ETSI GR NFV-IFA 051 V5.1.1 (2023-10)

GROUP REPORT

Network Functions Virtualisation (NFV) Release 5; Architectural framework; Report on VNF management gap analysis with open source projects

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

DGR/NFV-IFA051

Keywords

gap anaylsis, open source, VNF

ETSI

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

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

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

Important notice

The present document can be downloaded from:
https://www.etsi.org/standards-search

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

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

If you find a security vulnerability in the present document, please report it through our Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

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

Copyright Notification

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

Modal verbs terminology

In the present document "**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 performs a study on gap analysis on VNF management between ETSI NFV specifications and the related open source projects, which includes:

- Analysis of differences between VNFD model and the model used in open source projects, such as but not limited to native K8S Helm model, ONAP ASD.
- Analysis of differences between VNF LCM, PM, FM and other management aspects operations and the APIs used in open source projects.
- Recommendations on the normative work for improving VNFD information model and VNF LCM, PM, FM and other management aspects operations if needed based on the above analyses.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

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

[i.2] ETSI GS NFV 006: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Architectural Framework Specification".

[i.3] ETSI GS NFV-IFA 011 (V4.3.1): "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; VNF Descriptor and Packaging Specification".

[i.4] ETSI GS NFV-IFA 007 (V4.3.1): "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification".

[i.5] ETSI GS NFV-IFA 008 (V4.3.1): "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification".

[i.6] Application Service Descriptor (ASD) onboarding Information Model, version 1.0.

[i.7] ETSI GS NFV-SOL 001 (V4.3.1): "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; NFV descriptors based on TOSCA specification".

[i.8] ETSI GS NFV-SOL 003 (V4.3.1): "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; RESTful protocols specification for the Or-Vnfm Reference Point".

[i.9] ETSI OpenStack® Tacker: "VNF Descriptor (VNFD) based on ETSI GS NFV-SOL 001".

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

[i.10] Kubernetes® API v1.24.

[i.11] ETSI OpenStack® Tacker: "Virtualized Network Function Lifecycle Management Interface (VNF LCM) v2".

NOTE: VNF LCM API based on ETSI GS NFV-SOL 003 (V3.3.1).

[i.12] ETSI GS NFV-SOL 002 (V4.3.1): "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; RESTful protocols specification for the Ve-Vnfm Reference Point".

[i.13] ETSI GS NFV-SOL 003 (V3.3.1): "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; RESTful protocols specification for the Or-Vnfm Reference Point".

[i.14] AS LCM RESTful Protocols for SO CNF Manager.

NOTE: It belongs to ASD-Based CNF Orchestration PoC only.

[i.15] ETSI GS NFV-TST 010: "Network Functions Virtualisation (NFV) Release 3; Testing; API Conformance Testing Specification".

[i.16] OpenStack® Tacker: "Virtualized Network Function Performance Management Interface (VNF PM) v2".

[i.17] ETSI GS NFV-SOL 013 (V4.3.1): "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; Specification of common aspects for RESTful NFV MANO APIs".

[i.18] OpenStack® Tacker: "Virtualized Network Function Fault Management Interface (VNF FM) v1".

NOTE: VNF FM API based on ETSI GS NFV-SOL 003 (V3.3.1).

[i.19] OpenStack® Tacker: "Tracker Resources".

NOTE: Grant based on ETSI GS NFV-SOL 003 V3.3.1.

[i.20] OpenStack® Tacker: "ETSI NFV-SOL Tacker Use Cases".

[i.21] ETSI GS NFV-SOL 004 (V4.3.1): "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; VNF Package and PNFD Archive specification".

[i.22] ETSI GR NFV-IFA 029 (V3.3.1): "Network Functions Virtualisation (NFV) Release 3; Architecture; Report on the Enhancements of the NFV architecture towards "Cloud-native" and "PaaS"".

[i.23] ETSI GS NFV-IFA 010 (V4.3.1): "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Functional requirements specification".

[i.24] ETSI GS NFV-IFA 031 (V4.3.1): "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Requirements and interfaces specification for management of NFV-MANO".

[i.25] TOSCA-Simple-Profile-yaml-v1.2.

[i.26] ETSI GS NFV-IFA 040: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Requirements for service interfaces and object model for OS container management and orchestration specification".

[i.27] ETSI GS NFV-SOL 018: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; Profiling specification of protocol and data model solutions for OS Container management and orchestration".

[i.28] ETSI GR NFV-EVE 022: "Network Functions Virtualisation (NFV) Release 5; Architectural Framework; Report on VNF configuration".

[i.29] ETS GS NFV-SOL 005:" Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point".

[i.30] ETS GS NFV-SOL 014: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; YAML data model specification for descriptor-based virtualised resource management".

[i.31] ETSI GS NFV-SOL 002: "Network Functions Virtualisation (NFV); Protocols and Data Models; RESTful protocols specification for the Ve-Vnfm Reference Point".

[i.32] ETSI GS NFV-SOL 002 (V3.3.1): " Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; RESTful protocols specification for the Ve-Vnfm Reference Point.

[i.33] ETSI GS NFV-SOL 001 (V2.6.1): " Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; NFV descriptors based on TOSCA specification".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GR NFV 003 [i.1] apply.

NOTE: A term defined in the present document takes precedence over the definition of the same term, if any, in ETSI GR NFV 003 [i.1].

3.2 Symbols

Void.

3.3 Abbreviations

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

AS Application Service

NOTE: As referred in ONAP [i.14].

ASD Application Service Descriptor

NOTE: As referred in ONAP [i.14].

CNF Cloud Native Network Function

NOTE: As referred in ONAP [i.14].

4 Introduction of VNF management as defined in ETSI NFV

As described in ETSI GS NFV 006 [i.2], VNF is managed by the VNFM and has an associated VNFD which provides deployment and operational information to manage its lifecycle.

The VNFD information element is defined in ETSI GS NFV-IFA 011 [i.3], which contains Virtualised Deployment Units (VDUs), internal virtual link descriptors, external connection point descriptors, software image descriptors, and deployment flavour descriptors and etc. The Virtualisation Deployment Unit (VDU) is a construct supporting the description of the deployment and operational behaviour of a VNFC. A VNFC instance created based on the VDU maps to a single instance of atomic deployable unit, represented by a single VM for hypervisor-based virtualisation, or represented by one or a set of OS containers for OS virtualisation.

The related VNF management interfaces are defined in ETSI GS NFV-IFA 007 [i.4] and ETSI GS NFV-IFA 008 [i.5], which include VNF Lifecycle management (LCM) interface, Performance Management (PM) interface, Fault Management (FM) interface and VNF Indicator interface.

5 Analysis with related work in open source projects

5.1 Overview

Currently, some open source projects are also involved in network function management and their solution is introduced as an alternative to VNF management in ETSI NFV standards, and in some other cases as an ETSI NFV compliant solution.

The present document mainly analyses the open source projects including ONAP ASD, OpenStack® Tacker and Kubernetes®, and provides the gap analysis related to VNFD, VNF LCM, PM, and FM. Based on the analysis results, recommendation on improving ETSI NFV standards are also provided.

5.2 Gap analysis with ONAP ASD

5.2.1 Introduction of ONAP ASD

In ONAP Jakarta release, Application Service Descriptor (ASD) is proposed as a new, descriptor for containerized cloud native deployments. It contains the bare minimum information, which does not duplicate attributes that might be instead extracted from the Helm Charts. This helps maintain the principle that Helm Charts are the primary deployment artifact for a containerized application and avoids any possible source of error or confusion that such duplication could cause.

5.2.2 Gap analysis

5.2.2.1 Gap analysis related to VNFD

5.2.2.1.1 Introduction of the related work

ASD model as described in [i.6] can describe a complete application/NF, or parts of application/NF. It contains the information to prepare deployment as well as the pointers to cloud-native artifacts for LCM.

5.2.2.1.2 Comparison and analysis of VNFD and ASD information model

Table 5.2.2.1.2-1 illustrates a comparison of the attributes in VNFD as specified in clause 7.1.2 in ETSI GS NFV-IFA 011 [i.3] and ASD information model as specified in [i.6].

Table 5.2.2.1.2-1: Comparison of VNFD and ASD

Attributes of Vnfd information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of ASD information element		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
vnfdId	1	asdId	1	
vnfdExtInvariantId	0..1			No correspondence
vnfProvider	1	asdProvider	1	
vnfProductName	1	asdApplicationName	1	
vnfSoftwareVersion	1	asdApplicationVersion	1	
vnfdVersion	1	asdVersion	1	
vnfProductInfoName	0..1	asdApplicationInfo Name	0..1	
vnfProductInfoDescription	0..1	asdInfoDescription	0..1	
		asdSchemaVersion	1	See note 2.
vnfmInfo	1..N			No correspondence
localizationLanguage	0..N			No correspondence
defaultLocalizationLanguage	0..1			No correspondence
vdu	1..N			No correspondence. See note 1.
virtualComputeDesc	0..N			No correspondence. See note 1.
virtualStorageDesc	0..N			No correspondence. See note 1.
osContainerDesc	0..N			No correspondence. See note 1.
swImageDesc	0..N			No correspondence. See note 1.
intVirtualLinkDesc	0..N			No correspondence.
securityGroupRule	0..N			No correspondence.
vnfExtCpd	1..N	asdExtCpd	0..N	There is a potential synergy between asdExtCpd and vnfExtCpd. See detailed mapping in clause 5.2.2.1.3.
vipCpd	0..N			No correspondence.
virtualCpd	0..N			No correspondence.
deploymentFlavour	1..N			No correspondence.
configurableProperties	0..1			No correspondence.
modifiableAttributes	0..1			No correspondence.
lifeCycleManagementScript	0..N			No correspondence.
vnfIndicator	0..N			No correspondence.
autoScale	0..N			No correspondence.
vnfPackageChangeInfo	0..N			No correspondence.
lcmOperationCoordination	0..N			No correspondence.
		enhancedClusterCapabilities	0..1	No correspondence.
mciopId	0..N	deploymentItems	1..N	deploymentItems in ASD information element could map to mciopId and MciopProfile (in VnfDf) in Vnfd information element. See detailed mapping in clause 5.2.2.1.4.
NOTE 1: Resource related information is present in the DeploymentItem (e.g. Helm chart) in ASD. In VNFD case, this information is present in VNFD as well as in the Helm Chart. NOTE 2: There is a potential mapping of asdSchemaVersion to template_version as described in clause B.2 in ETSI GS NFV-SOL 001 [i.7].				

5.2.2.1.3 Comparison and analysis of vnfExtCpd and asdExtCpd

Table 5.2.2.1.3-1 illustrates a comparison of the attributes in vnfExtCpd as specified in clause 7.1.3 in ETSI GS NFV-IFA 011 [i.3] and asdExtCpd information model as specified in [i.6].

Table 5.2.2.1.3-1: Comparison of vnfExtCpd and asdExtCpd

Attributes of vnfExtCpd information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of asdExtCpd information element		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
intVirtualLinkDesc	0..1			No correspondence.
intCpd	0..1			No correspondence.
vipCpd	0..1			No correspondence.
virtualCpd	0..1			No correspondence.
virtualNetworkInterfaceRequirements	0..N	networkInterfaceRealizationRequirements	0..1	nicOptions, interfaceType, interfaceRedundancy and interfaceOptions in networkInterfaceRealizationRequirements in ASD has already been registered in ETSI registry, see https://register.etsi.org.
(inherited attributes in Cpd)				
cpdId	1	id	1	
layerProtocol	1..N			No correspondence.
cpRole	0..1			No correspondence.
description	0..1	description	1	
cpProtocol	0..N	networkInterfaceRealizationRequirements	0..1	networkInterfaceRealizationRequirements→ipam maps to Cpd→cpProtocol→addressData→L3AddressData→ipAddressAssignment and ipAddressAssignmentSubtype.
trunkMode	0..1	networkInterfaceRealizationRequirements	0..1	networkInterfaceRealizationRequirements→trunkMode maps to trunkMode in Cpd.
securityGroupRuleId	0..N			No correspondence.
		virtualLinkRequirement	1..N	The virtualLinkRequirement in asdExtCpd can map to external_virtual_link in VnfExtCp node type definition as specified in clause 6.8.2.6 in ETSI GS NFV-SOL 001 [i.7].
		inputParamMappings	0..1	No correspondence.
		resourceMapping	0..1	No correspondence.

5.2.2.1.4 Comparison and analysis of mciopProfile and deploymentItems

Table 5.2.2.1.4-1 illustrates a comparison of the attributes in mciopProfile as specified in clause 7.1.8.20 in ETSI GS NFV-IFA 011 [i.3] and deploymentItems information element as specified in [i.6].

Table 5.2.2.1.4-1: Comparison of mciopProfile and deploymentItems

Attributes of mciopProfile information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of deploymentItems information element		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
mciopId	1	deploymentItemId	1	
deploymentOrder	0..1	deploymentOrder	0..1	
affinityOrAntiAffinityGroupId	0..N			No correspondence.
associatedVdu	0..N			No correspondence.
		artifactType	1	artifactType in ASD is to specify the artifact type of the deployment Item, allowed values are "helm_chart", "helmfile", "crd", "terraform". The similar concept is defined in the stage 3 specification, see HelmChart artifact type as defined in ETSI GS NFV-SOL 001 [i.7].
		artifactId	1	artifactId in ASD describes the reference to the deployment artifact, it can be refer to URI or file path. The similar concept is defined in the stage 3 specification, see file attribute in the artifacts of opendb_mciop node template in clause A.18 in ETSI GS NFV-SOL 001 [i.7].
		lifecycleParameters	0..N	No correspondence.

5.2.2.2 Gap analysis related to VNF LCM

5.2.2.2.1 Introduction of the related work

The lifecycle management interface as defined in [i.14] allows the CNF Manager (CNFM in ONAP) client to invoke lifecycle management operations of AS instance towards the CNFM. The operations as provided through this interface are:

- Create AS Identifier
- Instantiate AS
- Delete AS Identifier
- Terminate AS

5.2.2.2.2 Comparison and analysis of Create VNF Identifier and Create AS identifier operation

Table 5.2.2.1.2-1 illustrates a comparison of the attributes in Create VNF Identifier operation as specified in clause 7.2.2 in ETSI GS NFV-IFA 007 [i.4] and Create AS identifier as specified in [i.11].

Table 5.2.2.2.2-1: Comparison of input parameters in Create VNF Identifier and Create AS identifier operation

Input Parameters in Create VNF Identifier operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Create AS identifier		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
vnfdId	1	asdId	1	
vnfInstanceName	0..1	asInstanceName	0..1	
vnfInstanceDescription	0..1	asInstanceDescription	0..1	
metadata	0..N			No correspondence
		additionalParams	0..1	No correspondence

Table 5.2.2.2.2-2: Comparison of output parameters returned by Create VNF Identifier and Create AS identifier operation

Output Parameters returned by Create VNF Identifier operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Create AS identifier response		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
vnfInstanceId	1	AsInstance	1	

5.2.2.2.3 Comparison and analysis of Instantiate VNF and Instantiate AS operation

Table 5.2.2.2.3-1 illustrates a comparison of the attributes in Instantiate VNF operation as specified in clause 7.2.3 in ETSI GS NFV-IFA 007 [i.4] and Instantiate AS operation as specified in [i.14].

Table 5.2.2.2.3-1: Comparison of input parameters in Instantiate VNF and Instantiate AS operation

Input Parameters in Instantiate VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in InstantiateAsRequest		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
vnfInstanceId	1			Maps to asInstanceId in the REST Interface, i.e. POST .../as_instances/{asInstanceId}/instantiate (InstantiateAsRequest)
flavourId	1			No correspondence
		deploymentItems	1..N	No correspondence
instantiationLevelId	0..1			No correspondence
targetScaleLevelInfo	0..N			No correspondence
extVirtualLink	0..N	asdExtCpdInputParams	0..N	See note
extManagedVirtualLink	0..N			No correspondence
vimConnectionInfo	0..N			No correspondence
localizationLanguage	0..1			No correspondence
additionalParam	0..N	additionalParams	0..1	
extension	0..N			No correspondence
vnfConfigurableProperty	0..N			No correspondence
NOTE: asdExtCpdInputParams can map to VnfExtCpConfig in extVirtualLink. See detailed mapping in Table 5.2.2.2.3-2.				

