnfInstanceId}/{indicatorId}	VNFM → EM
	GET	/indicators/{indicatorId}	VNFM → VNF
Subscribe	POST	/subscriptions	VNFM → EM VNFM → VNF
Query Subscription Information	GET	/subscriptions	VNFM → EM VNFM → VNF
	GET	/subscriptions/{subscriptionId}	VNFM → EM VNFM → VNF
Terminate subscription	DELETE	/subscriptions/{subscriptionId}	VNFM → EM VNFM → VNF
Notify	POST	(provided by API consumer)	EM → VNFM VNF → VNFM

A.6 VNF Configuration interface

ETSI GS NFV-IFA 008 [1] operation	HTTP method	Resource	Direction
Set Configuration	PATCH	/configuration	VNFM → VNF
	GET	/configuration	VNFM → VNF

A.7 LCM Coordination interface

ETSI GS NFV-IFA 008 [1] operation	HTTP method	Resource	Direction
Coordinate lifecycle operation	POST	/coordinations	VNFM → VNF/EM
	GET	/coordinations/{coordinationId}	VNFM → VNF/EM
	POST	/coordinations/{coordinationId}/cancel	VNFM → VNF/EM

Annex B (informative): Explanations

B.1 Introduction

This annex provides explanations of certain concepts introduced in the present document.

In clause B.2, the underlying concepts of scaling a VNF instance are explained.

In clause B.3, examples of VNF connectivity patterns and change of VNF external connectivity are provided.

B.2 Scaling of a VNF instance

A VNF instance can be scaled in the following ways:

- scale out: adding additional VNFC instances to the VNF to increase capacity
- scale in: removing VNFC instances from the VNF, in order to release unused capacity

This mechanism is called "horizontal scaling".

NOTE: Besides that, there is also "vertical scaling" which is not supported in the present document, and which includes scale up (adding further resources to existing VNFC instances) and scale down (removing resources from existing VNFC instances).

Potentially, different *aspects* of a VNF can be scaled independently. For example, a VNF could be designed to provide static capacity such as database nodes and dynamic capacity such as query processing nodes. Such a VNF might be scaled with regards to two separate aspects: the "static capacity" aspect can be scaled by adding VNFCs from VNF Deployment Units (VDUs) defining database nodes, and the "dynamic capacity" aspect can be scaled by adding VNFCs from VDUs defining query processing nodes. In complex VNF designs, scaling a VNF often requires adding/removing a number of related VNFC instances of several different types, possibly based on multiple VDUs. For example, in a high availability configuration, it might be required to add in each scaling step a pair of VNFC instances, one in active and one in standby configuration. The scaling aspects valid for a particular VNF are declared in the VNFD.

Each scaling aspect can only be scaled in discrete steps, the so-called "*scaling steps*". Each scaling step corresponds to adding or removing an *increment* (set of VNFCs based on one or more VDUs, and the related virtualised storage/virtualised network resources) to or from the VNF instance, and (re)configuring the virtualised resources. Per increment, the VNFM will figure out the necessary set of VNFCs and the related set of resources based on VNF-specific rules, for instance using the lifecycle management script associated to the Scale VNF or Scale VNF to Level event.

When scaling a VNF for a particular aspect, the number of scaling steps to apply to that aspect can be provided as a parameter. A scaling step is the smallest unit by which a particular aspect of a VNF can be scaled, and is mapped by the VNFM to the addition (or removal) of a certain set of resources. For each scaling aspect, the minimum scale level is assumed as zero, and the maximum scale level is defined in the VNFD. The maximum scale level corresponds to the maximum number of scaling steps that can be performed for this aspect, starting from the minimum scale level (i.e. zero). The maximum scale level represents the maximum configuration of that aspect of the VNF in a given deployment flavour. The minimum scale level represents the minimum configuration of that aspect of the VNF in a given deployment flavour. It usually corresponds to some deployed resources, but it is also possible to define in the VNFD that certain VDUs may not always have a corresponding VNFC instance, i.e. for certain aspects the minimum configuration may indeed be empty.

At each point in time between the completed VNF instantiation and the VNF termination, the current "size" of a particular scaling aspect of the VNF can be expressed by the current scale level w.r.t. that aspect. When the VNF is instantiated, the current scale level is initialized with values that are defined as part of the instantiation level in the VNFD for the associated aspect. Figure B.2-1 illustrates the concepts described above.

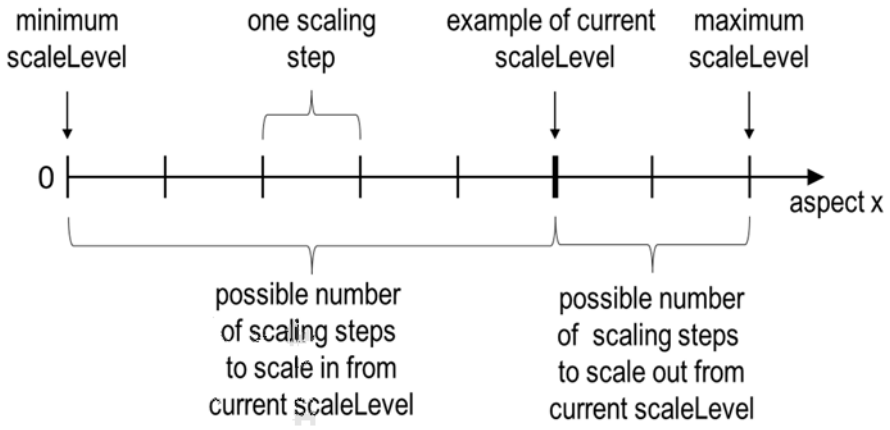

Figure B.2-1: Illustrating the concepts of scale level and scaling steps for a particular scaling aspect

As indicated above, a VNF can have one or more scaling aspects. Each individual aspect has a current scale level. All pairs of (aspect, scaleLevel) together are called the *scale status* of the VNF instance and can be obtained from the "scaleStatus" attribute of the VnfInstance structure which is returned when reading the "Individual VNF instance" resource or when querying the "VNF instances" resource. Example 1 illustrates a possible scale status.

EXAMPLE 1:

```
"scaleStatus": [
    {"aspectId": "processing", "scaleLevel": "2"},
    {"aspectId": "database", "scaleLevel": "3"}
]
```

When requesting scaling of a VNF instance, there are two methods: Scale VNF (see clause 5.4.5) and Scale VNF to Level (see clause 5.4.6). When using "Scale VNF", the scaling request defines how many increments (scaling steps) are requested to be added to or removed from the current "size" (scale level) *for a single aspect*. Depending on the VNF capabilities, single-step scaling or multiple-step scaling can be supported in a single scale request. When using "Scale VNF to Level", the scale request defines a target size of the VNF instance by defining the requested target size *for all aspects at once*, independent from the current scale status (current size) of the VNF instance. The target size can be expressed by referencing pre-defined sizes (called *instantiation levels*) declared in the VNFD, or by explicitly providing the target scale level for each scaling aspect, as illustrated in example 2.

EXAMPLE 2:

```
"scaleInfo": [
    {"aspectId": "processing", "scaleLevel": "4"},
    {"aspectId": "database", "scaleLevel": "2"}
]
```

These combinations allow four sub-modes of scaling:

- Scale VNF with a single step
- Scale VNF with multiple steps
- Scale VNF to Level based on pre-defined sizes (instantiation levels) only
- Scale VNF to Level with arbitrary sizes

B.3 Examples of VNF connectivity patterns

B.3.1 Introduction

Clause B.3.2 illustrates examples of possible connectivity patterns for a VNF. The purpose is to illustrate the relationship among the different information elements specified in clause 5.5 that are used to describe the connectivity of and within a VNF instance.

NOTE: The information related to connectivity as shown in clause B.3.2 is to be understood in the context of the present document, i.e. availability of certain information on the Ve-Vnfm reference point follows the conditions that are detailed in the respective attribute descriptions and notes in the present document.