Table 5.2.2.2.3-2: Comparison of VnfExtCpConfig and asdExtCpdInputParams

Attributes of VnfExtCpConfig in ETSI GS NFV-IFA 007 [i.4]		Attributes of asdExtCpdInputParams		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
cpInstanceId	0..1			No correspondence
linkPortId	0..1			No correspondence
createExtLinkPort	0..1			No correspondence
netAttDefResourceId	0..N	nadNames	0..N	
cpProtocolData	0..N	loadbalancerIP	0..1	See loadBalancerIp in VirtualCpAddressData in CpProtocolData data type as defined in ETSI GS NFV-SOL 003 [i.13]
		externalIPs	0..N	No correspondence
		nadNamespace	0..1	See note
NOTE: nadNamespace in asdExtCpdInputParams can map to containerNamespace in resourceHandle in NetAttDefResourceData referenced by netAttDefResourceId in VnfExtCpConfig.				

Table 5.2.2.2.3-3: Comparison of output parameters returned by Instantiate VNF and Instantiate AS operation

Output Parameters returned by Instantiate VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in InstantiateAsRequest response		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
lifecycleOperationOccurrenceId	1			No correspondence

5.2.2.2.4 Comparison and analysis of Terminate VNF and Terminate AS operation

Table 5.2.2.2.4-1 illustrates a comparison of the attributes in Terminate VNF operation as specified in clause 7.2.7 in ETSI GS NFV-IFA 007 [i.4] and Terminate AS operation as specified in [i.14].

Table 5.2.2.2.4-1: Comparison of input parameters in Terminate VNF and Terminate AS operation

Input Parameters in Terminate VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Terminate AS		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
vnfInstanceId	1			Maps to asInstanceId in the REST Interface, i.e. POST .../as_instances/{asInstanceId}/terminate (TerminateAsRequest)
terminationType	1	terminationType	1	
gracefulTerminationTimeout	0..1	gracefulTerminationTimeout	0..1	
additionalParam	0..N	additionalParams	0..1	

Table 5.2.2.2.4-2: Comparison of output parameters returned by Terminate VNF and Terminate AS operation

Output Parameters returned by Terminate VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Terminate AS response		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
lifecycleOperationOccurrenceId	1			No correspondence

5.2.2.2.5 Comparison and analysis of Delete VNF Identifier and Delete AS Identifier operation

Table 5.2.2.2.5-1 illustrates a comparison of the attributes in Delete VNF Identifier as specified in clause 7.2.8 in ETSI GS NFV-IFA 007 [i.4] and Delete AS Identifier operation as specified in [i.14].

Table 5.2.2.2.5-1: Comparison of input parameters in Delete VNF Identifier and Delete AS Identifier operation

Input Parameters in Delete VNF Identifier operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Delete AS Identifier		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
vnfInstanceId	1			Maps to asInstanceId in the REST Interface, i.e. DELETE .../as_instances/{asInstanceId}

Table 5.2.2.2.5-2: Comparison of output parameters returned by Delete VNF Identifier and Delete AS Identifier operation

Output Parameters returned by Delete VNF Identifier operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Delete AS Identifier response		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
No output parameter.		No output parameter.		

5.2.2.3 Gap analysis related to VNF performance management

5.2.2.3.1 Introduction of the related work

ONAP ASD open source project [i.14] does not specify any performance management related work.

5.2.2.3.2 Comparison and analysis

ONAP ASD open source project [i.14] does not specify any performance management related work, while in ETSI GS NFV-IFA 007 [i.4] and ETSI GS NFV-IFA 008 [i.5], VNF performance management interface is defined, which includes Create PM Job, Delete PM Jobs, Subscribe, Notify, Query PM Job, Create Threshold, Delete Thresholds, Query Threshold, Terminate Subscription, Query Subscription Info operations.

5.2.2.4 Gap analysis related to VNF fault management

5.2.2.4.1 Introduction of the related work

ONAP ASD open source project does not specify any fault management related work.

5.2.2.4.2 Comparison and analysis

ONAP ASD open source project does not specify any fault management related work, while in ETSI GS NFV-IFA 007 [i.4] and ETSI GS NFV-IFA 008 [i.5], VNF Fault management interface is defined, which includes Subscribe, Notify, Get Alarm List, Terminate Subscription, Query Subscription Info and Acknowledge alarms operations.

5.2.2.5 Gap analysis related to Grant

5.2.2.5.1 Introduction of the related work

In ONAP ASD, Grant exchange is not defined between orchestrator (e.g. SO in ONAP) and CNFM.

5.2.2.5.2 Comparison and analysis

ONAP ASD open source project does not specify Grant operation, while in ETSI GS NFV-IFA 007 [i.4], Grant VNF lifecycle operation is defined.

NOTE: In ONAP ASD, in order to deploy a CNF, CNFM can request placement control from other entities (e.g. OOF In ONAP) based on the outputs from the processing of ASD. This functionality is similar to the tasks that are performed during the granting in ETSI NFV.

5.2.2.6 Gaps and Potential solutions

5.2.2.6.1 Gap# 1: gap related to resource related information in VNFD

5.2.2.6.1.1 Introduction

This gap is identified in Table 5.2.2.1.2-1 comparison of VNFD and ASD, in which it shows that vdu is a mandatory attribute in VNFD, but there is no corresponding attribute in ASD. In VNFD case, virtualized resource related information, such as vdu, virtualComputeDesc, osContainerDesc, virtualStorageDesc and swImageDesc, is present in VNFD as well as in the MCIOP (e.g. Helm chart). However in the ASD case, virtualized resource related information is only included in the deploymentItems (e.g. Helm chart) in ASD.

5.2.2.6.1.2 Potential Solution SOL 1-1a

To resolve the gap as stated above, a simplified VNFD is proposed, in which the virtualised resource information is only contained in the MCIOP to avoid conflict, other information element such as, virtualComputeDesc, osContainerDesc, virtualStorageDesc and swImageDesc can be absent in the VNFD.

NOTE: The Vnfd information model as specified in ETSI GS NFV-IFA 011 [i.3] can include the simplified VNFD model by defining optional attributes of virtualComputeDesc, osContainerDesc, virtualStorageDesc and swImageDesc. In case the above attributes exist (cardinality other than 0), the virtualised resource information still refers to the description in these attributes.

In this solution, the CISM is assumed to be embedded into the VNFM (similar as option 2 as described in clause 7.2.4.3 in ETSI GR NFV IFA 029 [i.22]). A VNFM with CISM embedded is capable of processing MCIOP and can be identified by vnfmInfo attribute (e.g. CismEmbeddedVnfM) as described in the VNFD.

In this solution, the Grant request also needs to be enhanced to be capable of carrying required resource description information rather than the referenced descriptor identifier defined in the VNFD.

This solution is illustrated in figure 5.2.2.6.1.2-1 as shown below.

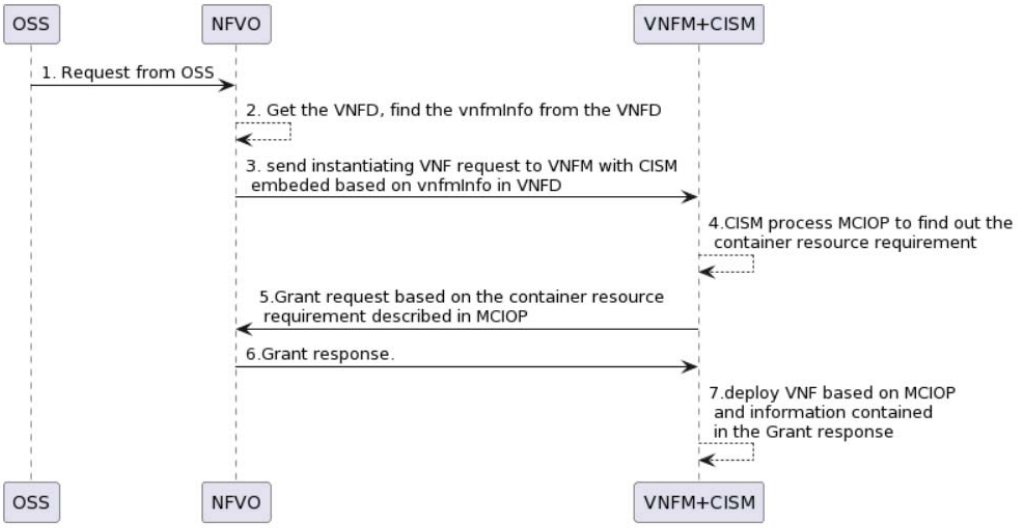
OSS
NFVO
VNFM+CISM
1. Request from OSS
2. Get the VNFD, find the vnfmInfo from the VNFD
3. send instantiating VNF request to VNFM with CISM embeded based on vnfmInfo in VNFD
4.CISM process MCIOP to find out the container resource requirement
5.Grant request based on the container resource requirement described in MCIOP
6.Grant response.
7.deploy VNF based on MCIOP and information contained in the Grant response
OSS
NFVO
VNFM+CISM

Figure 5.2.2.6.1.2-1: Deployment procedure of solution SOL 1-1a

5.2.2.6.1.3 Potential Solution SOL 1-1b

In this solution, it is proposed that the VNFM is capable of requesting the processing result of MCIOP from the CISM. After receiving the processing result of MCIOP from the CISM, the VNFM sends Grant request to the NFVO asking for VNF LCM operation authorization. In this solution, the Grant request also needs to be enhanced to be capable of carrying required resource description information rather than the referenced describer identifier defined in VNFD.

This solution is illustrated in figure 5.2.2.6.1.3-1 as shown below.

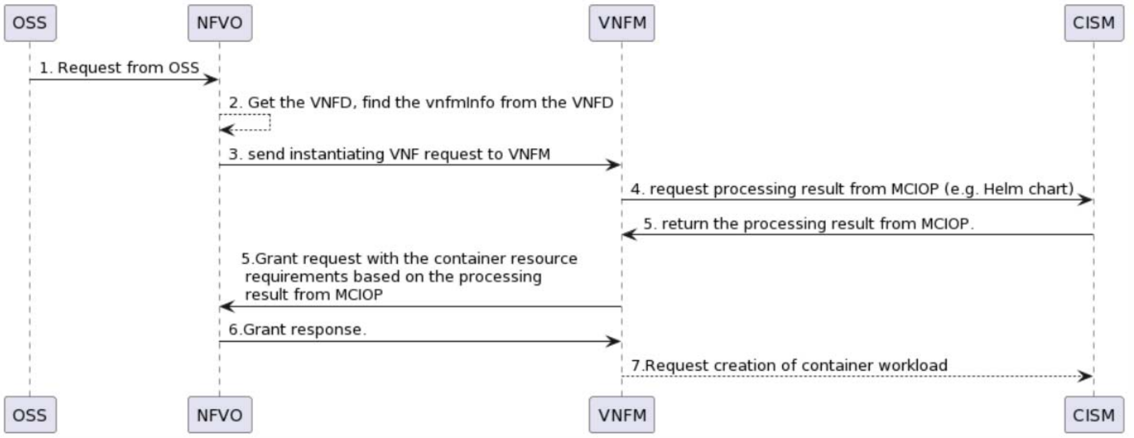
OSS
NFVO
VNFM
CISM
1. Request from OSS
2. Get the VNFD, find the vnfmInfo from the VNFD
3. send instantiating VNF request to VNFM
4. request processing result from MCIOP (e.g. Helm chart)
5. return the processing result from MCIOP.
5.Grant request with the container resource requirements based on the processing result from MCIOP
6.Grant response.
7.Request creation of container workload
OSS
NFVO
VNFM
CISM

Figure 5.2.2.6.1.3-1: Deployment procedure of solution SOL 1-1b

5.2.2.6.1.4 Potential Solution SOL 1-2

The virtualized resource information included in VNFD is mainly used for VNFM to send Grant request to NFVO.

In this solution, the virtualized resource information described in VNFD can focus on cluster resource requirements. In the VNFD, a new cluster requirement information element can be added, the new cluster requirement information element contains the summary of resource requirements for the deployment of vnfc in cluster nodes (e.g. virtual machine or bare metal server), the following information could be included:

- CPU, memory, storage, accelerator hardware and cluster enhancement capabilities requirement for each cluster node.
- Affinity and anti-affinity policy between cluster nodes.

Other virtualized resource related information element such as, virtualComputeDesc, osContainerDesc, virtualStorageDesc and swImageDesc can be absent in the VNFD.

Based on the cluster requirement described in VNFD, VNFM can exchange the grant request with NFVO. Since the cluster resource requirements in VNFD includes all the resource requirements for deployment of the VNF, once NFVO accepts the grant request and allocate an appropriate CISM to VNFM, the resource managed by CISM should fit the requirement to deploy the VNF.

In this proposal, there is no resource duplication between VNFD and MCIOP, since the resource requirements described in VNFD is at cluster layer and the resource described in MCIOP is at containers layer.

5.2.2.6.1.5 Potential Solution SOL 1-3

In this solution, it is suggested not to include MCIOP in the VNF package at all. The communication between VNFM and CISM is based on the information described in VNFD and transferred to Kubernetes® API.

This solution is related to gap analysis with Kubernetes® as described in clause 5.4.

5.2.2.6.1.6 Solution evaluation

Table 5.2.2.6.1.6-1 provides Impact evaluation of the solutions described for this gap.

Table 5.2.2.6.1.6-1: Solutions evaluation for Gap#1

	SOL 1-1a	SOL 1-1b	SOL 1-2	SOL 1-3
Impacts on VNFD	• New value (e.g. CismEmbeddedVnfM) of vnfmInfo attribute needs to be defined. • Guidance needs to be defined to state that for containerized VNF, virtualComputeDesc, osContainerDesc, virtualStorageDesc and swImageDesc are not mandatory to be present.	• Guidance needs to be defined to state that for containerized VNF, virtualComputeDesc, osContainerDesc, virtualStorageDesc and swImageDesc are not mandatory to be present.	• new cluster requirement information element needs to be defined. • Guidance needs to be defined to state that for containerized VNF, virtualComputeDesc, osContainerDesc, virtualStorageDesc and swImageDesc are not mandatory to be present.	• VNF package needs to be updated to allow the absence of MCIOP in the VNF package.
Impacts on VNFM	• CISM needs to be embedded into the VNFM, and this creates a tight coupling of CISM functionality into the VNFM. In case helm chart is used as MCIOP, at least Helm^{TM} needs to be embedded into VNFM. • The VNFM is expected to fill in the Grant request information with the result from the MCIOP processing, instead of the one from the VNFD (or at least complement it).	• The VNFM is expected to fill in the Grant request with resource description information produced by "translating" the containerized workload manifests resulting from the MCIOP processing, instead of providing a resource definition identification from the VNFD.	• No major impact, grant request is based on VM cluster requirements. • Note: Impacts for bare-metal cluster is not covered by the current version of the present document.	• VNFM is expected to request containerized workloads management operations from the CISM interfaces based on VNFD without MCIOP. • See note 3.
Impacts on NFVO	• NFVO needs to be capable of processing Grant request based on the resource requirement information described in the Grant request.	• NFVO needs to be capable of processing Grant request based on the resource requirement information described in the Grant request.	• The granularity of the correlation between the consumed resources of the VNF/VNFC and granted resources by NFVO will be changed.	• No impact.
Impacts on interface between NFVO and VNFM	• In the Grant request, the resource requirement information needs to be carried rather than the referenced describer identifier defined in VNFD.	• In the Grant request, the resource requirement information needs to be carried rather than the referenced describer identifier defined in VNFD.	• Not enough information in the granting and VNF LCM interfaces to perform a correlation between the different resource granularities. • See note 2.	• No impact.
Impacts on CISM	• No impact.	• No impact.	• No impact.	• No impact.

	SOL 1-1a	SOL 1-1b	SOL 1-2	SOL 1-3
Utilise Open source ecosystem	• Yes, e.g. Helm.	• Yes, e.g. Helm.	• Yes, e.g. Helm.	• VNFD is purely based on ETSI NFV standard. • See note 1.

NOTE 1: The evaluation of SOL 1-3 related open source is not covered by the current version of the present document.
NOTE 2: Impact for bare-metal cluster is not covered by the current version of the present document.
NOTE 3: This is already supported in ETSI GS NFV-IFA 010 [i.23].

In summary, solution SOL 1-1a and SOL 1-1b has impacts on VNFD, VNFM, NFVO and the Grant interface operation between NFVO and VNFM. SOL 1-1a binds CISM and VNFM to be deployed together, and in both SOL 1-1a and SOL 1-1b, the VNFM is expected to fill in the Grant request information with the result from the MCIOP processing. On the other hand, solution SOL 1-2 implies adding new cluster requirement information element in VNFD, and VNFM initiates Grant based on the cluster requirements. SOL 1-1a, SOL 1-1b and SOL 1-2 can take advantage of the current open source ecosystem, e.g. Helm, this simplifies the work of VNFD design and better integrates into the cloud native ecosystem.

5.2.2.6.2 Gap# 2: gap related to input parameter mapping in VNFD

5.2.2.6.2.1 Introduction

This gap is identified in Table 5.2.2.1.3-1 comparison of vnfExtCpd and asdExtCpd, in which it shows that in vnfExtCpd in VNFD there is no corresponding attribute mapped to inputParamMappings in asdExtCpd in ASD. According to the description of inputParamMappings in [i.6], this attribute specifies the mapping between the parameter name defined in ASD and the parameter name defined in the deployment artifact (e.g. Helm chart), so the orchestrator (e.g. SO in ONAP) can configure the input parameter of the deployment artifact based on the mappings.

5.2.2.6.2.2 Potential Solution SOL 2-1

The mapping between parameters from VNF lifecycle management interface and from the VNFD into a set of parameters to be provided to the MCIOP can be realised as an artifact in the VNF package.

5.2.2.6.3 Gap#3: gap related to parameter related information in VNF LCM API

5.2.2.6.3.1 Introduction

The gaps for ETSI NFV standard with regard to ONAP ASD are shown in Table 5.2.2.6.3.1-1.

Table 5.2.2.6.3.1-1: the gaps for ETSI NFV with regard to ONAP ASD

Table No.	Related operation in ETSI NFV	Missing parameter in ETSI NFV	Comments
Table 5.2.2.2.2-1	Create VNF Identifier	additionalParams	ONAP ASD introduces it to support any additional parameters for ONAP orchestrator, such as CNFM when create AS Instance.
Table 5.2.2.2.3-1 Table 5.2.2.2.3-2	Instantiate VNF	deploymentItems	ONAP ASD introduces it to support providing lifecycle parameters for deploymentItems.
		externalIPs	ONAP ASD introduces it to support setting of external IPs.

5.2.2.6.3.2 Potential Solution SOL 3-1

This is to resolve the gap for externalIPs as described in Table 5.2.2.6.3.1-1.

This solution proposes adding externalIP attribute in VirtualCpAddressData in InstantiateVnfRequest as defined in ETSI GS NFV-SOL 003 [i.8]. With this new attribute, the values of externalIP can be assigned to VNFM when instantiating a VNF.

5.3 Gap analysis with OpenStack® Tacker

5.3.1 Introduction of OpenStack® Tacker

5.3.1.1 Overview

OpenStack® Tacker is an official OpenStack® project building a Generic VNFM compliant with ETSI NFV ISG specification to deploy and operate VNFs on an NFVI with other NFV-MANO functional entities. OpenStack® Tacker is able to manage virtualized resources of hybrid VM and container for VNFs by VNF Package.

5.3.1.2 Relationship between OpenStack Tacker and ETSI NFV ISG

OpenStack Tacker supports VNF LCM API specified in V3.3.1 of ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13] with using VNF Package specified in V2.6.1 of ETSI GS NFV-SOL 004 [i.21] including TOSCA based VNFD specified in ETSI GS NFV-SOL 001 [i.7].

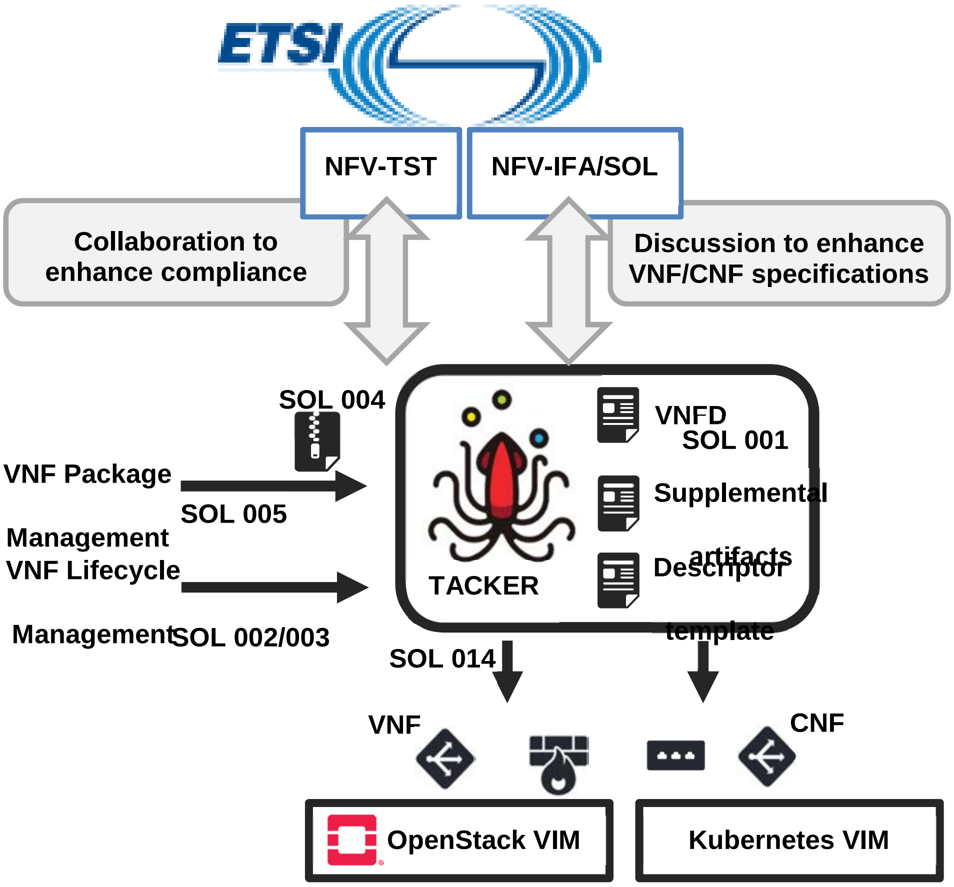

Figure 5.3.1.2-1: Overview of OpenStack Tacker project GS NFV-SOL 005

OpenStack® Tacker community continues to work on collaboration activities with ETSI NFV ISG. OpenStack® Tacker supports ETSI NFV-SOL V2.6.1 specification (Wallaby release) and V3.3.1 specification (Xena release) and tested API conformance test using ETSI GS NFV-TST 010 [i.15] Robot framework.

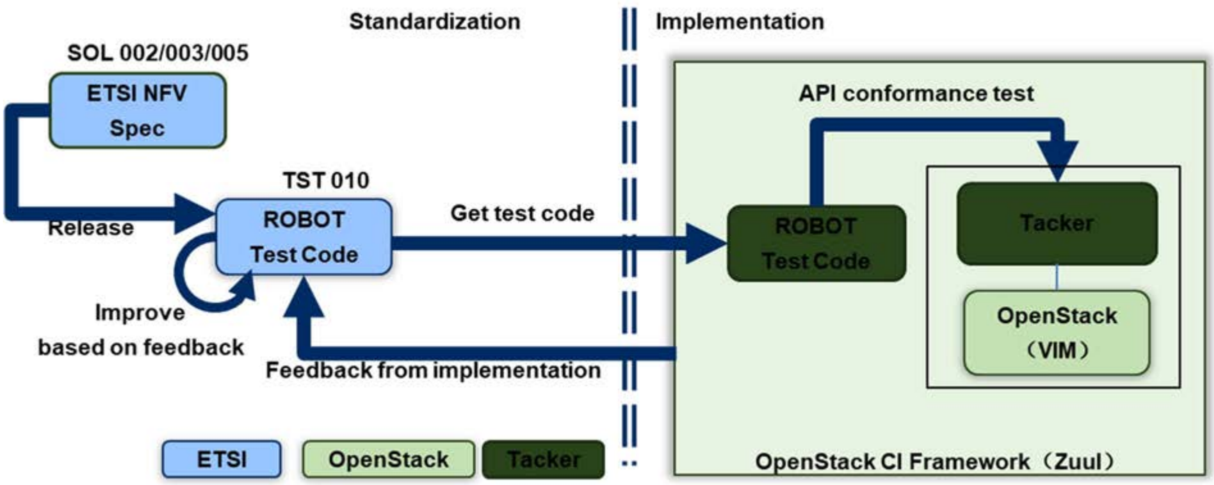

Figure 5.3.1.2-2: Collaboration between ETSI NFV ISG and OpenStack Tacker project

5.3.1.3 Proprietary solution for OS container related

In order to manage OS container by VNF LCM API specified in of ETSI GS NFV-SOL 002 (V3.3.1) [i.32] and ETSI GS NFV-SOL 003 [i.13], OpenStack Tacker uses additionalParams that is Key Value Pair described in ETSI NFV-SOL Tacker Use Cases [i.20] as following.

Table 5.3.1.3-1: Proprietary attributes in additionalParams

Operation	Key	Description of value	Example
Instantiate VNF operation	helm_chart_path	File path of helm chart.	"helm_chart_path": "Files/kubernetes/test-chart-0.1.0.tgz"
	namespace	Namespace to deploy Kubernetes resources.	"namespace": "test_A"

	helm_parameters	Parameters to install helm chart.	"helm_parameters": {
			"service.port": 8081
			}

	helm_values_names	This parameter specifies the parameter name to be set as Helm install parameter. In this operation, "replica" is this parameter name.	"helm_value_names": {
			"VDU1": {
			"replica": "replicaCountVdu1"
			},
			}

Scale VNF operation	lcm-kubernetes-def-files	path of Kubernetes resource definition file	"lcm-kubernetes-def-files": ["Files/kubernetes/deployment_scale.yaml"]
Heal VNF operation	lcm-kubernetes-def-files	path of Kubernetes resource definition file	"lcm-kubernetes-def-files": ["Files/kubernetes/deployment_heal_simple.yaml"]
Change current VNF package operation	upgrade_type	Type of file update operation method. Specify Blue-Green or Rolling update.	"upgrade_type": "RollingUpdate"
	vdu_params	VDU information of target VDU to update. In this operation, "vdu_id" is this VDU information.	"vdu_params": [{ "vdu_id": "VDU1" }]
	lcm-kubernetes-def-files	path of Kubernetes resource definition file	"lcm-kubernetes-def-files": ["Files/new_kubernetes/new_deployment.yaml"]

5.3.2 Gap analysis

5.3.2.1 Gap analysis related to VNFD

5.3.2.1.1 Introduction of the related work

OpenStack Tacker VNFD model of as described in [i.9] describes a VNF by using one or more TOSCA service template as defined in TOSCA-Simple-Profile-yaml-v1.2 [i.25]. TOSCA.meta is designed following ETSI GS NFV-SOL 004 [i.21], VNFD.yaml is composed as a collection of multiple yaml files that are downloaded from ETSI website, i.e. etsi_nfv_sol001_common_types.yaml and etsi_nfv_sol001_vnfd_types.yaml, and specific VNFD files for particular VNF following ETSI GS NFV-SOL 001 [i.7]. As OpenStack proprietary solution, k8s.yaml is composed as a collection of multiple yaml files designed for particular VNF.

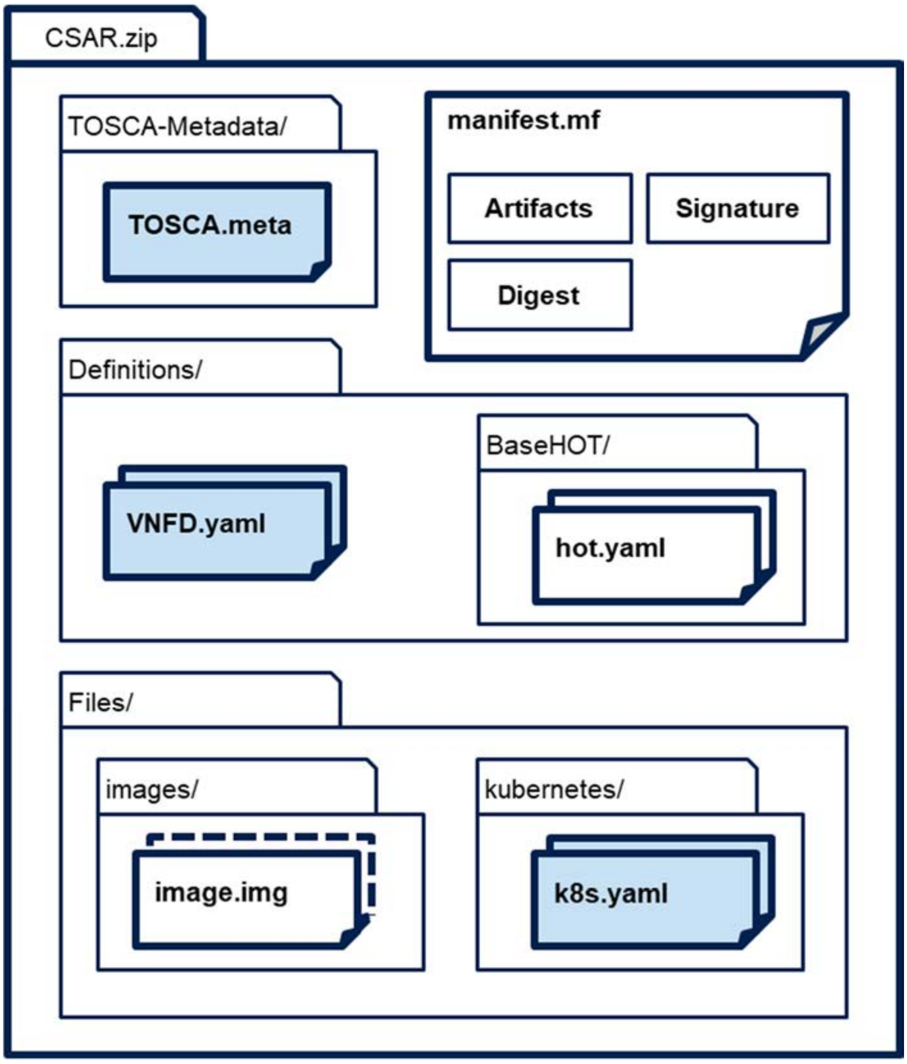

Figure 5.3.2.1.1-1: Supported VNF Package by OpenStack Tacker

5.3.2.1.2 Comparison and analysis of VNFD

Table 5.3.2.1.2-1 illustrates an analysis of the attributes in VNFD as specified in clause 7.1.2 in ETSI GS NFV-IFA 011 [i.3] and OpenStack® Tacker VNFD model as specified in [i.9]. The analysis illustrates the implementation of the VNFD specified by ETSI NFV by OpenStack® Tacker.

NOTE: Current OpenStack Tacker VNFD is based on the ETSI GS NFV-SOL 001 (V2.6.1) [i.33], and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNFD are likely to be identified due to the version mismatch.