Clause B.3.3 illustrates the use of the "Change external VNF connectivity" task resource to re-connect external CPs of a VNF instance to a different external VL.

B.3.2 Example of a VNF instance with two different types of external connection points

The present example shows a regular connectivity pattern of a VNF where the two external CPs of the VNF use different connectivity patterns. Figure B.3.2-1 illustrates the example, from which it is highlighted the following:

- An external CP of the VNF instance (see VnfExtCp #1) that maps to an internal CP, i.e. a CP of a specific VNFC.
- An external CP of the VNF instance (see VnfExtCp #2) that refers to a link port of an internal VL of the VNF (see VnfLinkPort #2.2).
- An internal VL of the VNF instance (see VnfVirtualLink #1) that is only used for connectivity of VNFCs within the VNF.
- An internal VL of the VNF instance (see VnfVirtualLink #2) that is used as provider of a link port for connectivity of external CPs of the VNF.
- Link ports of internal VLs of the VNF instance (see VnfLinkPort #1.1 to #1.3 and VnfLinkPort #2.1), which are exposed on Ve-Vnfm reference point.
- Internal CPs, i.e. CPs of specific VNFCs (see VNFC CPs), which are exposed on the Ve-Vnfm reference point.

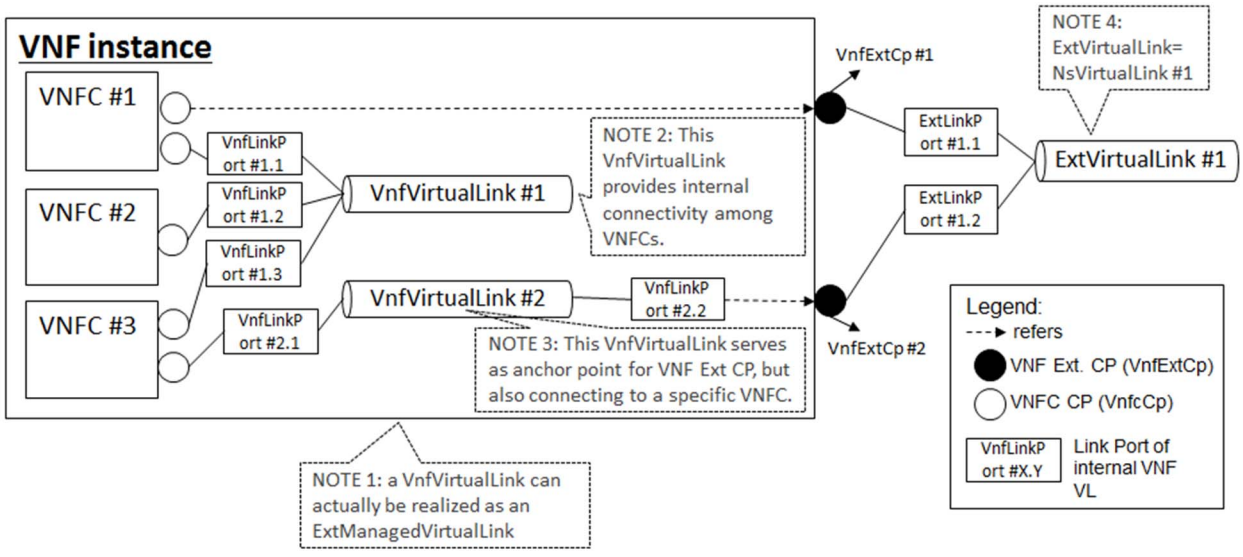

Figure B.3.2-1: Example of a VNF instance with two different types of external connection points

B.3.3 Example of changing VNF connectivity

This example illustrates changing the external connectivity of a VNF instance using the "Change external VNF connectivity" task resource (clause 5.4.11). The scenario depicted disconnects from a "source" external VL all those external CP instances that were created based on a particular CPD, and connects them to a "target" external VL.

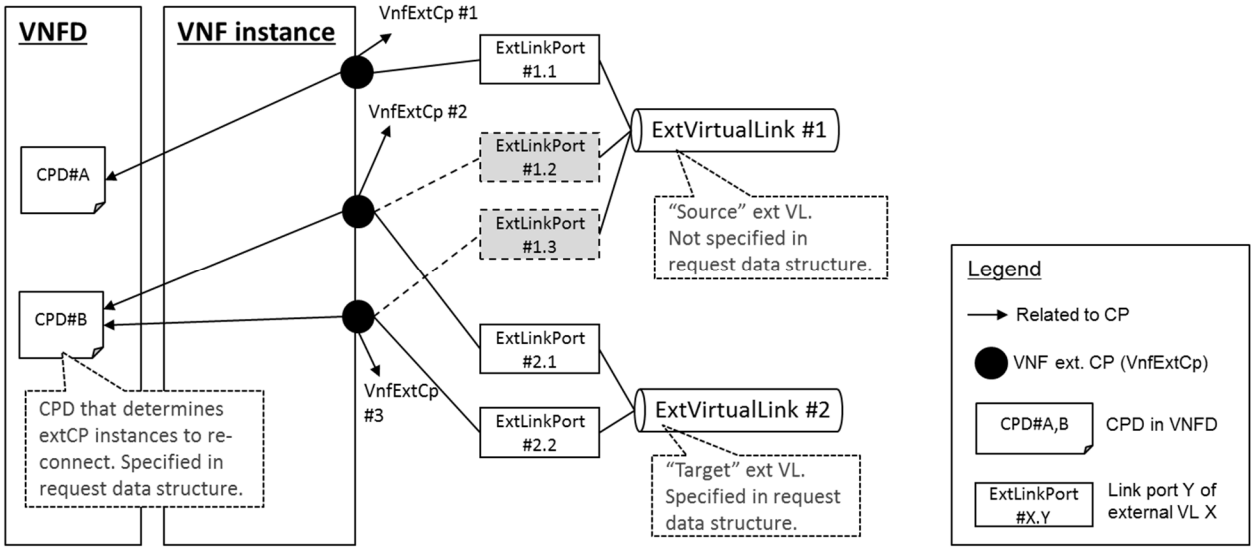

Figure B.3.3-1: Illustration of disconnecting external CPs from one external VL and connecting them to another external VL

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.4], for the Ve-Vnfm 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-SOL002.

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 to represent changes to the OpenAPI representation without changing the present document (see clause 9.1.2 of ETSI GS NFV-SOL 013 [6]).

It is specified in clause 6 of ETSI GS NFV-SOL 015 [i.9] how the OpenAPI specification references the present document and signals the version information.

Annex D (informative): Differences between ETSI GS NFV-SOL 002 and ETSI GS NFV-SOL 003

D.1 Overview

The set of APIs defined in ETSI GS NFV-SOL 002 (the present document) and ETSI GS NFV-SOL 003 [i.2] are overlapping to a large extent. A number of APIs are present in both ETSI GS NFV-SOL 002 and ETSI GS NFV-SOL 003 [i.2]. However, for each of these APIs, there are certain differences depending on whether they are exposed towards the NFVO, or towards the VNF/EM. These differences are described in clause D.2.

Other APIs are only present in one of the two specifications, as they only make sense either on the Ve-Vnfm reference point, or on the Or-Vnfm reference point. These APIs are listed in clause D.3.

D.2 Interfaces present in both ETSI GS NFV-SOL 002 and ETSI GS NFV-SOL 003

D.2.1 Basic principles

When the NFVO requests VNF management functionality from the VNFM, it addresses each VNF instance as a whole. Detailed information about VNF internals (such as VNFC instances or internal topology) is typically not needed and cannot be managed; however, the NFVO needs to understand the resource view of the VNFCs in order to perform resource orchestration. In contrast, the entity actually performing the management of a VNF instance (the EM or the VNF instance itself) needs detailed information about VNF internals, such as VNFC instances or internal topology. This difference results in more detailed VNFC-related information to be exposed in ETSI GS NFV-SOL 002 (as part of certain resource representations, notifications and request parameters) than in ETSI GS NFV-SOL 003 [i.2]. Also, certain operation modes such as graceful termination are not needed by the EM (as opposed to the NFVO), as the EM can ensure to take a VNF instance out of service before requesting termination.