Table 5.3.2.1.2-1: Comparison of VNFD and OpenStack Tacker VNFD

Attributes of Vnfd information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of OpenStack Tacker VNFD information element		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
vnfdId	1	descriptor_id	1	"descriptor_id" is property name of ETSI GS NFV-SOL 001 [i.7].
vnfdExtInvariantId	0..1			No correspondence. See note 1.
vnfProvider	1	provider	1	"provider" is property name of ETSI GS NFV-SOL 001 [i.7].
vnfProductName	1	product_name	1	"product_name" is property name of ETSI GS NFV-SOL 001 [i.7].
vnfSoftwareVersion	1	software_version	1	"software_version" is property name of ETSI GS NFV-SOL 001 [i.7].
vnfdVersion	1	descriptor_version	1	"descriptor_version" is property name of ETSI GS NFV-SOL 001 [i.7].
vnfProductInfoName	0..1			No correspondence
vnfProductInfoDescription	0..1			No correspondence
vnfmInfo	1..N	vnfm_info	1..N	"vnfm_info" is property name of ETSI GS NFV-SOL 001 [i.7].
localizationLanguage	0..N			No correspondence
defaultLocalizationLanguage	0..1			No correspondence
vdu	1..N	tosca.nodes.nfv.VduCp, tosca.nodes.nfv.Vdu.Compute, tosca.nodes.nfv.Vdu.VirtualBlockStorage	1..N	OpenStack Tacker VNFD support partial, and OpenStack Tacker VNFD does not support tosca.nodes.nfv.Vdu.OsContainerDeployableUnit, tosca.nodes.nfv.Vdu.OsContainer, tosca.nodes.nfv.Vdu.VirtualObjectStorage and tosca.nodes.nfv.Vdu.VirtualFileStorage. See note 2.
virtualComputeDesc	0..N	virtual_compute in tosca.nodes.nfv.Vdu.Compute	1	OpenStack Tacker VNFD support tosca.nodes.nfv.Vdu.Compute when properties in virtual_compute lists property of VirtualCompute. OpenStack Tacker VNFD does not support tosca.capabilities.nfv.VirtualCompute.
virtualStorageDesc	0..N	tosca.nodes.nfv.Vdu.VirtualBlockStorage	0..N	OpenStack Tacker VNFD support partial, and OpenStack Tacker VNFD does not support tosca.nodes.nfv.Vdu.VirtualObjectStorage and tosca.nodes.nfv.Vdu.VirtualFileStorage. See note 2.
osContainerDesc	0..N			No correspondence. See note 1.
swImageDesc	0..N	tosca.artifacts.nfv.SwImage	0..N	
intVirtualLinkDesc	0..N	tosca.nodes.nfv.VnfVirtualLink	0..N	
securityGroupRule	0..N			No correspondence.
vnfExtCpd	1..N	tosca.nodes.nfv.VduCp	0..N	OpenStack Tacker VNFD support partial, and OpenStack Tacker VNFD does not support tosca.nodes.nfv.VnfExtCp and tosca.nodes.nfv.VipCp. See note 2.
vipCpd	0..N			No correspondence. See note 1.
virtualCpd	0..N			No correspondence. See note 1.
deploymentFlavour	1..N	flavour_id and low level service template	1..N	OpenStack Tacker VNFD follows requirements a, b, c and in clause 6.11.2 of ETSI GS NFV-SOL 001 [i.7].

Attributes of Vnfd information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of OpenStack Tacker VNFD information element		Comments
Attribute	Cardinality	Attribute	Cardinality	
configurableProperties	0..1			No correspondence
modifiableAttributes	0..1			No correspondence
lifeCycleManagementScript	0..N	vnflcm	0..N	OpenStack Tacker VNFD follows interface type vnflcm of ETSI GS NFV-SOL 001 [i.7]. OpenStack Tacker VNFD does not support "tosca.interfaces.nfv.ChangeCurrentVnfPackage". See note 1.
vnfIndicator	0..N			No correspondence. See note 1.
autoScale	0..N			No correspondence. See note 1.
vnfPackageChangeInfo	0..N			No correspondence. See note 1.
lcmOperationCoordination	0..N			No correspondence. See note 1.
mciopId	0..N			No correspondence. See note 1. "lcm-kubernetes-def-files" of "additionalParams" attribute is workable solution to specify path of Kubernetes resource definition file for each operation described in Table 5.3.1.3-1 as OpenStack Tacker proprietary solution.
NOTE 1: "etsi_nfv_sol001_vnfd_types.yaml" (V2.6.1) does not include this attribute. NOTE 2: OpenStack Tacker VNFD follows mapping of VDU defined in clause 6.1 of ETSI GS NFV-SOL 001 [i.7] that are different from ETSI GS NFV-IFA 011 [i.3].				

5.3.2.2 Gap analysis related to VNF LCM

5.3.2.2.1 Introduction of the related work

The lifecycle management interface of OpenStack® Tacker VNF LCM API as defined in [i.11] manages the VNF lifecycle management operations of VNF instances. This interface allows the NFVO to invoke VNF lifecycle management operations of VNF instances towards the VNFM. The operations as provided through this interface are:

- Create a new VNF instance resource
- Instantiate a VNF instance
- Terminate a VNF instance
- Heal a VNF instance
- Delete a VNF instance
- Show VNF instance
- List VNF instance
- Scale a VNF instance
- Update a VNF instance
- Change External VNF Connectivity
- Change Current VNF Package

OpenStack® Tacker has Database (DB) that manages managed object such as VNF Information and is able to request to virtualized resource operation to VIM and CISM to deploy and operate actual VNF instances.

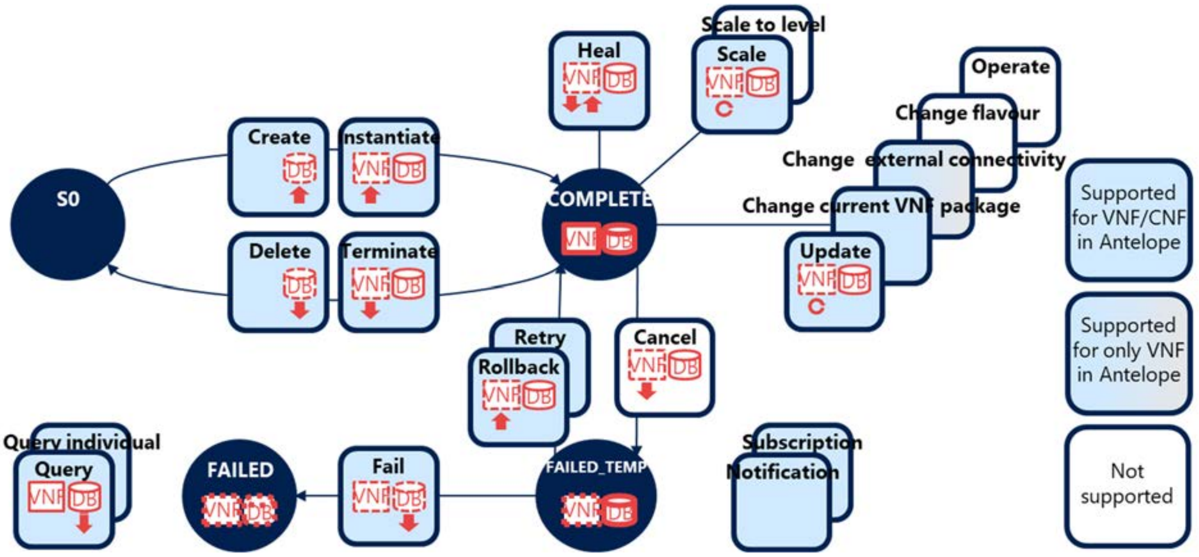

Figure 5.3.2.2.1-1: supported VNF LCM operation by OpenStack Tacker

5.3.2.2.2 Comparison and analysis of Create VNF Identifier and Create a new VNF instance resource

Table 5.3.2.2.2-1 illustrates a comparison of the attributes in Create VNF Identifier operation as specified in clause 7.2.2 in ETSI GS NFV-IFA 007 [i.4] and Create a new VNF instance resource as specified in [i.11].

NOTE: Current OpenStack Tacker VNF LCM API is based on the ETSI GS NFV-SOL 003 [i.13] V3.3.1, and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF LCM API are likely to be identified due to the version mismatch.

Table 5.3.2.2.2-1: Comparison of input parameters in Create VNF Identifier and Create a new VNF instance resource operation

Input Parameters in Create VNF Identifier operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Create a new VNF instance resource		Comments
Parameter	Cardinality	Attribute	Cardinality	
vnfdId	1	vnfdId	1	
vnfInstanceName	0..1	vnfInstanceName	0..1	
vnfInstanceDescription	0..1	vnfInstanceDescription	0..1	
metadata	0..N	metadata	0..1	See note.
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.2.2-2: Comparison of output parameters returned by Create VNF Identifier and Create a new VNF instance resource operation

Output Parameters returned by Create VNF Identifier operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Create a new VNF instance resource		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
vnfInstanceId	1	Id	1	See note.
		vnfInstanceName	0..1	See note.
		vnfInstanceDescription	0..1	See note.
		vnfdId	1	See note.
		vnfProvider	1	See note.
		vnfProductName	1	See note.
		vnfSoftwareVersion	1	See note.
		vnfdVersion	1	See note.
		vnfConfigurableProperties	0..1	See note.
		instantiationState	1	See note.
		metadata	0..1	See note.
		extensions	0..1	See note.
		_links	1	See note.
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.2.3 Comparison and analysis of Instantiate VNF and Instantiate a VNF instance operation

Table 5.3.2.2.3-1 illustrates a comparison of the attributes in Instantiate VNF operation as specified in clause 7.2.3 in ETSI GS NFV-IFA 007 [i.4] and Instantiate a VNF instance as specified in [i.11].

NOTE: Current OpenStack Tacker VNF LCM API is based on the ETSI GS NFV-SOL 003 [i.13] V3.3.1, and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF LCM API are likely to be identified due to the version mismatch.

Table 5.3.2.2.3-1: Comparison of input parameters in Instantiate VNF and Instantiate a VNF instance operation

Input Parameters in Instantiate VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Instantiate a VNF instance operation		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
vnfInstanceId	1			Maps to vnfInstanceId in the REST Interface, i.e. POST .../vnf_instances/{vnfInstanceId}/instantiate (Instantiate a vnf instance) See note 1.
flavourId	1	flavourId	1	
instantiationLevelId	0..1	instantiationLevelId	0..1	
targetScaleLevelInfo	0..N			No correspondence. See note 2.
extVirtualLink	0..N	extVirtualLinks	0..N	See note 1.
extManagedVirtualLink	0..N	extManagedVirtualLinks	0..N	See note 1.
vimConnectionInfo	0..N	vimConnectionInfo	0..N	
localizationLanguage	0..1	localizationLanguage	0..1	
additionalParam	0..N	additionalParams	0..1	In the case of containerised VNF, namespace and parameter to install helm described in Table 5.3.1.3-1 as OpenStack Tacker proprietary solution. See note 1.
extension	0..N	extensions	0..1	See note 1.

Input Parameters in Instantiate VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Instantiate a VNF instance operation		Comments
Attribute	Cardinality	Attribute	Cardinality	
vnfConfigurableProperty	0..N	vnfConfigurableProperty	0..1	
NOTE 1: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13] which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4]. NOTE 2: V3.3.1 of ETSI GS NFV-SOL 003 [i.13] does not include this attribute.				

Table 5.3.2.2.3-2: Comparison of output parameters returned by Instantiate VNF and Instantiate a VNF instance operation

Output Parameters returned by Instantiate VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Instantiate a VNF instance operation		Comments
Attribute	Cardinality	Attribute	Cardinality	
lifecycleOperationOccurrenceId	1			Maps to lifcecycleOperationOccurenceId in the URI in "Location" HTTP header response of the request.

5.3.2.2.4 Comparison and analysis of Terminate VNF and Terminate a VNF instance operation

Table 5.3.2.2.4-1 illustrates a comparison of the attributes in Terminate VNF operation as specified in clause 7.2.7 in ETSI GS NFV-IFA 007 [i.4] and Terminate a VNF instance as specified in [i.11].

NOTE: Current OpenStack Tacker VNF LCM API is based on the ETSI GS NFV-SOL 003 [i.13] V3.3.1, and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF LCM API are likely to be identified due to the version mismatch.

Table 5.3.2.2.4-1: Comparison of input parameters in Terminate VNF and Terminate a VNF instance operation

Input Parameters in Terminate VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Terminate a VNF instance		Comments
Parameter	Cardinality	Attribute	Cardinality	
vnfInstanceId	1			Maps to vnfInstanceId in the REST Interface, i.e. POST .../vnf_finstances/{vnfInstanceId}/instantiate (Terminate a vnf instance) See note.
terminationType	1	terminationType	1	
gracefulTerminationTimeout	0..1	gracefulTerminationTimeout	0..1	
additionalParam	0..N	additionalParams	0..1	See note.
NOTE : OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4]				

Table 5.3.2.2.4-2: Comparison of output parameters returned by Terminate VNF and Terminate a VNF instance operation

Output Parameters returned by Terminate VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Terminate a VNF instance operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
lifecycleOperationOccurrenceId	1			Maps to lifecycleOperationOccurenceId in the URI in "Location" HTTP header response of the request.

5.3.2.2.5 Comparison and analysis of Delete VNF Identifier and Delete a VNF instance

Table 5.3.2.2.5-1 illustrates a comparison of the attributes in Delete VNF Identifier operation as specified in clause 7.2.8 in ETSI GS NFV-IFA 007 [i.4] and Delete a VNF instance as specified in [i.11].

NOTE: Current OpenStack Tacker VNF LCM API is based on the ETSI GS NFV-SOL 003 [i.13] V3.3.1, and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF LCM API are likely to be identified due to the version mismatch.

Table 5.3.2.2.5-1: Comparison of input parameters in Delete VNF Identifier and Delete a VNF instance operation

Input Parameters in Delete VNF Identifier operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Delete a VNF instance		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
vnfInstanceId	1			Maps to vnfInstanceId in the REST Interface, i.e. DELETE .../vnf_instances/{vnfInstanceId}

Table 5.3.2.2.5-2: Comparison of output parameters returned by Delete VNF Identifier and Delete a VNF instance operation

Output Parameters returned by Instantiate VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Delete a VNF instance response		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
N/A		N/A		

5.3.2.2.6 Comparison and analysis of Query VNF and Show VNF instance operation

Table 5.3.2.2.6-1 illustrates a comparison of the attributes in Query VNF operation as specified in clause 7.2.9 in ETSI GS NFV-IFA 007 [i.4] and Show VNF instance as specified in [i.11].

NOTE 1: Current OpenStack Tacker VNF LCM API is based on the ETSI GS NFV-SOL 003 [i.13] V3.3.1, and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF LCM API are likely to be identified due to the version mismatch.

NOTE 2: ETSI GS NFV-SOL 003 [i.13] specifies 2 operations (Read an "Individual VNF instance" resource and Query multiple VNF instances) to realize Query VNF operation as clause A.6 in ETSI GS NFV-SOL 003 [i.13], and Show VNF instance operation is corresponding to Read an "Individual VNF instance" resource.

Table 5.3.2.2.6-1: Comparison of input parameters in Query VNF and Show VNF instance operation

Input Parameters in Query VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Show VNF instance		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
filter	1			Maps to vnfInstanceId in the REST Interface, i.e. GET .../vnf_finstances/{vnfInstanceId} See note.
attributeSelector	0..N			No correspondence.
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.2.6-2: Comparison of output parameters returned by Query VNF and Show VNF instance operation

Output Parameters returned by Query VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Show VNF instance operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
vnfInfo	0..N	VNFInstance	1	See note.
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.2.6-3: Comparison of information model of vnfInfo and VNFInstance

Attributes in vnfInfo in ETSI GS NFV-IFA 007 [i.4]		Attributes in VNFInstance		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
vnfInstanceId	1	Id	1	See note 1.
vnfInstanceName	0..1	vnfInstanceName	0..1	
vnfInstanceDescription	0..1	vnfInstanceDescription	0..1	
vnfdId	1	vnfdId	1	
vnfProvider	1	vnfProvider	1	
vnfProductName	1	vnfProductName	1	
vnfSoftwareVersion	1	vnfSoftwareVersion	1	
vnfdVersion	1	vnfdVersion	1	
vnfConfigurableProperty	0..N	vnfConfigurableProperties	0..1	See note 1.
vimConnectionInfo	0..N	vimConnectionInfo	0..N	
cirConnectionInfo	0..N			No correspondence. See note 2.
mciopRepositoryInfo	0..N			No correspondence. See note 2.
instantiationState	1	instantiationState	1	
instantiatedVnfInfo	0..1	instantiatedVnfInfo	0..1	
metadata	0..N	metadata	0..1	See note 1.
extension	0..N	extensions	0..1	See note 1.
		_links	1	See note 1.
NOTE 1: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4]. NOTE 2: V3.3.1 of ETSI GS NFV-SOL 003 [i.13] does not include this attribute.				

Table 5.3.2.2.6-4: Comparison of information model of instantiatedVnfInfo

Attributes in instantiatedVnfInfo in ETSI GS NFV-IFA 007 [i.4]		Attributes in instantiatedVnfInfo		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
flavourId	1	flavourId	1	
vnfState	1	vnfState	1	
scaleStatus	0..N	scaleStatus	0..N	
maxScaleLevel	0..N	maxScaleLevels	0..N	See note 1.
extCpInfo	1..N	extCpInfo	1..N	
vipCpInfo	0..N			No correspondence. See note 2.
virtualCpInfo	0..N			No correspondence. See note 2.
extVirtualLinkInfo	0..N	extVirtualLinkInfo	0..N	
extManagedVirtualLinkInfo	0..N	extManagedVirtualLinkInfo	0..N	See note 1.
monitoringParameter	0..N			No correspondence.
localizationLanguage	0..1			No correspondence.
vnfcResourceInfo	0..N	vnfcResourceInfo	0..N	
vnfVirtualLinkResourceInfo	0..N	vnfVirtualLinkResourceInfo	0..N	
virtualStorageResourceInfo	0..N	virtualStorageResourceInfo	0..N	
mcioInfo	0..N			No correspondence. See note 2.
		vnfcInfo	1..N	VNFInstance data type in ETSI GS NFV-SOL 002 [i.31] includes vnfcInfo as difference between ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13] described in clause D.2.2 in ETSI GS NFV-SOL 002 [i.31], and OpenStack Tacker merged VNF LCM API of both ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13] as OpenStack Tacker proprietary solution.
		metadata	0..1	OpenStack Tacker proprietary solution to use other lifecycle operation after successful instantiation is stored.
NOTE 1: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4]. NOTE 2: V3.3.1 of ETSI GS NFV-SOL 003 [i.13] does not include this attribute.				

5.3.2.2.7 Comparison and analysis of Query VNF and List VNF instance operation

Table 5.3.2.2.7-1 illustrates a comparison of the attributes in Query VNF operation as specified in clause 7.2.9 in ETSI GS NFV-IFA 007 [i.4] and List VNF instance as specified in [i.11].

NOTE 1: Current OpenStack Tacker VNF LCM API is based on the ETSI GS NFV-SOL 003 [i.13] V3.3.1, and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF LCM API are likely to be identified due to the version mismatch.

NOTE 2: ETSI GS NFV-SOL 003 [i.13] specifies 2 operation (Read an "Individual VNF instance" resource and Query multiple VNF instances) to realize Query VNF operation as clause A.6 in ETSI GS NFV-SOL 003 [i.13], and List VNF instance operation is corresponding to Read an "Individual VNF instance" resource.

Table 5.3.2.2.7-1: Comparison of input parameters in Query VNF and List VNF instance operation

Input Parameters in Query VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in List VNF instance		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
filter	1			Attribute-based filtering in REST interface, i.e. GET ../vnf_instances?filter=<filterExpr> See note.
attributeSelector	0..N			Attribute selector in REST interface, i.e. all_fields, fields, exclude_fields and exclude_default. See note.
NOTE : OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4]				

Table 5.3.2.2.7-2: Comparison of output parameters returned by Query VNF and List VNF instance operation

Output Parameters returned by Query VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in List VNF instance operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
vnfInfo	0..N	VNFinstance	0..N	See note. Analysis between vnfInfo and VNFinstance. See Table 5.3.2.2.7-3 and Table 5.3.2.2.7-4.
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.2.8 Comparison and analysis of Scale VNF and Scale a VNF instance operation

Table 5.3.2.2.8-1 illustrates a comparison of the attributes in Scale VNF operation as specified in clause 7.2.4 in ETSI GS NFV-IFA 007 [i.4] and Scale a VNF instance as specified in [i.11].

NOTE: Current OpenStack Tacker VNF LCM API is based on the ETSI GS NFV-SOL 003 [i.13] V3.3.1, and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF LCM API are likely to be identified due to the version mismatch.

Table 5.3.2.2.8-1: Comparison of input parameters in Scale VNF and Scale a VNF instance operation

Input Parameters in Scale VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Scale a VNF instance		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
vnfInstanceId	1			Maps to vnfInstanceId in the REST Interface, i.e. GET .../vnf_instances/{vnfInstanceId}/scale See note.
type	1	type	1	
aspectId	1	aspectId	1	
numberOfSteps	0..1	numberOfSteps	0..1	

Input Parameters in Scale VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Scale a VNF instance		Comments
Parameter	Cardinality	Attribute	Cardinality	
additionalParam	0..N	additionalParams	0..1	In the case of containerised VNF, path of Kubernetes resource definition file described in Table 5.3.1.3-1 as OpenStack Tacker proprietary solution. See note.
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.2.8-2: Comparison of output parameters returned by Scale VNF and Scale a VNF instance operation

Output Parameters returned by Scale VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Scale a VNF instance operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
lifecycleOperationOccurrenceId	1			Maps to lifecycleOperationOccurenceId in the URI in "Location" HTTP header response of the request. See note.
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.2.9 Comparison and analysis of Heal VNF and Heal a VNF instance operation

Table 5.3.2.2.9-1 illustrates a comparison of the attributes in Heal VNF operation as specified in clause 7.2.10 in ETSI GS NFV-IFA 007 [i.4] and Heal a VNF instance as specified in [i.11].

NOTE: Current OpenStack Tacker VNF LCM API is based on the ETSI GS NFV-SOL 003 [i.13] V3.3.1, and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF LCM API are likely to be identified due to the version mismatch.

Table 5.3.2.2.9-1: Comparison of input parameters in Heal VNF and Heal a VNF instance operation

Input Parameters in Heal VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Heal a VNF instance		Comments
Parameter	Cardinality	Attribute	Cardinality	
vnfInstanceId	1			Maps to vnfInstanceId in the REST Interface, i.e. GET .../vnf_instances/{vnfInstanceId}/heal See note.
cause	0..1	cause	0..1	
additionalParam	0..N	additionalParams	0..1	In the case of containerised VNF, path of Kubernetes resource definition file described in Table 5.3.1.3-1 as OpenStack Tacker proprietary solution. See note.
		all	0..1	OpenStack Tacker proprietary solution to request whether network resource and storage resource are included in this heal operation.

Input Parameters in Heal VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Heal a VNF instance		Comments
Parameter	Cardinality	Attribute	Cardinality	
		vnfcInstanceId	0..N	HealVnfRequest data type in ETSI GS NFV-SOL 002 [i.31] includes vnfcInstanceId as difference between ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13] described in clause D.2.2 in ETSI GS NFV-SOL 002 [i.31], and OpenStack Tacker merged VNF LCM API of both ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13] as OpenStack Tacker proprietary solution.
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.2.9-2: Comparison of output parameters returned by Heal VNF and Heal a VNF instance operation

Output Parameters returned by Heal VNF operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Heal a VNF instance operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
lifecycleOperationOccurrenceId	1			Maps to lifecycleOperationOccurenceId in the URI in "Location" HTTP header response of the request. See note.
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.2.10 Comparison and analysis of Change External VNF Connectivity and Change External VNF Connectivity operation

Table 5.3.2.2.10-1 illustrates a comparison of the attributes in Change External VNF Connectivity operation as specified in clause 7.2.18 in ETSI GS NFV-IFA 007 [i.4] and Change External VNF Connectivity operation as specified in [i.11].

NOTE: Current OpenStack Tacker VNF LCM API is based on the ETSI GS NFV-SOL 003 [i.13] V3.3.1, and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF LCM API are likely to be identified due to the version mismatch.

Table 5.3.2.2.10-1: Comparison of input parameters in Change External VNF Connectivity and Change External VNF Connectivity operation

Input Parameters in Change External VNF Connectivity operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Change External VNF Connectivity operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
vnfInstanceId	1			Maps to vnfInstanceId in the REST Interface, i.e. GET .../vnf_finstances/{vnfInstanceId}/change_ext_conn See note.
extVirtualLink	0..N	extVirtualLinks	0..N	See note.
additionalParam	0..N	additionalParams	0..1	See note.

Input Parameters in Change External VNF Connectivity operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Change External VNF Connectivity operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
vimConnectionInfo	0..N	vimConnectionInfo	0..N	
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.2.10-2: Comparison of output parameters returned by Change External VNF Connectivity and Change External VNF Connectivity operation

Output Parameters returned by Change External VNF Connectivity operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Change External VNF Connectivity operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
lifecycleOperationOccurrenceId	1			Maps to lifecycleOperationOccurenceId in the URI in "Location" HTTP header response of the request. See note.
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.2.11 Comparison and analysis of Change current VNF Package and Change Current VNF Package operation

Table 5.3.2.2.11-1 illustrates a comparison of the attributes in Change current VNF Package operation as specified in clause 7.2.23 in ETSI GS NFV-IFA 007 [i.4] and Change Current VNF Package as specified in [i.11].

NOTE: Current OpenStack Tacker VNF LCM API is based on the ETSI GS NFV-SOL 003 [i.13] V3.3.1, and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF LCM API are likely to be identified due to the version mismatch.

Table 5.3.2.2.11-1: Comparison of input parameters in Change current VNF Package and Change Current VNF Package operation

Input Parameters in Change current VNF Package operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Change Current VNF Package operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
vnfInstanceId	1			Maps to vnfInstanceId in the REST Interface, i.e. GET .../vnf_instances/{vnfInstanceId}/change_vnfpkg See note.
vnfdId	1	vnfdId	1	
extVirtualLink	0..N	extVirtualLinks	0..N	See note.
extManagedVirtualLink	0..N	extManagedVirtualLinks		See note.
vimConnectionInfo	0..N	vimConnectionInfo	0..N	
additionalParam	0..N	additionalParams	0..1	In the case of containerised VNF, upgrade type, vdu information, and path of Kubernetes resource definition file described in Table 5.3.1.3-1 as OpenStack Tacker proprietary solution. See note.
extension	0..N			No correspondence.
vnfConfigurableProperties	0..N			No correspondence.
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.2.11-2: Comparison of output parameters returned by Change current VNF Package and Change Current VNF Package operation

Output Parameters returned by Change current VNF Package operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Change Current VNF Package operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
lifecycleOperationOccurrenceId	1			Maps to lifecycleOperationOccurenceId in the URI in "Location" HTTP header response of the request. See note.
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.2.12 Comparison and analysis of Modify VNF information and Update a VNF instance operation

Table 5.3.2.2.12-1 illustrates a comparison of the attributes in Modify VNF information operation as specified in clause 7.2.12 in ETSI GS NFV-IFA 007 [i.4] and Update a VNF instance as specified in [i.11].

NOTE: Current OpenStack Tacker VNF LCM API is based on the ETSI GS NFV-SOL 003 [i.13] V3.3.1, and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF LCM API are likely to be identified due to the version mismatch.

Table 5.3.2.2.12-1: Comparison of input parameters in Modify VNF information and Update a VNF insntance operation

Input Parameters in Modify VNF information operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Update a VNF instance operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
vnfInstanceId	1			Maps to vnfInstanceId in the REST Interface, i.e. PATCH .../vnf_instances/{vnfInstanceId} See note.
newValues	1..N			
		vnfInstanceName	0..1	See note.
		vnfInstanceDescription	0..1	See note.
		vnfdId	0..1	See note.
		vnfConfigurableProperties	0..1	See note.
		metadata	0..1	See note.
		extensions	0..1	See note.
		vimConnectionInfo	0..N	See note.
		vnfcInfoModifications	0..N	VnfInfoModificationRequest data type in ETSI GS NFV-SOL 002 [i.31] includes vnfcInfoModifications as difference between ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13] described in clause D.2.2 in ETSI GS NFV-SOL 002 [i.31], and OpenStack Tacker merged VNF LCM API of both ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13] as OpenStack Tacker proprietary solution.
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.2.12-2: Comparison of output parameters returned by Modify VNF information and Update a VNF instance operation

Output Parameters returned by Modify VNF information operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Update a VNF instance operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
lifecycleOperationOccurrenceId	1			Maps to lifecycleOperationOccurenceId in the URI in "Location" HTTP header response of the request. See note.
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.3 Gap analysis related to VNF performance management

5.3.2.3.1 Introduction of the related work

The performance management interface of OpenStack® Tacker VNF PM API as defined in [i.16] manages the VNF performance management operations of VNF instances. This interface allows the NFVO to invoke VNF performance management operations of VNF instances towards the VNFM. The operations as provided through this interface are:

- Create a PM job
- Delete a PM job
- Get for PM jobs
- Get a PM job
- Modify a PM job
- Get individual performance report

5.3.2.3.2 Comparison and analysis of Create PM job and Create a PM job

Table 5.3.2.3.2-1 illustrates a comparison of the attributes in Create PM job operation as specified in clause 7.4.2 in ETSI GS NFV-IFA 007 [i.4] and Create a PM job as specified in [i.16].

NOTE 1: Current OpenStack Tacker VNF PM API is based on the ETSI GS NFV-SOL 003 V3.3.1 [i.13] and ETSI GS NFV-SOL 013 v3.4.1 [i.17], and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF PM API are likely to be identified due to the version mismatch.

NOTE 2: ETSI GS NFV-SOL 003 [i.13] specifies Create PM Job operation that combines Subscribe operation and Create PM Job operation specified in ETSI GS NFV-IFA 007 [i.4] as clause A.7 in ETSI GS NFV-SOL 003 [i.13].

Table 5.3.2.3.2-1: Comparison of input parameters in Create PM job and Create a PM job operation

Input Parameters in Create PM job operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Create a PM job operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
objectSelector	1			No correspondence. See note 1.
> objectType	0..N	objectType	1	See note 1.
> objectFilter	0..1			No correspondence. See note 1.
> objectInstanceId	0..N	objectInstanceIds	1..N	See note 1.
		subObjectInstanceIds	0..N	See note 1.
		citeria	1	See note 1.
performanceMetric	0..N	> performanceMetric	0..N	
performanceMetricGroup	0..N	> performanceMetricGroup	0..N	
collectionPeriod	1	> collectionPeriod	1	

Input Parameters in Create PM job operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Create a PM job operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
reportingPeriod	1	> reportingPeriod	1	
reportingBoundary	0..1	> reportingBoundary	0..1	
		callbackUri	1	See note 1.
		authentication	1	See notes 1 and 2.
		metadata	1	OpenStack Tacker proprietary solution to input access information of external monitoring tool.
NOTE 1: OpenStack Tacker VNF PM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4]. NOTE 2: OpenStack Tacker VNF PM API follows ETSI GS NFV-SOL 013 V3.4.1 [i.17].				

Table 5.3.2.3.2-2: Comparison of output parameters returned by Create PM job and Create a PM job operation

Output Parameters returned by Create PM job operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Create a PM job operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
pmJobId	1	id	1	See note.
		objectType	1	See note.
		objectInstanceIds	1..N	See note.
		subObjectInstanceIds	0..N	See note.
		criteria	1	See note.
		callbackUri	1	See note.
		reports	0..N	See note.
		_links	1	See note.
NOTE: OpenStack Tacker VNF PM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.3.3 Comparison and analysis of Delete PM jobs and Delete a PM job

Table 5.3.2.3.3-1 illustrates a comparison of the attributes in Delete PM jobs operation as specified in clause 7.4.3 in ETSI GS NFV-IFA 007 [i.4] and Delete a PM job as specified in [i.16].

NOTE: Current OpenStack Tacker VNF PM API is based on the ETSI GS NFV-SOL 003 V3.3.1 [i.13], and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF PM API are likely to be identified due to the version mismatch.

Table 5.3.2.3.3-1: Comparison of input parameters in Delete PM jobs and Delete a PM job operation

Input Parameters in Delete PM jobs operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Delete a PM job operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
pmJobId	1..N			Maps to pmJobId in the REST Interface, i.e. DELETE .../pm_jobs/{pmJobId} Not support bulk PM jobs deletion. See note.
NOTE: OpenStack Tacker VNF PM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.3.3-2: Comparison of output parameters returned by Delete PM job and Delete a PM job operation

Output Parameters returned by Delete PM jobs operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Delete a PM job operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
deletedPmJobId	1			No correspondence. See note.
NOTE: OpenStack Tacker VNF PM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.3.4 Comparison and analysis of Query PM job and Get a PM job

Table 5.3.2.3.4-1 illustrates a comparison of the attributes in Query PM job operation as specified in clause 7.4.6 in ETSI GS NFV-IFA 007 [i.4] and Get a PM job as specified in [i.16].

NOTE 1: Current OpenStack Tacker VNF PM API is based on the ETSI GS NFV-SOL 003 V3.3.1 [i.13], and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF PM API are likely to be identified due to the version mismatch.

NOTE 2: ETSI GS NFV-SOL 003 [i.13] specifies 2 operations (Read a single PM job and Query PM jobs) to realize Query VNF operation as clause A.7 in ETSI GS NFV-SOL 003 [i.13], and Get a PM job operation is corresponding to Read a single PM job operation.