D.2.2 VNF Lifecycle Management interface

Certain attributes are only available on either ETSI GS NFV-SOL 002 or ETSI GS NFV-SOL 003 [i.2], or have restrictions w.r.t. their value set. Some operations on certain resources are only available on one branch of Ve-Vnfm, i.e. either towards the VNF or towards the EM, or only available on either ETSI GS NFV-SOL 002 or ETSI GS NFV SOL 003 [i.2]. Resources and attributes with such restrictions are documented in table D.2.2-1.

Table D.2.2-1: VNF Lifecycle Management interface - differences between ETSI GS NFV-SOL 002 and ETSI GS NFV-SOL 003 [i.2]

Attribute/Resource with method	SOL 002	SOL 003
VnfInstance.instantiatedVnfInfo.vnfcInfo	present	not present
VnfInstance.vimConnectionInfo	not present	present
VnfInfoModificationRequest.vnfcInfoModifications	present	not present
VnfInfoModificationRequest.vimConnectionInfo	not present	present
VnfInfoModifications.vnfcInfoModifications	present	not present
VnfInfoModifications.vimConnectionInfo	not present	present
InstantiateVnfRequest.vimConnectionInfo	not present	present
ChangeVnfFlavourRequest.vimConnectionInfo	not present	present
ChangeExtVnfConnectivityRequest.vimConnectionInfo	not present	present
ChangeCurrentVnfPkgRequest.vimConnectionInfo	not present	present
ModificationsTriggeredByVnfPkgChange.vimConnectionInfo	not present	present
HealVnfRequest.vnfcInstanceId	present	not present
HealVnfRequest.healScript	present	not present
OperateVnfRequest.vnfcInstanceId	present	not present
VnfcResourceInfo.zoneId	not present	present
VnfVirtualLinkResourceInfo.zoneId	not present	present
VirtualStorageResourceInfo.zoneId	not present	present
AffectedVnfc.zoneId	not present	present
AffectedVnfc.resourceDefinitionId	not present	present
AffectedVirtualLink.zoneId	not present	present
AffectedVirtualLink.resourceDefinitionId	not present	present
AffectedExtLinkPort.resourceDefinitionId	not present	present
AffectedVirtualStorage.zoneId	not present	present
AffectedVirtualStorage.resourceDefinitionId	not present	present
CreateVnfSnapshotRequest.vnfcInstanceId	present	not present
VnfSnapshot.vnfStateSnapshotInfo	not present	present
VnfSnapshot._links	not present	present
CreateVnfSnapshotInfoRequest.vnfSnapshot	not present	present
RevertToVnfSnapshotRequest.vnfcInstanceId	present	not present
RevertToVnfSnapshotRequest.vnfcSnapshotInfoId	present	not present
ExtManagedVirtualLinkData.vnfLinkPort	not present	present
ExtManagedVirtualLinkData.extManagedMultisiteVirtualLinkId	not present	present
ExtManagedVirtualLinkInfo.extManagedMultisiteVirtualLinkId	not present	present
RevertToVnfSnapshotRequest.vnfSnapshotInfoId	conditional	mandatory
VnfcResourceInfo.vnfcCpInfo	mandatory	conditional
VnfVirtualLinkResourceInfo.vnfLinkPorts	mandatory	conditional
POST .../vnf_instances	available to EM only	available
POST .../vnf_instances/{vnfInstanceId}/instantiate	available to EM only	available
POST .../vnf_instances/{vnfInstanceId}/change_flavour	available to EM only	available
POST .../vnf_instances/{vnfInstanceId}/terminate	available to EM only	available
DELETE .../vnf_instances/{vnfInstanceId}	available to EM only	available
POST .../vnf_instances/{vnfInstanceId}/operate	available to EM only	available
POST .../vnf_instances/{vnfInstanceId}/change_ext_conn	available to EM only	available
PATCH .../vnf_instances/{vnfInstanceId}	available to EM only	available
PATCH .../vnf_snapshots/{vnfSnapshotInfoId}	not available	available
GET .../vnf_snapshots/{vnfSnapshotInfoId}/vnf_state_snapshot	not available	available

D.2.3 VNF Performance Management interface

The same set of resources and methods is specified in SOL002 and SOL003. However, as opposed to ETSI GS NFV-SOL 003 [i.2], information about VNFCs and VNF-internal connection points can be added to the measurements in ETSI GS NFV-SOL 002, if applicable to the actual measurement as defined in ETSI GS NFV-IFA 027 [5].

D.2.4 VNF Fault Management interface

As opposed to ETSI GS NFV-SOL 003 [i.2], information about VNFCs affected by a fault is added to the alarms in ETSI GS NFV-SOL 002. Also, it is possible for the EM to suggest escalation of the perceived severity of an alarm in ETSI GS NFV-SOL 002.

Certain attributes are only available on either ETSI GS NFV-SOL 002 or ETSI GS NFV-SOL 003 [i.2], or have restrictions w.r.t. their value set. Some operations on certain resources are only available on either ETSI GS NFV-SOL 002 or ETSI GS NFV-SOL 003 [i.2]. Resources and attributes with such restrictions are documented in table D.2.4-1.

Table D.2.4-1: VNF Fault Management interface - differences between ETSI GS NFV-SOL 002 and ETSI GS NFV-SOL 003 [i.2]

Attribute/Resource with method	SOL 002	SOL 003
Alarm.vnfcInstanceIds	available	not available
POST …/alarms/{alarmId}/escalate	available	not available

D.2.5 VNF Indicator interface

The present document specifies one more resource in addition to those specified in ETSI GS NFV-SOL 003 [i.2] for the VNF Indicator interface. This difference is summarized in table D.2.5-1.

Table D.2.5-1: VNF Indicator interface - differences between ETSI GS NFV-SOL 002 and ETSI GS NFV-SOL 003 [i.2]

Resource with method	SOL 002	SOL 003
GET …/indicators/{indicatorId}	Available when the interface is exposed by a VNF instance	Not available

Furthermore, in the present document, the API consumer of this interface is the VNFM and the API producer can be either an EM, a VNF instance or both, depending on the resource considered.

D.3 Interfaces present in one of ETSI GS NFV-SOL 002 and ETSI GS NFV-SOL 003

D.3.1 Interfaces only present in ETSI GS NFV-SOL 002

The following interfaces are only present in ETSI GS NFV-SOL 002 (the present document):

- VNF Configuration interface
- VNF LCM Coordination interface

D.3.2 Interfaces only present in ETSI GS NFV-SOL 003

The following interfaces are only present in ETSI GS NFV-SOL 003 [i.2]:

- VNF Package Management interface
- VNF Lifecycle Operation Granting
- Virtualised Resources Quota Available Notification interface
- VNF Snapshot Package Management interface

Annex E (informative): History of features added to the present document

E.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 2.8.1 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.

E.2 Features added in Release 3

E.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 E.2.1-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
5.4.11a	vnflcm	/vnf_instances/{vnfInstanceId}/change_vnfpkg	New resource
5.5.2.2	vnflcm	VnfInstance._links.changeCurrentVnfPkg	New attribute
5.5.2.11a	vnflcm	Data type: ChangeCurrentVnfPkgRequest	New resource data type
5.5.2.13	vnflcm	VnfLcmOpOcc.operationParams: new structure "ChangeCurrentVnfPkgRequest"	Modified permitted attribute values
5.5.2.13	vnflcm	VnfLcmOpOcc.modificationsTriggeredByVnfPkgChange	New attribute
5.5.2.17	vnflcm	VnfLcmOperationOccurrenceNotification Trigger "Change of current VNF package"	New trigger condition
5.5.2.17	vnflcm	VnfLcmOperationOccurrenceNotification.modificationsTriggered ByVnfPkgChange	New attribute
5.5.3.5	vnflcm	ExtManagedVirtualLinkInfo.vnfdId	New attribute
5.5.3.7	vnflcm	ScaleInfo.vnfdId	New attribute
5.5.3.8	vnflcm	VnfcResourceInfo.vnfdId	New attribute
5.5.3.9	vnflcm	VnfVirtualLinkResourceInfo.vnfdId	New attribute
5.5.3.10	vnflcm	VirtualStorageResourceInfo.vnfdId	New attribute
5.5.3.17	vnflcm	MonitoringParameter.vnfdId	New attribute
5.5.3.18	vnflcm	VnfExtCpInfo.vnfdId	New attribute
5.5.3.19	vnflcm	AffectedVnfc.vnfdId	New attribute
5.5.3.20	vnflcm	AffectedVirtualLink.vnfdId	New attribute
5.5.3.21	vnflcm	AffectedVirtualStorage.vnfdId	New attribute
5.5.4.5	vnflcm	LcmOperationType: added value CHANGE_VNFPKG	New enum value
8.4.7.3	vnfind	<<Callback URI>>(SupportedIndicatorsChangeNotification)	New notification
8.5.2.6	vnfind	SupportedIndicatorsChangeNotification	New notification
8.5.3.2	vnfind	VnfIndicatorNotificationsFilter.notificationTypes	New attribute

E.2.2 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 E.2.2-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
5.4.21	vnflcm	/vnf_instances/{vnfInstanceId}/create_snapshot	New resource
5.4.22	vnflcm	/vnf_instances/{vnfInstanceId}/revert_to_snapshot	New resource
5.4.23	vnflcm	/vnf_snapshots	New resource
5.4.24	vnflcm	/vnf_snapshots/{vnfSnapshotInfoId}	New resource
5.5.2.2	vnflcm	VnfInstance._links.createSnapshot	New attribute
5.5.2.2	vnflcm	VnfInstance._links.revertToSnapshot	New attribute
5.5.2.13	vnflcm	VnfLcmOpOcc.operationParams: new structures "CreateVnfSnapshotRequest" and "RevertToVnfSnapshotRequest"	Modified permitted attribute values
5.5.2.13	vnflcm	VnfLcmOpOcc.vnfSnapshotInfoId	New attribute
5.5.2.13	vnflcm	VnfLcmOpOcc._links.vnfSnapshot	New attribute
5.5.2.20	vnflcm	CreateVnfSnapshotInfoRequest	New resource data type
5.5.2.21	vnflcm	CreateVnfSnapshotRequest	New resource data type
5.5.2.22	vnflcm	VnfSnapshotInfo	New resource data type
5.5.2.23	vnflcm	VnfSnapshot	New resource data type
5.5.2.24	vnflcm	RevertToVnfSnapshotRequest	New resource data type
5.5.4.5	vnflcm	LcmOperationType: added values CREATE_SNAPSHOT and REVERT_TO_SNAPSHOT	New enum value

E.2.3 Additional new functionality outside the "NFV features" scheme

E.2.3.1 Trunking support

The parameters that provide external CP data have been modified to support trunking and to allow easier modification.

Table E.2.3.1-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
5.5.3.6	vnflcm	VnfExtCpData.cpConfig	Other change: turn this attribute into a map, Modified attribute semantics
5.5.3.6a	vnflcm	VnfExtCpConfig.parentCpConfigId	New attribute
5.5.3.6a	vnflcm	VnfExtCpConfig.id	Removed attribute
5.5.3.6a	vnflcm	VnfExtCpConfig.cpInstanceId	Removed attribute
5.5.3.6c	vnflcm	IpOverEthernetAddressData.segmentationType	New attribute
5.5.3.6c	vnflcm	IpOverEthernetAddressData.segmentationId	New attribute
5.5.3.3	vnflcm	ExtVirtualLinkInfo.currentVnfExtCpData	New attribute
5.5.3.8	vnflcm	VnfcResourceInfo.vnfcCpInfo.parentCpId	New attribute
5.5.3.11	vnflcm	VnfLinkPortInfo.trunkResourceId	New attribute
5.5.3.12	vnflcm	ExtLinkPortInfo.trunkResourceId	New attribute
5.5.3.12a	vnflcm	ExtLinkPortData.trunkResourceId	New attribute
5.5.3.16	vnflcm	IpOverEthernetAddressInfo.segmentationId	New attribute
5.5.3.25	vnflcm	VnfExtCpInfo.cpConfigId	New attribute

E.2.3.2 Verbosity of VNF LCM operation occurrence notifications

This change enables to control the verbosity of VNF LCM operation occurrence notifications.

Table E.2.3.2-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
5.5.2.15	vnflcm	LccnSubscriptionRequest.verbosity	New attribute
5.5.2.16	vnflcm	LccnSubscription.verbosity	New attribute
5.5.2.17	vnflcm	VnfLcmOperationOccurrenceNotification.verbosity	New attribute
5.5.2.17	vnflcm	VnfLcmOperationOccurrenceNotification	Other change: Allow to omit certain attributes depending on the value of the "verbosity" attribute

E.2.3.3 LCM coordination

LCM coordination allows an ongoing LCM operation occurrence to trigger a related management operation by the EM or VNF, to wait for its result, and to coordinate that management operation with the LCM operation occurrence. This functionality is used e.g. by FEAT02, FEAT12 and FEAT15 if such coordination is needed.

Table E.2.3.3-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
10	lcmcoord	.../{apiRoot}/lcmcoord	New API
5.5.2.13	vnflcm	VnfLcmOpOcc.lcmCoordinations	New attribute
5.5.2.13	vnflcm	VnfLcmOpOcc.rejectedLcmCoordinations	New attribute
5.5.2.13	vnflcm	VnfLcmOpOcc.warnings	New attribute

E.2.3.4 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 [i.10] for the supported use cases.

Table E.2.3.4-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
5.5.2.2	vnflcm	VnfInstance.instantiatedVnfInfo.vipCpInfo	New attribute
5.5.2.13	vnflcm	VnfLcmOpOcc.affectedVipCps	New attribute
5.5.2.17	vnflcm	VnfLcmOperationOccurrenceNotification. affectedVipCps	New attribute
5.5.3.2	vnflcm	ExtVirtualLinkData.extLinkPorts	Modified attribute semantics (added support for ports related to a VIP CP)
5.5.3.6a	vnflcm	VnfExtCpConfig.createExtLinkPort	New attribute
5.5.3.11	vnflcm	VnfLinkPortInfo.vipCpInstanceId	New attribute
5.5.3.12	vnflcm	ExtLinkPortInfo.secondaryCpInstanceId	New attribute
5.5.3.25	vnflcm	VnfExtCpInfo.associatedVipCpId	New attribute

E.2.4 FEAT12: Enhancement support for MEC in NFV deployments

This feature defines updates to selected NFV interfaces. The intent is to define extensions which can be used to specify how ETSI MEC can be deployed in an NFV environment, allowing to run MEC applications on the NFVI besides VNFs, and re-using ETSI NFV MANO components to perform common MANO tasks on the MEC applications.

Table E.2.4-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
10.6	lcmcoord	Standardized coordination actions	Other change

E.2.5 FEAT03: NFVI software modification

This feature addresses the support for the coordination of the NFVI software modification process with the VNFs hosted on the NFVI in order to minimize impact on service availability.