Table 5.3.2.3.4-1: Comparison of input parameters in Query PM job and Get a PM job operation

Input Parameters in Query PM job operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Get a PM job operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
filter	1			Maps to pmJobId in the REST Interface, i.e. GET .../pm_jobs/{pmJobId} See note.
NOTE: OpenStack Tacker VNF PM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.3.4-2: Comparison of output parameters returned by Query PM job and Get a PM job operation

Output Parameters returned by Query PM job operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Get a PM job operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
pmJob	0..N	PmJob	1	See note.
NOTE: OpenStack Tacker VNF PM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.3.4-3: Comparison of information model of pmJob and PmJob

Attributes in pmJob in ETSI GS NFV-IFA 007 [i.4]		Attributes in PmJob		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
pmJobId	1	id	1	See note.
objectSelector	1			No correspondence. See note.
> objectType	0..N	objectType	1	See note.
> objectFilter	0..1			No correspondence. See note.
> objectInstanceId	0..N	objectInstanceIds	1..N	See note.
		subObjectInstanceIds	0..N	See note.
		criteria	1	See note.
performanceMetric	0..N	> performanceMetric	0..N	
performanceMetricGroup	0..N	> performanceMetricGroup	0..N	
collectionPeriod	1	> collectionPeriod	1	
reportingPeriod	1	> reportingPeriod	1	
reportingBoundary	0..1	> reportingBoundary	0..1	
		callbackUri	1	See note.
		reports	0..N	See note.
		_links	1	See note.
NOTE: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.3.5 Comparison and analysis of Query PM job and Get for PM jobs

Table 5.3.2.3.5-1 illustrates a comparison of the attributes in Query PM job operation as specified in clause 7.4.6 in ETSI GS NFV-IFA 007 [i.4] and Get for PM jobs as specified in [i.16].

NOTE 1: Current OpenStack Tacker VNF PM API is based on the ETSI GS NFV-SOL 003 V3.3.1 [i.13], and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF PM API are likely to be identified due to the version mismatch.

NOTE 2: ETSI GS NFV-SOL 003 [i.13] specifies 2 operations (Read a single PM job and Query PM jobs) to realize Query VNF operation as clause A.7 in ETSI GS NFV-SOL 003 [i.11], and Get for PM jobs operation is corresponding to Query PM jobs operation.

Table 5.3.2.3.5-1: Comparison of input parameters in Query PM job and Get for PM jobs operation

Input Parameters in Query PM job operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Get for PM jobs operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
filter	1			Attribute-based filtering in the REST Interface, i.e. GET .../pm_jobs?filter=<filterExpr> See note.
				Attribute selector in REST interface, i.e. all_fields, fields, exclude_fields and exclude_default. See note.
NOTE: OpenStack Tacker VNF PM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.3.5-2: Comparison of output parameters returned by Query PM job and Get for PM jobs operation

Output Parameters returned by Query PM job operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Get for PM jobs operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
pmJob	0..N	PmJob	0..N	See note. Analysis between pmJob and PmJob. See Table 5.3.2.3.4-3.
NOTE: OpenStack Tacker VNF PM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.3.6 Comparison and analysis of Update PM job callback and Modify a PM job

Modify PM Job operation is not specified in ETSI GS NFV-IFA 007 [i.4] but specified in ETSI GS NFV-SOL 003 [i.13] as Update PM job callback. Table 5.3.2.3.6-1 illustrates a comparison of the attributes in the Update PM job callback operation as specified in clause 6.4.3.3.4 in ETSI GS NFV-SOL 003 [i.13] and Modify a PM job as specified in [i.16].

NOTE : Current OpenStack Tacker VNF PM API is based on the ETSI GS NFV-SOL 003 V3.3.1 [i.13] and ETSI GS NFV-SOL 013 V3.4.1 [i.17], and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF PM API are likely to be identified due to the version mismatch.

Table 5.3.2.3.6-1: Comparison of input parameters in Update PM job callback and Modify a PM job operation

Input Parameters in Update PM job callback operation in ETSI GS NFV-SOL 003 [i.13]		Attributes in Modify a PM job operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
pmJobId in resource URI, ../pm_jobs/{pmJobId}	1	pmJobId in resource URI, ../pm_jobs/{pmJobId}	1	
PmJobModifications	1	PmJobModifications	1	
> callbackUri	0..1	> callbackUri	0..1	
> authentication	0..1	> authentication	0..1	

Table 5.3.2.3.6-2: Comparison of output parameters returned by Update PM job callback and Modify a PM job operation

Output Parameters returned by Update PM job callback operation in ETSI GS NFV-SOL 003 [i.13]		Attributes in Modify a PM job operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
PmJobModifications	1	PmJobModifications	1	
> callbackUri	0..1	> callbackUri	1	Correspond with specification of "callbackUri" attribute that does not permit "null" in ETSI GS NFV-SOL 003 [i.13].
> authentication	0..1			Correspond with specification of "authentication" attribute that is not present in response bodies in ETSI GS NFV-SOL 003 [i.13].

5.3.2.3.7 Comparison and analysis of PerformanceReport and Get individual performance report

Table 5.3.2.3.7-1 illustrates a comparison of the attributes in PerformanceReport information element as specified in clause 8.7.5 in ETSI GS NFV-IFA 007 [i.4] and Get individual performance report operation as specified in [i.16].

NOTE 1: Current OpenStack Tacker VNF PM API is based on the ETSI GS NFV-SOL 003 V3.3.1 [i.13], and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF PM API are likely to be identified due to the version mismatch.

NOTE 2: ETSI GS NFV-IFA 007 [i.4] does not specify delivery mechanism for the PerformanceReport but specify only information element of PerformanceReport.

Table 5.3.2.3.7-1: Comparison of input parameters in PerformanceReport information element and Get individual performance report operation

Attributes in PerformanceReport information element in ETSI GS NFV-IFA 007 [i.4]		Attributes in Get individual performance report operation		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
		pmJobId		pmJobId in resource URI, ../pm_jobs/{pmJobId}/reports/{reportId} See note.
		reportId		reportId in resource URI, ../pm_jobs/{pmJobId}/reports/{reportId} See note.
performanceReport	1..N	PerformanceReport	1	Combining "PerformanceReport" and "entries" attributes fulfil cardinality "1..N" of performanceReport. See note.
		> entries	1..N	Combining "PerformanceReport" and "entries" attributes fulfil cardinality "1..N" of performanceReport. See note.
> objectType	1	>> objectType	1	
> objectInstanceId	1	>> objectInstanceId	1	
		>> subObjectInstanceId	0..1	See note.
> performanceMetric	1	>> performanceMetric	1	
> performanceValue	1..N	>> performanceValues	1..N	See note.
NOTE: OpenStack Tacker VNF PM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.4 Gap analysis related to VNF fault management

5.3.2.4.1 Introduction of the related work

The fault management interface of OpenStack® Tacker VNF FM API as defined in [i.18] manages the VNF fault management operations of VNF instances. This interface allows the NFVO to invoke VNF fault management operations of VNF instances towards the VNFM. The operations as provided through this interface are:

- Create a subscription
- Delete a subscription
- Get a subscription
- Get all subscriptions
- Get the individual alarm
- Get all alarms
- Modify the confirmation status

5.3.2.4.2 Comparison and analysis of Subscribe and Create a subscription

Table 5.3.2.4.2-1 illustrates a comparison of the attributes in Subscribe operation as specified in clause 7.5.2 in ETSI GS NFV-IFA 007 [i.4] and Create a subscription as specified in [i.18].

NOTE: Current OpenStack Tacker VNF FM API is based on the ETSI GS NFV-SOL 003 V3.3.1 [i.13] and ETSI GS NFV-SOL 013 v3.4.1 [i.17], and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF FM API are likely to be identified due to the version mismatch.

Table 5.3.2.4.2-1: Comparison of input parameters in Subscribe and Create a subscription operation

Input Parameters in Subscribe operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Create a subscription operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
filter	1	filter	0..1	See note 1.
		> vnfInstanceSubscriptionFilter	0..1	See note 1.
		> notificationTypes	0..N	See note 1.
		> faultyResourceTypes	0..N	See note 1.
		> perceivedSeverities	0..N	See note 1.
		> eventTypes	0..N	See note 1.
		> probableCauses	0..N	See note 1.
		callbackUri	1	See note 1.
		authentication	0..1	See notes 1 and 2.
NOTE 1: OpenStack Tacker VNF FM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4]. NOTE 2: OpenStack Tacker VNF FM API follows ETSI GS NFV-SOL 013 v3.4.1 [i.17].				

Table 5.3.2.4.2-2: Comparison of output parameters returned by Subscribe and Create a subscription operation

Output Parameters returned by Subscribe operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Create a subsription operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
subscriptionId	1	id	1	See note.
		filter	0..1	See note.
		> vnfInstanceSubscriptionFilter	0..1	See note.
		> notificationTypes	0..N	See note.
		> faultyResourceTypes	0..N	See note.
		> perceivedSeverities	0..N	See note.
		> eventTypes	0..N	See note.
		> probableCauses	0..N	See note.
		callbackUri	1	See note.
		_links	1	See note.
NOTE: OpenStack Tacker VNF FM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.4.3 Comparison and analysis of Terminate Subscription and Delete a subscription

Table 5.3.2.4.3-1 illustrates a comparison of the attributes in Terminate Subscription operation as specified in clause 7.5.5 in ETSI GS NFV-IFA 007 [i.4] and Delete a subscription as specified in [i.18].

NOTE: Current OpenStack Tacker VNF FM API is based on the ETSI GS NFV-SOL 003 V3.3.1 [i.13], and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF FM API are likely to be identified due to the version mismatch.

Table 5.3.2.4.3-1: Comparison of input parameters in Terminate Subscription and Delete a subscription operation

Input Parameters in Terminate Subscription operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Delete a subscription operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
subscriptionId	1			Maps to subscriptionId in the REST Interface, i.e. DELETE .../subscriptions/{subscriptionId} See note.
NOTE: OpenStack Tacker VNF FM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.4.3-2: Comparison of output parameters returned by Terminate Subscription and Delete a subscription operation

Output Parameters returned by Terminate Subscription operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Delete a subscription operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
N/A		N/A		

5.3.2.4.4 Comparison and analysis of Query Subscription Info and Get a subscription

Table 5.3.2.4.4-1 illustrates a comparison of the attributes in Query Subscription Info operation as specified in clause 7.5.6 in ETSI GS NFV-IFA 007 [i.4] and Get a subscription as specified in [i.18].

NOTE 1: Current OpenStack Tacker VNF FM API is based on the ETSI GS NFV-SOL 003 V3.3.1 [i.13], and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF FM API are likely to be identified due to the version mismatch.

NOTE 2: ETSI GS NFV-SOL 003 [i.13] specifies 2 operations (Read an individual subscription and Query multiple subscriptions) to realize Query Subscription Info operation as clause A.8 in ETSI GS NFV-SOL 003 [i.13], and Get a subscription operation is corresponding to Read an individual subscription operation.

Table 5.3.2.4.4-1: Comparison of input parameters in Query Subscription Info and Get a subscription operation

Input Parameters in Query Subscription Info operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Get a subscription operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
filter	1			Maps to subscriptionId in the REST Interface, i.e. GET .../subscriptions/{subscriptionId} See note.
NOTE: OpenStack Tacker VNF FM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.4.4-2: Comparison of output parameters returned by Query Subscription Info and Get a subscription operation

Output Parameters returned by Query Subscription Info operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Get a subscription operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
queryResult	0..N	FmSubscription	1	See note.
NOTE: OpenStack Tacker VNF FM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.4.4-3: Comparison of information model of queryResult and FmSubscription

Attributes in data type FmSubscription of queryResult in ETSI GS NFV-SOL 003 [i.13] (see note)		Attributes in FmSubscription		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
id	1	id	1	
filter	0..1	filter	0..1	
> vnfInstanceSubscriptionFilter	0..1	> vnfInstanceSubscriptionFilter	0..1	
> notificationTypes	0..N	> notificationTypes	0..N	
> faultyResourceTypes	0..N	> faultyResourceTypes	0..N	
> perceivedSeverities	0..N	> perceivedSeverities	0..N	
> eventTypes	0..N	> eventTypes	0..N	
> probableCauses	0..N	> probableCauses	0..N	
callbackUri	1	callbackUri	1	
_links	1	_links	1	
NOTE: Data type of queryResult is "Not Specified" in ETSI GS NFV-IFA 007 [i.4] and data model of queryResult is specified in ETSI GS NFV-SOL 003 [i.13] as FmSubscription.				

5.3.2.4.5 Comparison and analysis of Query Subscription Info and Get all subscriptions

Table 5.3.2.4.5-1 illustrates a comparison of the attributes in Query Subscription Info operation as specified in clause 7.5.6 in ETSI GS NFV-IFA 007 [i.4] and Get all subscriptions as specified in [i.18].

NOTE 1: Current OpenStack Tacker VNF FM API is based on the ETSI GS NFV-SOL 003 V3.3.1 [i.13], and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF FM API are likely to be identified due to the version mismatch.

NOTE 2: ETSI GS NFV-SOL 003 [i.13] specifies 2 operations (Read an individual subscription and Query multiple subscriptions) to realize Query Subscription Info operation as clause A.8 in ETSI GS NFV-SOL 003 [i.13], and Get all subscriptions operation is corresponding to Query multiple subscriptions operation.

Table 5.3.2.4.5-1: Comparison of input parameters in Query Subscription and Get all subscriptions operation

Input Parameters in Query Subscription Info operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Get all subscriptions operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
filter	1			Attribute-based filtering in the REST Interface, i.e. GET .../subscriptions?filter=<filterExpr> See note.
				Attribute selector in REST interface, i.e. all_fields, fields, exclude_fields and exclude_default. See note.
NOTE: OpenStack Tacker VNF PM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.4.5-2: Comparison of output parameters returned by Query Subscription Info and Get all subscriptions operation

Output Parameters returned by Query Subscription Info operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Get all subscriptions operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
queryResult	0..N	FmSubscription	0..N	See note. Analysis between queryResult and FmSubscription. See Table 5.3.2.4.4-3.
NOTE : OpenStack Tacker VNF PM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.4.6 Comparison and analysis of Get Alarm List and Get the individual alarm

Table 5.3.2.4.6-1 illustrates a comparison of the attributes in Get Alarm List operation as specified in clause 7.5.4 in ETSI GS NFV-IFA 007 [i.4] and Get the individual alarm as specified in [i.18].

NOTE 1: Current OpenStack Tacker VNF FM API is based on the ETSI GS NFV-SOL 003 V3.3.1 [i.13], and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF FM API are likely to be identified due to the version mismatch.

NOTE 2: ETSI GS NFV-SOL 003 [i.13] specifies 2 operations (Read individual alarm and Query alarms) to realize Get Alarm List operation as clause A.8 in ETSI GS NFV-SOL 003 [i.13], and Get the individual alarm operation is corresponding to Read individual alarm operation.

Table 5.3.2.4.6-1: Comparison of input parameters in Get Alarm List and Get the individual alarm operation

Input Parameters in Get Alarm List operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Get the individual alarm operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
filter	1			Maps to alarmId in the REST Interface, i.e. GET .../alarms/{alarmId} See note.
NOTE: OpenStack Tacker VNF FM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.4.6-2: Comparison of output parameters returned by Get Alarm List and Get the individual alarm operation

Output Parameters returned by Get Alarm List operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Get the individual alarm operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
alarm	0..N	Alarm	1	See note.
NOTE: OpenStack Tacker VNF FM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.4.6-3: Comparison of information model of alarm and Alarm

Attributes in data type FmSubscription of queryResult in ETSI GS NFV-SOL 003 [i.13]		Attributes in FmSubscription		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
alarmId	1	id	1	See note.
managedObjectId	1	managedObjectId	1	
		vnfcInstanceIds	0..N	Alarm data type in ETSI GS NFV-SOL 002 [i.31] includes vnfcInstanceIds as difference between ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13] described in clause D.2.4 in ETSI GS NFV-SOL 002 [i.31], and OpenStack Tacker merged VNF LCM API of both ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13] as OpenStack Tacker proprietary solution.
rootCauseFaultyResource	0..1	rootCauseFaultyResource	0..1	
alarmRaisedTime	1	alarmRaisedTime	1	
alarmChangedTime	0..1	alarmChangedTime	0..1	
alarmClearedTime	0..1	alarmClearedTime	0..1	
		alarmAcknowledgedTime	0..1	See note.
ackState	1	ackState	1	
perceivedSeverity	1	perceivedSeverity	1	
eventTime	1	eventTime	1	
eventType	1	eventType	1	
faultType	0..1	faultType	0..1	
probableCause	1	probableCause	1	
isRootCause	1	isRootCause	1	
correlatedAlarmId	0..N	correlatedAlarmId	0..N	
faultDetails	0..N	faultDetails	0..N	
		_links	1	See note.
NOTE: OpenStack Tacker VNF FM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.4.7 Comparison and analysis of Get Alarm List and Get all alarms

Table 5.3.2.4.7-1 illustrates a comparison of the attributes in Get Alarm List operation as specified in clause 7.5.4 in ETSI GS NFV-IFA 007 [i.4] and Get all alarms as specified in [i.18].