Table E.2.5-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
7.5.2.4	vnffm	Alarm.faultType	Modified permitted attribute values
7.5.2.4	vnffm	Alarm.probableCause	Modified permitted attribute values
7.5.2.4	vnffm	Alarm.rootCauseFaultyResource	Modified attribute semantics
7.5.2.4	vnffm	Alarm.eventTime	Modified attribute semantics
7.5.2.4	vnffm	Alarm.faultDetails	Modified attribute semantics

Annex F (informative): Change History

Version	Date	Information about changes
V0.0.1	May 2016	Skeleton and scope based on: • NFVSOL(16)000003r1_SOL002_-_Proposed_table_of_contents • NFVSOL(16)000004_SOL002_-_Proposed_scope
V0.0.2	May 2016	Implemented NFVSOL(16)000017r1
V0.0.3	April 2017	Contributions incorporated • NFVSOL(17)000073 Document structure of SOL002 • NFVSOL(17)000073 Document structure of SOL002 • NFVSOL(17)000102r1 SOL002 SOL003 Remove Annex C resp X • NFVSOL(17)000161 SOL002- 2. References • NFVSOL(17)000147r1 SOL002 SOL003 Definitions, symbols and abbreviations • NFVSOL(17)000162r1 SOL002-clause 4 General aspects • NFVSOL(17)000131r3 SOL002-VNFLCM interface based on SOL003 • NFVSOL(17)00082r3 SOL002 Additional parameters of VNF LCM • NFVSOL(17)000133r1 SOL002-vnfInstance data type • NFVSOL(17)000132r1 SOL002-VNF Indicator interface based on SOL003 • NFVSOL(17)000163 SOL002-Annex B. Mapping operations • NFVSOL(17)000121_Conventions_three_parts_of_remarks_column • NFVSOL(17)000187r1_SOL002_SOL003_Conventions_global_fix_for_normative_statement • NFVSOL(17)000150r1 SOL002 SOL003 Adding description of rollback retry cancel fail to clause 5.4.13.1 • NFVSOL(17)000149 SOL002 SOL003 LCM ed note error handling bugfix • NFVSOL(17)000125r1 SOL002 SOL003 MonitoringParameters data structure • NFVSOL(17)000141 SOL002 SOL003 Rename ind to vnfind • NFVSOL(17)000157 SOL003 SOL002 5.2.1 Fixing Vnf Instance Creation flow • NFVSOL(17)000155r1 SOL003 SOL002 resolve Auto-X editor s note • NFVSOL(17)000151r1 SOL003 SOL002 clause 5.6.1 Basic concepts (for LCM errors) • NFVSOL(17)000126r1 SOL002 SOL003 NetworkAddress data structure • NFVSOL(17)000182 SOL002 SOL003 Indicators clean up NFVSOL(17)000188 SOL002 SOL003 Notification id NFVSOL(17)000189r1 SOL002 SOL003 VnfInstanceSubscriptionFilter general data type • NFVSOL(17)000190 SOL002 SOL003 VnfLcOpOcc fixes for ModifyVnfInfo • NFVSOL(17)000191r1 SOL002 SOL003 state change timestamp and affected resources • NFVSOL(17)000192 SOL002 SOL003 Remove editor's note in clause 5.4.3.3.4 • NFVSOL(17)000199_SOL002_SOL003_Renaming_attribute_selectors • NFVSOL(17)000200_SOL002_SOL003_Attribute_filter_equality • NFVSOL(17)000078r3_SOL002_SOL003_VimId_fixes • NFVSOL(17)000209r1_SOL002_SOL003_SOL005_all_fields__and_defaults_for_selectors (1) • NFVSOL(17)000221r1_SOL002_-_Data_model_of_VNF_indicator_interface_based_on_SOL0 • NFVSOL(17)000222r1_SOL002_-_VNF_FM_interface_based_on_SOL003 • NFVSOL(17)000223r1_SOL002_-_VNF_PM_interface_based_on_SOL003

Version	Date	Information about changes
V0.1.0	May 2017	• NFVSOL(17)000373r2 SOL002 - Mirror for Authorization • NFVSOL(17)000380r1 SOL002 - Mirror for clause 5 • NFVSOL(17)000381 SOL002 - Mirror for clause 6 • NFVSOL(17)000219R4 SOL002: VNF FM Interface - New Escalate Perceived Severity Operation • NFVSOL(17)000331r2 SOL002 B.2 and 5.1: Update Mapping Operations to Protocol for Scale VNF to Level • NFVSOL(17)000368 SOL002 SOL003 4.3.3.1 adding informative to overview and example consistently • NFVSOL(17)000354r3 SOL003 SOL002 autoscale autoheal description • NFVSOL(17)000349R2 SOL003 SOL002 5.x Error code 404 if task resource not supported • NFVSOL(17)000384 SOL003 SOL002 Refactoring VNF Instance link in AlarmNotification • NFVSOL(17)000375r1 SOL003 SOL002 meaning of OperateVnf • NFVSOL(17)000374 SOL003 SOL002 Notification Authorization future proofing • NFVSOL(17)000394_SOL002_SOL003_address_comments_from_Procera_Networks • NFVSOL(17)000393 SOL002 SOL003 Move VimConnectionInfo to the correct clause • NFVSOL(17)000392r1 SOL003 SOL002 missing notification triggers • NFVSOL(17)000343 SOL002 SOL003 global Renaming of attribute filters to attribute-based filtering • NFVSOL(17)000363 SOL002 SOL003 global consistency of enum type names • NFVSOL(17)000396r2 SL002 SOL003 6.4 7.4 8.4 9.4 10.4 11.4 Addition of the note about how to retrieve the resource id • NFVSOL(17)000397 SOL002 SOL003 5.3.3 Clarification of notification flow in the figure 5.3.3.1-1 • NFVSOL(17)000399 SOL003 SOL002 changedExtVLs in LcmOpOccNotif
V0.2.0	June 2017	• NFVSOL(17)000336r1 SOL003 many Fixing some conditions • NFVSOL(17)000282_SOL003_-_Removing_normative_dependencies_on_SOL001 • NFVSOL(17)000341_SOL003_7_5_2_5_Refactoring_links_in_AlarmNotification • NFVSOL(17)000217r4_SOL003_VNF_FM_Acknowledge_Alarm_operation • NFVSOL(17)000227r1_SOL003_4_3_2_2_EN_Attribute_filters_point_in_time • NFVSOL(17)000232_SOL003_5_5_3_7_rapp_note__information_element • NFVSOL(17)000280r2_SOL003_-_Clause_4_4_2_-_5_5_3_-_MAC_and_IP_address_represent • NFVSOL(17)000293r1_SOL003_4_2_Consistency_of_URI_and_OAuth • NFVSOL(17)000298r2_SOL003_5_4_3_2_Improvement_of_resource_definition_descriptioon • NFVSOL(17)000299r1_SOL003_5_4_13_2_Improvement_of_resource_definition_description • NFVSOL(17)000300_SOL003_4_4_2_Clear_meaning_of_IdentifierLocal_type • NFVSOL(17)000301r1_SOL003_5_4_16_Add_the_supplement_to_Finally_Failed • NFVSOL(17)000302r1_SOL003_4_3_5_5_Consistency_between_4_3_5_4_and_4_3_5_5 • NFVSOL(17)000303r3_SOL003_-_Editorial_changes • NFVSOL(17)000306r1_SOL003_4_3_2_2_Filter_Spec_Fix • NFVSOL(17)000307r2_SOL003_4_3_4_3_Adding_retry-after_header_field • NFVSOL(17)000335r1_SOL003_4_3_3_2_1_Fixes_to_attribute_selector • NFVSOL(17)000355_SOL003_SOL002_Replace_entity_body_by_payload_body
V0.3.0	June 2017	• NFVSOL(17)000423 SOL002 ExtCP ExtVL fixes related to IFA discussion • NFVSOL(17)000429r1 SOL002 Implementing mirror contributions based on SOL003 • NFVSOL(17)000020r3 - SOL002 REST based Resource Design for VNF Configuration Interface
V0.4.0	June 2017	• NFVSOL(17)000471 - SOL002 Resolving differences between SOL002 and SOL003 • NFVSOL(17)000470 - SOL002 FaultyResourceInfo data type in clause 7.5.3.3 • NFVSOL(17)000469 - SOL002 Modifications on clause 6.4.5.2 • NFVSOL(17)000468 - SOL002 Implementing mirror contributions of SOL003 • NFVSOL(17)000472r2 - SOL002 Clause 9. VNF Configuration • NFVSOL(17)000021r5 - SOL002 - VNF configuration flows • NFVSOL(17)000444R1_Summary_of_differences_between_SOL002___SOL003

Version	Date	Information about changes
V0.5.0	July 2017	• NFVSOL(17)000480r3_SOL002_Alignment_with_SOL003_general_clauses_1_to_4 • NFVSOL(17)000482r1_SOL002_Annex_to_compare_SOL002_and_SOL003 (1) • NFVSOL(17)000484_SOL002_VNFC_configuration_management_in_ModifyVnfInfo • NFVSOL(17)000442r6_SOL002_Data_structure_for_VNF_configuration • NFVSOL(17)000479_SOL002_VimConnectionInfo_removal_aligned_with_IFA 008 • NFVSOL(17)000481r7_SOL002_Alignment_with_SOL003_clause_5_LCM_interface • NFVSOL(17)000483r2_SOL002_Fixing_clauses_6_and_7_PM_and_FM_interfaces • NFVSOL(17)000485r1_SOL002_Fixing_clause_8_Indicator_interface • NFVSOL(17)000486_SOL002_Alignment_with_SOL003_-_Annexes • NFVSOL(17)000487r2_SOL002_-_GET_method_for_VNF_Configuration • NFVSOL(17)000489r1_SOL002_handling_of_VNFC_in_FM_and_PM_interfaces • NFVSOL(17)000490_SOL002_2_1_4_3_Consistency_of_Range_Requests • NFVSOL(17)000491_SOL002_-_Annex_A_and_Editors_notes
V0.5.1	July 2017	• Implement additional editorial review comments received from SOL email list.
V0.6.0	July 2017	• NFVSOL(17)000503r1_SOL002_bug_vnfcInfo_wrong_level
V2.3.2	November 2017	• NFVSOL(17)000565_SOL002_SOL003_Fixing_actors_in_authorization_flows • NFVSOL(17)000589_SOL002_handling_of_VNFC_in_FM_interface_aligned_with_stage_2 • NFVSOL(17)000590r1_SOL002_handling_of_VNFC_in_PM_interface_aligned_with_stage_2 • NFVSOL(17)000593r1_SOL002_SOL003_miscellaneous_small_bugfixes.docx • NFVSOL(17)000622r2_SOL002_miscellaneuos_bugfixes • NFVSOL(18)000472r1_SOL002ed251__Add_note_to_MAJOR_version_field
V2.3.3	December 2017	• NFVSOL(17)000644r1 SOL002 remove graceful mode from OperateVnf • NFVSOL(17)000636 SOL002 VNFC CP changes in AffectedVnfc • NFVSOL(17)000667r2 SOL002 SOL003 Add description to VNF fault management interface • NFVSOL(17)000668 SOL002 SOL003 complement the description of CancelModeType • NFVSOL(17)000691r1 SOL002/003 - Use of verbal forms for the expression of provisions • NFVSOL(17)000646 SOL002 SOL003 Add resource metadata to AffectedVnfc/VirtualLink/Storage • NFVSOL(17)000670r2 SOL002 SOL003 Fixing statement for mandatory and conditional HTTP methods • NFVSOL(17)000722 SOL002 fixing VNF connectivity figure • NFVSOL(17)000669r3 SOL002 SOL003 Add cancel time out attribute • NFVSOL(17)000635r1 SOL002 SOL003 implicit changes in VnfInfoModifications • NFVSOL(17)000698 SOL002 SOL003 allow Fail operation in FAILED_TEMP • NFVSOL(17)000715r2 SOL002 SOL003 - Double Subscriptions for Notifications • NFVSOL(17)000718r2 SOL002 SOL003 sequence of requests responses and notifications • NFVSOL(17)000388r6 SOL002 ed2.4.1 and SOL003 ed 2.4.1: Authorization of API Requests and Notifications Document type should be "CR" • NFVSOL(17)000674r4 SOL002 SOL003 Authorization method negotiation • NFVSOL(17)000695r1 SOL003 Fixing normative status of notification endpoint • NFVSOL(17)000754_SOL002_SOL003_Remove_redundant_description_of_vnfConfigurabl • NFVSOL(17)000764 SOL002 align normative statements in trigger conditions mirror 734 • NFVSOL(17)000758r1 SOL002 Add clarification of ExManagedVirtualLink and ExtVirtualLink • NFVSOL(17)000671r2 SOL002/SOL003 ExtCpData changes from IFA1029r2 • NFVSOL(17)000780r1_SOL003_Fixing_leftovers_of_onboardedVnfPkg_Info_Id • NFVSOL(17)000789r2_Additional_modifications_for_SOL002_and_SOL003
V2.3.4	February 2018	Following comments raised during the approval period, it was decided to REMOVE all normative references to the IFA027 draft is this version of SOL002.
V2.4.1	February 2018	Publication
V2.4.2	March 2018	• NFVSOL(18)000036R1 SOL002 API Authorization clarification • NFVSOL(18)000086 SOL002ed251 mirror of 60r2 • NFVSOL(18)000087 SOL002ed251 mirror of 58r2
V2.4.3	March 2018	• NFVSOL(18)000118 SOL002ed251 empty collections clarification • NFVSOL(18)000120R1_SOL002_align_normative_statements_in_resource_tables

Version	Date	Information about changes
V2.4.4	June 2018	• NFVSOL(18)000165_SOL002ed251__Fix_cardinality_of_the_operationParams_attributes • NFVSOL(18)000189 SOL002ed251: Change the cardinality of the subscriptionId attribute in VNF LCM interface notifications • NFVSOL(18)000249 SOL002 Mirror of 153r6 • NFVSOL(18)000276 SOL002 - different names for virtual link descriptor ids • NFVSOL(18)000275r1 SOL002 - enhanced patch rules - deletion of array entries • NFVSOL(18)000274 SOL002 - Fix for the enhanced patch rules • NFVSOL(18)000273 SOL002 - Adding status codes • NFVSOL(18)000272 SOL002 - small fix in 400 response code description • NFVSOL(18)000271 SOL002 - MAC address optional in IpOverEthernetAddressInfo • NFVSOL(18)000270 SOL002 - Updating JSON RFC reference • NFVSOL(18)000269 SOL002-Fixing 400 response code definitions • NFVSOL(18)000268 SOL002 - Attributes selectors • NFVSOL(18)000211r1 SOL002 Normative attribute filters support
V2.4.5	July 2018	• NFVSOL(18)000258_SOL002ed251__Remove_the_current_values_of_the_MonitoringParam • NFVSOL(18)000362 Attribute filters • NFVSOL(18)000363 SOL002 small fix replace queried by read • NFVSOL(18)000364 SOL002 VnfExtCpInfo type • NFVSOL(18)000365 SOL002 Metadata for CP Ies • NFVSOL(18)000366 Link for notifications • NFVSOL(18)000367 retry • NFVSOL(18)000368 data types Number and String • NFVSOL(18)000369 Normative reference IFA027 • NFVSOL(18)000290r6 SOL002 - VNF indicator resources • NFVSOL(18)000370 Annex for OpenAPI • NFVSOL(18)000371 bugs on Attribute selector • NFVSOL(18)000372 Minor versioning • NFVSOL(18)000457_SOL002ed251__Version_management • NFVSOL(18)000459_SOL002ed251__Version_signaling • NFVSOL(18)000463_SOL002ed251__Closing_pagination_gap • NFVSOL(18)000464_SOL002ed251__Define_patch_version_number
V2.5.1	September 2018	Publication
V2.5.2	January 2019	• NFVSOL(18)000564_SOL002ed261_Mirror_of_552r5 • NFVSOL(18)000723_SOL002ed261_Mirror_of_581r2 • NFVSOL(18)000724r1_SOL002ed261_Mirror_of_584r2
V2.5.3	February 2019	• NFVSOL(19)000032_SOL002ed261_-moving_note_to_clause_6_5_2_4_ThresholdCrossed • NFVSOL(19)000033_SOL002ed261_-_declaration_of_metadata_and_extensions • NFVSOL(19)000053_SOL002ed261-_Modification_on_VnfLcmOperationOccurrenceNotification • NFVSOL(19)000060_SOL002ed261-_Normative changes for TST WG
V2.5.4	February 2019	• NFVSOL(19)000079 SOL002ed261 vnfId replacing vnfPkgId in VNF LCM interface • NFVSOL(19)000105 SOL002ed261 Update API version fields
V2.6.1	April 2019	Publication
V2.6.2	June 2019	• NFVSOL(19)000189r2_SOL002ed271_Clause_5_5_correct_mistakes_for_consistency_with SOL003 • NFVSOL(19)000217r1_SOL002ed271_-_PATCH_version_fields_of_the_interfaces