NOTE 1: Current OpenStack Tacker VNF FM API is based on the ETSI GS NFV-SOL 003 V3.3.1 [i.13], and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF FM API are likely to be identified due to the version mismatch.

NOTE 2: ETSI GS NFV-SOL 003 [i.13] specifies 2 operations (Read individual alarm and Query alarms) to realize Get Alarm List operation as clause A.8 in ETSI GS NFV-SOL 003 [i.13], and Get all alarms operation is corresponding to Query alarms operation.

Table 5.3.2.4.7-1: Comparison of input parameters in Get Alarm List and Get all alarms operation

Input Parameters in Get Alarm List operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Get all alarms operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
filter	1			Attribute-based filtering in the REST Interface, i.e. GET .../alarms?filter=<filterExpr> See note.
				Attribute selector in REST interface, i.e. all_fields, fields, exclude_fields and exclude_default. See note.
NOTE: OpenStack Tacker VNF FM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.4.7-2: Comparison of output parameters returned by Get Alarm List and Get all alarms operation

Output Parameters returned by Get Alarm List operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Get all alarms operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
alarm	0..N	Alarm	0..N	See note. Analysis between alarm and Alarm. See Table 5.3.2.4.6-3.
NOTE: OpenStack Tacker VNF FM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.4.8 Comparison and analysis of Acknowledge alarms and Modify the confirmation status

Table 5.3.2.4.8-1 illustrates a comparison of the attributes in Acknowledge alarms operation as specified in clause 7.5.7 in ETSI GS NFV-IFA 007 [i.4] and Modify the confirmation status as specified in [i.18].

NOTE: Current OpenStack Tacker VNF FM API is based on the ETSI GS NFV-SOL 003 V3.3.1 [i.13], and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF FM API are likely to be identified due to the version mismatch.

Table 5.3.2.4.8-1: Comparison of input parameters in Acknowledge alarms and Modify the confirmation status operation

Input Parameters in Acknowledge alarms operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Modify the confirmation status operation		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
alarmId	1			Maps to alarmId in the REST Interface, i.e. DELETE .../alarms/{alarmId} See note
		ackState	1	See note
NOTE : OpenStack Tacker VNF FM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

Table 5.3.2.4.8-2: Comparison of output parameters returned by Acknowledge alarms and Modify the confirmation status operation

Output Parameters returned by Acknowledge alarms operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Modify the confirmation status operation		Comments
Parameter	Cardinality	Attribute	Cardinality	
acknowledgeAlarmId				Maps to alarmId in the REST Interface, i.e. .../alarms/{alarmId} See note
		ackState	1	See note
NOTE: OpenStack Tacker VNF FM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13], which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4].				

5.3.2.5 Gap analysis related to Grant

5.3.2.5.1 Introduction of the related work

OpenStack® Tacker requests Grant to NFVO as VFNM as described in [i.19]. This interface supports for VNFM to interoperate with ETSI NFV SOL compliant NFVO to invoke VNF lifecycle management operations of VNF instances.

- Grants

5.3.2.5.2 Comparison and analysis of Grant VNF Lifecycle and Grants

Table 5.3.2.5.2-1 illustrates a comparison of the attributes in Grant VNF Lifecycle operation as specified in clause 6.3.2 in ETSI GS NFV-IFA 007 [i.4] and Grants as described in [i.19].

NOTE: Current OpenStack Tacker VNF LCM API is based on the ETSI GS NFV-SOL 003 (V3.3.1) [i.13], and it is the version used for the comparison analysis. Gaps in the OpenStack® Tacker VNF LCM API are likely to be identified due to the version mismatch.

Table 5.3.2.5.2-1: Comparison of input parameters in Grant VNF Lifecycle and Grants operation

Input Parameters in Grant VNF Lifecycle operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Grants		Comments
Parameter	Cardinality	Attribute	Cardinality	
vnfInstanceId	1	vnfInstanceId	1	
vnfdId	1	vnfdId	1	
dstVnfdId	0..1			No correspondence.
flavourId	0..1	flavourId	0..1	
lifecycleOperation	1	operation	1	See note 1.
isAutomaticInvocation	1	isAutomaticInvocation	1	
lifecycleOperationOccurrenceId	1	vnfLcmOpOccId	1	See note 1.
instantiationLevelId	0..1			No correspondence.
targetScaleLevelInfo	0..N			No correspondence. See note 2.
addResource	0..N	addResource	0..N	
tempResource	0..N			No correspondence.
removeResource	0..N	removeResource	0..N	
updateResource	0..N			No correspondence.
placementConstraint	0..N	placementConstraint	0..N	
vimConstraint	0..N			No correspondence.
additionalParam	0..N			No correspondence.
		_links	1	See note 1.
NOTE 1: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13] which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4]. NOTE 2: V3.3.1 of ETSI GS NFV-SOL 003 [i.13] does not include this attribute.				

Table 5.3.2.5.2-2: Comparison of output parameters returned by Grant VNF Lifecycle and Grants operation

Output Parameters returned by Grant VNF Lifecycle operation in ETSI GS NFV-IFA 007 [i.4]		Attributes in Grants		Comments
Parameter	**Cardinality**	**Attribute**	**Cardinality**	
		id	1	See note 1.
		vnfInstanceId	1	See note 1.
		vnfLcmOpOccId	1	See note 1.
vimConnection	0..N	vimConnections	0..N	See note 1.
cirConnectionInfo	0..N			No correspondence. See note 2. "helm_parameters" of "additionalParams" attribute can be set URL of CIR when install helm described in Table 5.3.1.3-1 as OpenStack Tacker proprietary solution.
mciopRepositoryInfo	0..N			No correspondence. See note 2. "helm_chart_path" of "additionalParams" attribute can be set file path of helm chart when install helm described in Table 5.3.1.3-1 as OpenStack Tacker proprietary solution.
zone	0..N	zones	0..N	See note 1.
zoneGroup	0..N			No correspondence.
addResource	0..N	addResources	0..N	See note 1.
tempResource	0..N			No correspondence.
removeResource	0..N	removeResources	0..N	See note 1.
updateResource	0..N			No correspondence.
vimAssets	0..1	vimAssets	0..1	
extVirtualLink	0..N			No correspondence.
extManagedVirtualLink	0..N			No correspondence.
additionalParam	0..N	additionalParams	0..1	See note 1.
		_links	1	See note 1.
NOTE 1: OpenStack Tacker VNF LCM API follows ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13] which as protocol and data model solutions provide more details than ETSI GS NFV-IFA 007 [i.4]. NOTE 2: V3.3.1 of ETSI GS NFV-SOL 003 [i.13] does not include this attribute.				

5.3.2.6 Gaps and Potential solutions

5.3.2.6.1 Gap# 1: gap related to parameter related information in VNF LCM API

The gaps of OpenStack® Tacker proprietary solution are shown in Table 5.3.2.6.1-1.

Table 5.3.2.6.1-1: the gaps of OpenStack® Tacker proprietary solution

Table No.	operation	parameter	purpose
Table 5.3.2.2.6-4	Query VNF	VNFInstance.instantiatedVnfInfo.metadata	To use other lifecycle operation after successful instantiation is stored
		VNFInstance.instantiatedVnfInfo.vnfcInfo	To support VNF LCM API of both ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13]
Table 5.3.2.2.9-1	Heal VNF	all	To indicate whether network resource and storage resource are included in this heal operation
		vnfcInstanceId	To support VNF LCM API of both ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003
Table 5.3.2.2.12-1	Modify VNF Information	vnfcInfoModifications	To support VNF LCM API of both ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003
Table 5.3.1.3-1	Change current VNF package operation	upgrade_type	To specify Blue-Green or Rolling update for change current VNF package operation

The gaps that OpenStack® Tacker does not support except for version mismatch are shown in Table 5.3.2.6.1-2.

Table 5.3.2.6.1-2: unsupported GAP by OpenStack® Tacker

Table No.	operation	parameter	reason
Table 5.3.2.2.6-4	Query VNF	VNFInstance.instantiatedVnfInfo.monitoringParameter	OpenStack® Tacker does not support corresponding attributes in VNFD to expose monitoringParameter in the current version. OpenStack® Tacker may implement it in future version.
		VNFInstance.instantiatedVnfInfo.localizationLanguage	OpenStack® Tacker does not support corresponding attributes in VNFD to expose localizationLanguage in the current version. OpenStack® Tacker may implement it in future version.
Table 5.3.2.2.11-1	Change current VNF Package	extension	OpenStack® Tacker does not have usecase to use extension in Change current VNF Package operation. Tacker may implement it if usecase become clear.
		vnfConfigurableProperties	OpenStack® Tacker realizes VNF Configuration by Management Driver described in OpenStack® Tacker ETSI NFV-SOL Tacker Use Cases [i.20] as Tacker proprietary solution which is another way from ETSI GS NFV-SOL 002 [i.31], and does not have usecase to use vnfConfigurableProperties in Change current VNF Package operation in the current version. Tacker may implement it after implement VNF Configuration based on ETSI GS NFV-SOL 002 [i.31].

5.3.2.6.2 Gap# 2: gap related to parameter related information in VNF PM API

The gaps of OpenStack® Tacker proprietary solution are shown in Table 5.3.2.6.2-1.

Table 5.3.2.6.2-1: the gaps of OpenStack® Tacker proprietary solution

Table No.	Operation	parameter	purpose
Table 5.3.2.3.2-1	Create PM job	metadata	To input access information of external monitoring tool.

5.3.2.6.3 Gap# 3: gap related to parameter related information in VNF FM API

The gaps of OpenStack® Tacker proprietary solution are shown in Table 5.3.2.6.3-1.

Table 5.3.2.6.3-1: the gaps of OpenStack® Tacker proprietary solution

Table No.	Operation	parameter	purpose
Table 5.3.2.4.6-3	Alarm	vnfcInstanceIds	To support VNF LCM API of both ETSI GS NFV-SOL 002 [i.31] and ETSI GS NFV-SOL 003 [i.13].

5.4 Gap analysis with Kubernetes®

5.4.1 Introduction of Kubernetes®

Kubernetes® is a portable, extensible open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. Kubernetes® has a number of features, it provides a container-centric management environment and can orchestrate computing, networking, and storage resources on behalf of user workloads. This provides much of the simplicity of Platform as a Service (PaaS) with the flexibility of Infrastructure as a Service (IaaS), therefore Kubernetes® can be considered as a container platform, a microservice platform, as well as a portable cloud platform.

NOTE: In this clause, "Not Applicable" means that the attribute is not applicable when the Vdu is realized by a set of OS containers.

5.4.2 Gap analysis

5.4.2.1 Gap analysis related to VNFD

5.4.2.1.1 Introduction of the related work

According to clause 5.2.2.2 of ETSI GS NFV-IFA 040 [i.26], the VNF package for a containerized VNF contains both the VNFD and one or multiple MCIOPs and the VNFD of a containerized VNF has references to one or multiple MCIOPs which contain declarative descriptors and configuration files for MCIOs consumed by the containerized VNF.

The purpose of this gap analysis is twofold:

- Identify whether the information elements contained in a VNFD as currently specified in ETSI GS NFV-IFA 011 [i.3] would be sufficient to enable communication between a VNFM and a CISM if no MCIOPs were present in the VNF package.
- Identify whether the Kubernetes® object model is lacking information elements to convey all requirements and constraints specified in a VNFD.

The Information Elements (IEs) contained in a VNFD can be classified in two categories. VNF-level IEs, that are used to manage the lifecycle of a VNF as a whole, and constituent-level IEs, that are used to manage the lifecycle of the VNF constituents (VNFCs, VLs, etc.). The information contained in VNF-Level IEs is not intended to be processed by a CISM.

Today, the Kubernetes API does not allow to manage applications, i.e. abstract constructs representing a composition of resources such as a VNF. Therefore, VNF-level IEs do not have a correspondence with any Kubernetes IE.

The shaded area on Figure 5.4.2.1.1-1 shows the composition-related data, as opposed to constituent-level data. That is, information related to the configuration (architecture and behaviour) of the VNF as a whole, as opposed to information dedicated to one of its constituents. Although the VnfExtCpd is represented within the shared area, it can also be regarded as describing a constituent-level element in some cases (e.g. when mapped to VirtualCpd or a VduCpd).

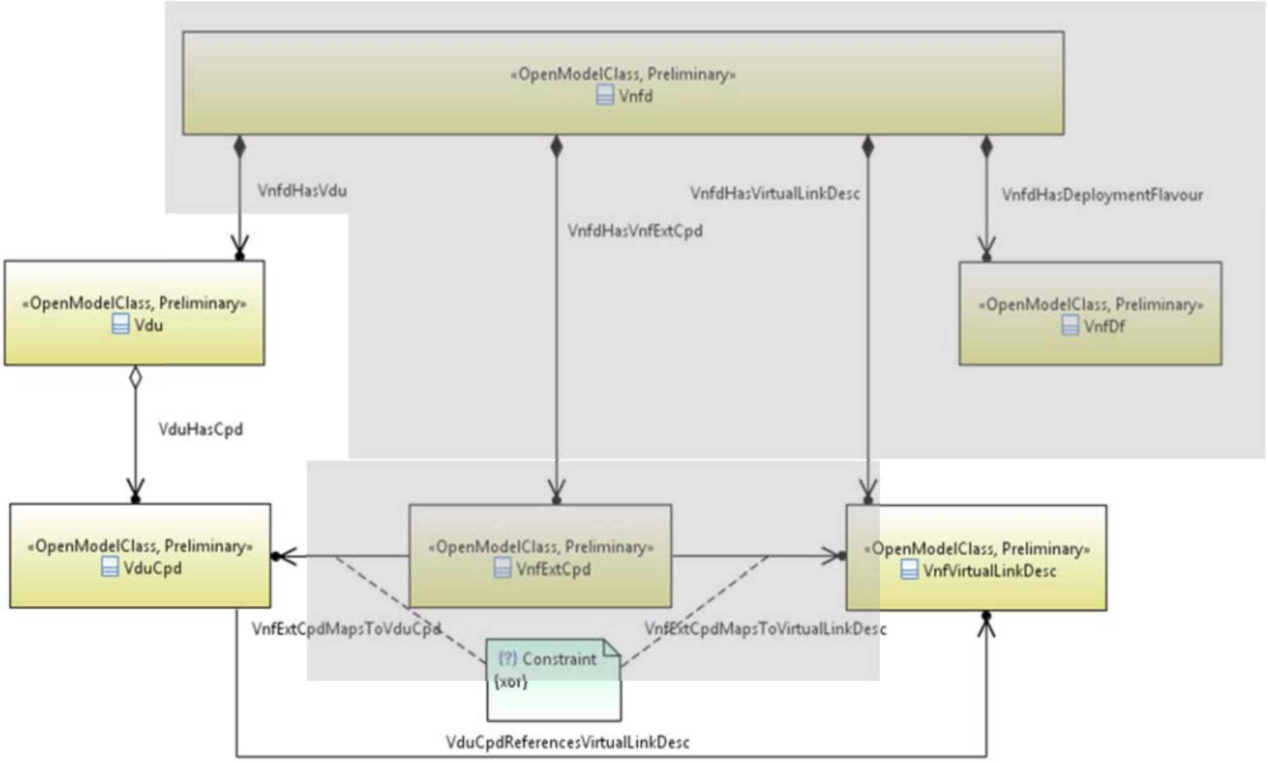
«OpenModelClass, Preliminary» Vnfd
VnfdHasVdu
VnfdHasVnfExtCpd
VnfdHasVirtualLinkDesc
VnfdHasDeploymentFlavour
«OpenModelClass, Preliminary» Vdu
«OpenModelClass, Preliminary» VnfDf
VduHasCpd
«OpenModelClass, Preliminary» VduCpd
«OpenModelClass, Preliminary» VnfExtCpd
«OpenModelClass, Preliminary» VnfVirtualLinkDesc
VnfExtCpdMapsToVduCpd
VnfExtCpdMapsToVirtualLinkDesc
{?} Constraint
{xor}
VduCpdReferencesVirtualLinkDesc

Figure 5.4.2.1.1-1: Composition-related data vs resource-level data

As analysed in the following clauses, the VNFD mapping to Kubernetes manifest data, covers most of the VNF virtual resources configuration data, but very few of the VNF software configuration data.