Version	Date	Information about changes
V2.6.3	November 2019	• NFVSOL(19)000333r1_SOL002ed271_Indicator_interface_optional • NFVSOL(19)000212r2_SOL002ed271_Clause_5_5_clarify_id_of_vnfcResourceInfo • NFVSOL(19)000463_SOL002_Clause_4_alignment_with_SOL003 • NFVSOL(19)000453r3_SOL002ed271_LCM_alignment_with_SOL003 • NFVSOL(19)000458r1_SOL002ed271_PM_alignment_with_SOL003 • NFVSOL(19)000495_SOL002ed271_mirror_of_change_5_in_454 • NFVSOL(19)000461_SOL002ed271_VnfConfig_alignment_with_rest_of_SOL002 • NFVSOL(19)000459r1_SOL002_FM_alignment_with_SOL003 • NFVSOL(19)000462_SOL002_Annexes_alignment_with_SOL003 • NFVSOL(19)000589r2_SOL002ed271__Mirror_588__Moving_pre_and_post-conditions_into • NFVSOL(19)000466r2_SOL002ed271_fixes_related_to_IFA027 • NFVSOL(19)000485r1_SOL002ed271_fixing_missing_VNFC_Info_in_VnfInfoModifications • NFVSOL(19)000460r1_SOL002ed271_Indicator_alignment_with_SOL003 • NFVSOL(19)000552r1_SOL002ed271_Mirror_of_331_-_Aligning_version_indication_with • NFVSOL(19)000603_SOL002ed271_PATCH_alarm_acknowledge_status • NFVSOL(19)000600_SOL002ed271_fixes_to_FM_interface • NFVSOL(19)000612_SOL002ed271_Annex_D_fixes_related_to_467 • NFVSOL(19)000582r1_SOL002ed271_mirror_of_581_fixing_the_PM_interface_wrt_subscr • NFVSOL(19)000534_SOL002ed271_Mirror_of_483_Exposing_MaxScaleLeve • NFVSOL(19)000673_SOL002ed271_Mirror_of_576_Initial_configurable_properties_va • NFVSOL(19)000680r1_SOL002ed271_mirror_of_679_adding_error_response_for_faile • NFVSOL(19)000570r1_SOL002ed271__Mirror_of_569__Support_rollback_for_failing_VNF • NFVSOL(19)000697_SOL002ed271_mirror_of_667_fixes_to_IFA_mapping_annex_related • NFVSOL(19)000691_SOL002ed271_mirror_of_328r3_Clarify_passing_of_external_conn • NFVSOL(19)000775r3 API versions for SOL002v271 • NFVSOL(19)000779r1_SOL002ed271_mirror_of_752r2_rapporteur_s_cleanup
CR on V2.6.3	November 2019	• NFVSOL(19)000661_SOL003ed271_replacing_client_by_API_producer
V2.7.1	January 2020	Publication
V3.0.0	January 2020	Contributions incorporated: • NC:NFVSOL(19)000233_SOL002ed311_FEAT15_Clause_5_Adding_snapshot_creation__resour • NC:NFVSOL(19)000339_SOL002ed311_FEAT15_Clause_5_1__5_2_and_5_3_Revert_to_snapsho • NBWCP:NFVSOL(19)000510r1_SOL002ed331_FEAT02_Mirror_of_508r4_and_618r2_Add_changeCurre • BWC:NFVSOL(19)000518_SOL002ed311_FEAT15_Mirror_of_299_-_Fixes_to_VNF_snapshot_dat • BWC:NFVSOL(19)000519_SOL002ed311_FEAT15_Mirror_of_353r2_-_Improving_snapshot_crea • BWC:NFVSOL(19)000520_SOL002ed331_FEAT15_Bug_fix_of_snapshot_resources_for_VNFC • BWC:NFVSOL(19)000811r1_SOL002ed331_FEAT15_mirroring_VNF_C_Snapshot_Pkg_mgmt_API • BWC:NFVSOL(19)000839r1_SOL002ed341_Buxfix_change_package_via_ModifyVnfInfo_conditio Editorials: • Changed Release number to 3 • Applied the convention for {apiMajorVersion} to those resources that were newly added on top of content from V 2.6.5.

Version	Date	Information about changes
V3.0.1	March 2020	Contributions incorporated: • NBWCP:NFVSOL(19)000846r1_SOL002ed331_mirror_of_845r1_Patch_semantics_of_passing_metad • BWC:NFVSOL(19)000849r3_SOL002ed331_mirror_of_619r2_FEAT02_extensions__confprops_and • BWC:NFVSOL(20)000013r2_SOL002ed331_mirror_of_12_adding_missing_extensions_and_vnfCo • BWC:NFVSOL(20)000072r1_SOL002ed331_mirror_of_71_add_missing_support_statements • BWC:NFVSOL(20)000097_SOL002ed331_Feature_annex • BWC:NFVSOL(20)000098_SOL002ed331_Editorial_correction • BWC:NFVSOL(20)000122_SOL002ed331_Correction_of_small_bugs_mirroring_from_100r1 Editorials: • Correct broken table format • Correct bugs of clause number and table number
V3.0.2	April 2020	Contributions incorporated: • BWC:NFVSOL(20)000155_SOL002ed331_mirror_of_152r1_Normative_statement_to_reject_Cr • BWC:NFVSOL(20)000128_SOL002ed331_forward_port_of_99r1_API_uniformity_wrt_graceful • NBWCP:NFVSOL(20)000180_SOL002ed331_mirror_of_178r1_Failing_Instantiate_and_ChangeFl • BWC:NFVSOL(20)000287_SOL002ed331_FEAT15_Moving_VNF_snapshot_package_mgmt Editorials: • Correct bug of history. • Added CR BWC/NBWC classification
V3.0.3	April 2020	Contributions incorporated: • BWC:NFVSOL(20)000177_SOL002ed331_mirror_of_159_Short_LcmOpOccNotifications • BWC:NFVSOL(20)000212r1_SOL002ed331_Support_of_Trunking • BWC:NFVSOL(20)000265r1_SOL002ed331_mirror_of_264r1_followup_on_PM_refactoring • BWC:NFVSOL(20)000289_SOL002ed331_Forward_mirror_of_249r1_SOL016_review_alignments • BWC:NFVSOL(20)000316_SOL002ed331_Forward_mirror_of_315_Fixing_notifying_informati
V3.0.4	May 2020	Contributions incorporated: • BWC:NFVSOL(20)000129r1_SOL002ed331_FEAT15_EN_resolution_bulk_mirrors • BWC:NFVSOL(20)000261_SOL002ed331_mirror_of_260r1_Indicator_changes_triggered_by_c • BWC:NFVSOL(20)000342r1_SOL002ed331_mirror_of_860r2_110r1_Notification_callback_URI_ • BWC:NFVSOL(20)000363r1_SOL002ed331_Forward_mirror_of_294_Guidelines_link_ports_noti • BWC:NFVSOL(20)000370_SOL002ed331_mirror_of_349_adding_PM_job_id_to_notification Editorials: • Aligning the description of "extensions" and "vnfConfigurableProperties" in tables 5.5.2.4-1, 5.5.2.7-1, 5.5.2.11a-1 • Changed "inline" to "inlined" • Fixed clause number and font style

Version	Date	Information about changes
V3.0.5	May 2020	Contributions incorporated: • BWC:NFVSOL(20)000282r1_SOL002ed331_mirror_of_42r5_VnfcResourceInfo_bugfix • BWC:NFVSOL(20)000329r1_SOL002ed331_mirror_of_328_VNFC_instance_related_fixes • BWC:NFVSOL(20)000417r1_SOL002ed331_VnfExtCpData_EN_resolution • BWC:NFVSOL(20)000440_SOL002ed331_mirror_of_439_FEAT02_address_ENs_on_metadata_dur • BWC:NFVSOL(20)000480r6_SOL002ed331_Nokia_review_comments • BWC:NFVSOL(20)000488_SOL002ed331_mirror_of_444_Fixing_EN_on_ModificationsTriggere • BWC:NFVSOL(20)000491r2_SOL002ed331_fixing_ENs_in_Annex_E • BWC:NFVSOL(20)000494_SOL002ed331_Update_Annex_A__Mapping_operations • BWC:NFVSOL(20)000495r1_SOL002ed331_Update_Annex_D__Difference_between_SOL002_and_SO • BWC:NFVSOL(20)000496r1_SOL002ed331_API_version Editorials: • Aligning the meaning of "Individual threshold" in Table 6.2-1 • Fixed data type and cardinality were written in reverse in Table 5.5.3.11-1 • Fixed bug of attribute name in Table 5.5.3.26-1 when implementing NFVSOL(20)000129r1 • Fixed typos and font style
V3.0.6	June 2020	Contributions incorporated: • NWCP:NFVSOL(20)000211r1_SOL002ed331_mirror_of_210r1_ChangeExtVnfConnectivity_using_p • BWC:NFVSOL(20)000450r1_SOL002ed331_mirror_of_345_Addressing_ENs_and_RNs • BWC:NFVSOL(20)000522_SOL002ed331_FEAT15_mirror_of_521r1_extending_the_description • BWC:NFVSOL(20)000530r2_SOL002ed331_Resolve_miscellaneous_inconsistencies • BWC:NFVSOL(20)000546r1_SOL002ed331_Rapporteur_s_clean-up • BWC:NFVSOL(20)000554r2_SOL002ed331_Aligning_SOL002_V030006r1_with_SOL003 Editorials: • Fixed style and clause number of Annex E. • Fixed "New type" of Type of change in Annex E to "New resource data type" • Global change, i.e. "JSON Merge Patch" • Fixed font style and table format • Changed bullet order to align with IFA 008 in clause 5.5.2.17
V3.3.1	August 2020	Version update for publication
V3.3.2	October 2020	Contributions incorporated: • BWC: NFVSOL(20)000688r1_SOL002ed341_Mirror_of_686_FEAT15_Clarifications_about_snapsh • BWC: NFVSOL(20)000698r1_SOL002ed341_FEAT15_Improve_the_description_in_VNF_LCM • BWC: NFVSOL(20)000715r1_SOL002ed341_Mirror_of_714_Fixing_of_the_flows_of_updating_the_callback_URI_of_the_PM_interface
V3.3.3	January 2021	Contributions incorporated: • BWC: NFVSOL(20)000710_SOL002ed351_mirror_of_709r2_VipCp_related_changes_from_IFA_C • BWC: NFVSOL(20)000748r6_SOL002ed351_Adding_Trunk_Logical_Topology_between_VNFC_CPs • BWC: NFVSOL(20)000790_SOL002ed351_mirror_of_489_fix_identifier_datatypes_in_VnfExt Editorials: • Year changed to 2021

Version	Date	Information about changes
V3.3.4	March 2021	Contributions incorporated: • BWC: NFVSOL(20)000780r3_SOL002ed351_mirror_of_778_extManagedVirtualLinkInfo_clarific • BWC: NFVSOL(21)000048_SOL002ed351_lcmcoord_API_resource_structure • BWC: NFVSOL(21)000049r1_SOL002ed351_lcmcoord_API_resources_details • BWC: NFVSOL(21)000050r3_SOL002ed351_lcmcoord_API_data_structures • BWC: NFVSOL(21)000051r1_SOL002ed351_lcmcoord_API_hooks_with_other_GS_parts • BWC: NFVSOL(21)000052r1_SOL002ed351_lcmcoord_API_LcmOpOcc_additions • BWC: NFVSOL(21)000065r2_SOL002ed351_lcmcoord_API_flows • BWC: NFVSOL(21)000072r2_SOL002ed351_mirror_of_71_update_FEAT_Annex_list_reading_Ext • NBWC: NFVSOL(21)000075_SOL002ed351_mirror_of_74_add_ModificationsTriggeredByVnfPkgC • BWC: NFVSOL(21)000091r1_SOL002ed351_mirror_of_90_warnings_in_LcmOpOcc • BWC: NFVSOL(21)000135r1_SOL002ed351_lcmcoord_API_addressing_ENs
V3.3.5	April 2021	Contributions incorporated: • BWC: NFVSOL(21)000156r2_SOL002ed351_standardized_coordination_actions Editorials: • Removed the extra "201 Created" in Table 10.4.2.3.1-2.
V3.3.6	April 2021	Contributions incorporated: • BWC:NFVSOL(21)000184r1_SOL002ed351_mirror_of_183r2_notification_delivery_clarificat • BWC:NFVSOL(21)000191r3_SOL002ed351_conventions_for_coordination_action_names • BWC:NFVSOL(21)000250_SOL002ed351_Mirror_of_207__Handling_of_security_sensitive_pr
V3.3.7	May 2021	Contributions incorporated: • BWC:NFVSOL(21)000182r5_SOL002ed351_FEAT02_Add_enumeration_values_of_LcmCoordResultT • BWC:NFVSOL(21)000293r1_SOL002ed351_Updating_API_versions_and_Feature_Annex • BWC:NFVSOL(21)000307r1_SOL002ed351_mirror_of_306_Nokia_review_comment Editorials: • Fixed the typo of data type "KeyValuePairs" of "metadata in Table 5.5.3.28-1.
V3.3.8	May 2021	Contributions incorporated: • BWC: NFVSOL(21)000332_SOL002ed351_mirror_of_331_ChangeCurrentVnfPkg_related_termin
V3.5.1	July 2021	Version update for publication
V3.5.2	September 2021	Contributions incorporated: • BWC:NFVSOL(21)000416_SOL002ed361_mirror_of_414_Use_of_old_assets_after_ChgCurrent • BWC:NFVSOL(21)000420r3_SOL002ed361_mirror_of_418_and_476r1 • BWC:NFVSOL(21)000443r1_SOL002ed361_Editorial_fix_on_the_API_version_of_VNF_indicato
V3.5.3	December 2021	Contributions incorporated: • BWC:NFVSOL(21)000559_SOL002ed361_mirror_of_557r2_vnfdId_in_resource_info_elements • BWC:NFVSOL(21)000576r5_SOL002ed361_FEAT03_considering_the_case_of_NFVI_operation_an • BWC:NFVSOL(21)000598_SOL002ed361_Mirror_of_596_changeCurrentVnfPkg_bugfix • BWC:NFVSOL(21)000613_SOL002ed361_mirror_of_585_cpConfigId_clarifications • BWC:NFVSOL(21)000633_SOL002ed361_mirror_of_631r1_CP_configuration_information_cla • BWC:NFVSOL(21)000662_SOL002ed361_mirror_of_660_AffectedExtLinkPort_bugfix

Version	Date	Information about changes
V3.5.4	December 2021	Contributions incorporated: • BWC:NFVSOL(21)000665r1_SOL002ed361_Updating_API_versions_and_Feature_Annex • BWC:NFVSOL(21)000678_SOL002ed361_FEAT03_fix_faultType_value_name_related_to_OAM

History

Document history		
V3.3.1	August 2020	Publication
V3.5.1	July 2021	Publication
V3.6.1	February 2022	Publication