5.4.2.1.2 Comparison and analysis of VNFD and Kubernetes information model

Table 5.4.2.1.2-1 illustrates a comparison of the attributes in VNFD as specified in clause 7.1.2 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.2-1: Comparison for the VNFD top-level information element

Attributes of Vnfd information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api*.v1.		Comments
Attribute	Cardinality	Attribute	Cardinality	
vnfdId	1			No correspondence. See note 1.
vnfdExtInvariantId	0..1			No correspondence. See note 1.
vnfProvider	1			No correspondence. See note 1.
vnfProductName	1			No correspondence. See note 1.
vnfSoftwareVersion	1			No correspondence. See note 1.
vnfdVersion	1			No correspondence. See note 1.
vnfProductInfoName	0..1			No correspondence. See note 1.
vnfProductInfoDescription	0..1			No correspondence. See note 1.

Attributes of Vnfd information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api*.v1.		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
vnfmInfo	1..N			No correspondence. See note 1.
localizationLanguage	0..N			No correspondence. See note 1.
defaultLocalizationLanguage	0..1			No correspondence. See note 1.
Vdu	1..N			See note 2 and clause 5.4.2.1.4.
virtualComputeDesc	0..N	N/A	N/A	Not applicable.
virtualStorageDesc	0..N			See clause 5.4.2.1.18 and note 2.
osContainerDesc	0..N			See note 2 and clause 5.4.2.1.17.
swImageDesc	0..N			See clause 5.4.2.1.23 and note 2.
intVirtualLinkDesc	0..N			See clause 5.4.2.1.16.
securityGroupRule	0..N			See note 2 and clause 5.4.2.1.23.
vnfExtCpd	1..N			Conditional correspondence. See note 3.
vipCpd	0..N			See note 2 and clause 5.4.2.1.12.
virtualCpd	0..N			See note 2 and clause 5.4.2.1.13.
deploymentFlavour	1..N			No correspondence. See note 1.
configurableProperties	0..1			See note 4.
modifiableAttributes	0..1			See note 4.
lifeCycleManagementScript	0..N			No correspondence. See note 1.
vnfIndicator	0..N			No correspondence. See note 1.
autoScale	0..N			No correspondence. See note 1.
vnfPackageChangeInfo	0..N			No correspondence. See note 1.
lcmOperationCoordination	0..N			No correspondence. See note 1.
mciopId	0..N			Maps to Helm Chart names if the CISM OS container workload management service interface is used, no correspondence otherwise. See note 1.
NOTE 1: The content of this VNF-level IE is not intended to be sent to the CISM/VIM. NOTE 2: Although represented as a VNF-level IE in the VNFD information model, this attribute contains information applicable to the VNF constituents. Instances of this attribute are referenced from constituent-level IEs. For an assessment of the correspondence with Kubernetes objects, refer to the comparison table dedicated to the corresponding IE and to the tables applicable to the IEs where this attribute is referenced. NOTE 3: No correspondence if the VnfExtCps are attached to internal virtual link. Otherwise, see clause 5.4.2.1.3. NOTE 4: The analysis is not covered by the current version of the present document.				

5.4.2.1.3 VNF external connection point

In "VnfExtCp", the term 'Ext' is relative to the concept of VNF. So, no equivalent information is expected on CISM side (see Table 5.4.2.1.2-1). However, some of its attributes are IEs describing the resource level Cps to be exposed as VNF external CPs. How these IEs may correspond to some Kubernetes IE, described in the relevant IE table.

Table 5.4.2.1.3-1 illustrates a comparison of the attributes the VnfExtCpd information element as specified in clause 7.1.3 in ETSI GS NFV-IFA 011 [i.3] and the Kubernetes® information model as specified in [i.10].

Table 5.4.2.1.3-1: Comparison for the VnfExtCpd information element

Attributes of VnfExtCpd information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api*.v1.		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
intVirtualLinkDesc	0..1			No correspondence. See note 1.
intCpd	0..1			No correspondence. See note 1.
vipCpd	0..1			No correspondence.
virtualCpd	0..1			No correspondence. See clause 5.4.2.1.13.
virtualNetworkInterfaceRequirements	0..N			Not applicable for primary container cluster networks (see note 2). See note 3 for connections to secondary container cluster networks.
(Inherited attributes)	1			No correspondence. See note 1.
NOTE 1: No equivalent to VNF-level information is expected on CISM side. NOTE 2: Not applicable as per ETSI GS NFV-IFA 011 [i.3], Table 7.1.3.2.2-1, note 5. NOTE 3: For an assessment of the correspondence with Kubernetes objects, see clause 5.4.2.1.11.				

5.4.2.1.4 Vdu

The shaded area on Figure 5.4.2.1.4-1 shows the Vdu related IE that are not natively managed by the Kubernetes CISM.

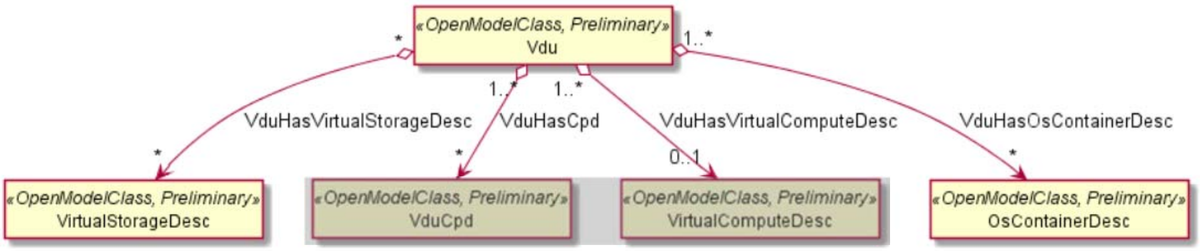

Figure 5.4.2.1.4-1: VDU-related IEs not natively managed by Kubernetes

Table 5.4.2.1.4-1 illustrates a comparison of the attributes of the Vdu information element as specified in clause 7.1.6 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.4-1: Comparison for the Vdu information element

Attributes of Vdu information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element in io.k8s.api.apps*.v1. [i.10]		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
vduId	1			No correspondence. See note 1.
name	1		0..1	No Correspondence.
description	1			No correspondence. See note 2.
intCpd	0..N	- --- (Deployment / StatefulSet)[]. metadata.annotations. k8s.v1.cni.cncf.io/networks[]. (name / (name, namespace))		No correspondence if the Vdu containers only connect to primary container cluster networks. Correspondence for secondary container cluster networks. See notes 3 and 7.
virtualComputeDesc	0..1	N/A	N/A	Not applicable. See note 4.
osContainerDesc	0..N	(Deployment / StatefulSet). spec.template.spec.containers[]		Correspondence. (Correspondence is partial for entries, see notes 5 and 7.)
virtualStorageDesc	0..N	Deployment.spec.template.spec. volumes[].persistentVolumeClaim. claimName / StatefulSet.spec. volumeClaimTemplates[]. metadata.name	1 0..1	Feasible correspondence. (Correspondence is partial for entries.) See notes 6 and 7.
bootOrder	0..N	N/A	N/A	Not applicable. See note 4.
swImageDesc	0..1	N/A	N/A	Not applicable. See note 4.
nfviConstraint	0..N			See note 8.
monitoringParameter	0..N	N/A	N/A	Not applicable. See note 4.
configurableProperties	0..1			See note 8.
bootData	0..1	N/A	N/A	Not applicable. See note 4.
trunkPort	0..N	N/A	N/A	Not applicable.
logicalNode	0..N			See note 8.
requestAdditionalCapabilities	0..N			See note 8.
mcioConstraintParams	0..N	(Deployment / StatefulSet). spec.template.spec.affinity		Feasible correspondence. (As populated by Grant response). See note 7.
mcioIdentificationData	0..1	(Deployment / StatefulSet). metadata.name (Deployment / StatefulSet). metadata.kind	N/A	Correspondence.

Attributes of Vdu information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element in io.k8s.api.apps*.v1. [i.10]		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
NOTE 1: As per ETSI GS NFV-IFA 011 [i.3], VduId is the unique identifier of the Vdu in the Vnfd and is used to identify in the VNF runtime information stored by the VNFM the VDU that has been used to instantiate a VNFC instance. NOTE 2: However, spec.metadata.annotations may be an appropriate place for a 'description'. NOTE 3: This required key with value of type string is the name of a NetworkAttachmentDefinition object, either in the Pod's namespace (if the "namespace" key is missing or empty) or another namespace specified by the "namespace" key. NOTE 4: As per ETSI GS NFV-IFA 011 [i.3], Table 7.1.6.2.1-1, note 6, "Only one of virtualComputeDesc or osContainerDesc is part of a Vdu. If the Vdu includes osContainerDesc, bootOrder, swImageDesc, monitoringParameters and bootData is not present in the Vdu". NOTE 5: In ETSI GS NFV-SOL 001 [i.7], the K8S IE corresponds to the 'container' requirement in node templates of type OsContainerDeployableUnit. NOTE 6: In ETSI GS NFV-SOL 001 [i.7], the K8S IE corresponds to the 'virtual_storage' requirement in node templates of type OsContainerDeployableUnit. NOTE 7: For an assessment of the correspondence with Kubernetes objects, refer to the clauses applicable to the IEs where this attribute is referenced. NOTE 8: The analysis is not covered by the current version of the present document.				

5.4.2.1.5 VduCpd

A partial correspondence can be set with K8S CNI extensions for secondary container cluster networks.

Table 5.4.2.1.5-1 illustrates a comparison of the attributes of the VduCpd information element as specified in clause 7.1.6.3.1 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.5-1: Comparison for the VduCp information element

Attributes of VduCp information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.ap.apps*.v1. [i.10] and in io.cncf.api.cni.k8s/v1		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
intVirtualLinkDesc	0..1	- NetworkAttachmentDefinition. spec.config.plugins[].name & (Deployment / StatefulSet).spec. template.metadata.annotations. k8s.v1.cni.cncf.io/networks[0]. "name" or in short notation: (Deployment / StatefulSet).spec. template.metadata.annotations. k8s.v1.cni.cncf.io/networks	0..1	No correspondence for connection to primary container cluster networks. -------------------------------------- --- Possible correspondence for a connection to secondary container cluster networks, when supported by the CNI plugin in the cluster instance. For short notation, the network name is one of a comma-separated strings when the pod is attached based on several NetworkAttachmentDefinition s. See note 3.

Attributes of VduCp information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.ap.apps*.v1. [i.10] and in io.cncf.api.cni.k8s/v1		Comments
Attribute	Cardinality	Attribute	Cardinality	
bitrateRequirement	0..N	- --- (Deployment / StatefulSet).spec. template.metadata.annotations. k8s.v1.cni.cncf.io/networks[index]. "bandwidth". ("ingressRate" + "egressRate")	0..1	- No correspondence for connection to a primary container cluster network. -- - correspondence for connection to a secondary container cluster network, when supported by the CNI plugin in the cluster instance. See note 1.
virtualNetworkInterface Requirements	0..N	N/A		See note 2.
order	0..1	N/A		
vnicType	0..1	- --- NetworkAttachmentDefinition. plugins[0].type	0..1	No correspondence for connection to a primary container cluster network. -- Correspondence for connection to a secondary container cluster network, when supported by the CNI plugin in the cluster instance.
{Cp-inherited attributes}		N/A		See note 4.
NOTE 1: Where i denotes the range of the relevant CNI plugin in the chain specified by the corresponding NAD. NOTE 2: For an assessment of the correspondence with Kubernetes objects, see clause 5.4.2.1.11. NOTE 3: A NAD specifies how to attach a pod to the network. The NAD descriptor is identified by its name, in NetworkAttachmentDefinition.metadata.name. NOTE 4: For an assessment of the correspondence with Kubernetes objects, see clause 5.4.2.1.6.				

5.4.2.1.6 Cpd

A partial correspondence can be set with K8S CNI extensions for secondary container cluster networks.

Table 5.4.2.1.6-1 illustrates a comparison of the attributes the Cpd information element as specified in clause 7.1.6.4 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.6-1: Comparison for the Cpd information element

Attributes of Cpd information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api.{apps, networking}.v1. [i.10] and in io.cncf.api.cni.k8s/v1		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
cpdId	1	- --- (Deployment / StatefulSet)[]. metadata.annotations. k8s.v1.cni.cncf.io/networks[]. interface		No correspondence for connection to a primary container cluster network. ------------------------------------- Possible correspondence for connection to secondary container cluster network, when supported by the CNI plugin. See note 1.

layerProtocol	1..N	Examples correspondence:
		layerProtocol[0] = Ethernet ⇔ NetworkAttachmentDefinition.
		spec.config.plugins[].
		type = bridge
		isGateway = false
		isDefaultGateway = false.
				Partial correspondence on layerProtocol[0] for secondary cluster networks, when supported by the CNI plugin.
				See note 3.

cpRole	0..1	-		No correspondence.
description	0..1	-		No correspondence.
cpProtocol	0..N			No correspondence. See note 2.
trunkMode	0..1	N/A		trunkMode would equal 'false' for K8S.
securityGroupRuleId	0..N	NetworkPolicy.metadata.name and $index in NetworkPolicy.spec. (egress / ingress)[$index]		Possible correspondence. See clause 5.4.2.1.24.
NOTE 1: cpId intervenes in the relationship between the Cpd and another entity. For instance, in ETSI GS NFV-SOL 001 [i.7], it would identify the source of a VirtualLinksTo relationships, i.e. match the 'virtual_link' requirement in a *Cp node. NOTE 2: For an assessment of the correspondence with Kubernetes objects, refer to clause 5.4.2.1.7 where this attribute is referenced. NOTE 3: Whether a correspondence can be found for the top-level protocol in the protocol stack depends on the CNI plugin.				

5.4.2.1.7 CpProtocolData

Table 5.4.2.1.7-1 illustrates a comparison of the attributes the CpProtocolData information element as specified in clause 7.1.6.8 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.7-1: Comparison for the CpProtocolData information element

Attributes of CpProtocolData information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api*.v1. [i.10]		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
associatedLayerProtocol	1	-		No correspondence
addressData	0..N	N/A		See note
NOTE: For an assessment of the correspondence with Kubernetes objects, refer to clause 5.4.2.1.8.				

5.4.2.1.8 AddressData

A partial correspondence can be set with K8S CNI extensions for secondary container cluster networks.

Table 5.4.2.1.8-1 illustrates a comparison of the attributes the AddressData information element as specified in clause 7.1.3.3 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.8-1: Comparison for the AddressData information element

Attributes of AddressData information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.cncf.api.cni.k8s/v1		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
addressType	1	- --- NetworkAttachmentDefinition. spec.config.plugins[0]. type: bridge & isGateway: false & isDefaultGateway: false - NetworkAttachmentDefinition. spec.config.plugins[0]. ipam.type: <some_value>		No correspondence for connection to a primary container cluster network. ------------------------------------- Possible correspondence for connection to a secondary container cluster network, when supported by the CNI plugin: Correspondence for addressType: MAC address - Correspondence for addressType: IP address
L2AddressData	0..1	N/A		See note 1
L3AddressData	0..1	N/A		See note 2
NOTE 1: For an assessment of the correspondence with Kubernetes objects, refer to clause 5.4.2.1.10. NOTE 2: For an assessment of the correspondence with Kubernetes objects, refer to clause 5.4.2.1.9.				

5.4.2.1.9 L3AddressData

No correspondence can be set with K8S CNI extensions for secondary container cluster networks.

Table 5.4.2.1.9-1 illustrates a comparison of the attributes the L3AddressData information element as specified in clause 7.1.3.4 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.9-1: Comparison for the L3AddressData information element

Attributes of L3AddressData information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api.{core, apps}.v1. [i.10]		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
ipAddressAssignment	1			No correspondence. (Not meant for a CISM)
ipAddressAssignmentSubtype	0..1			No correspondence. (Not meant for a CISM)
floatingIpActivated	0..1		1 (of 4)	No correspondence. However, the concept of floating IP and its usage is similar to the purpose of Loadbalancer type of K8S service

Attributes of L3AddressData information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api.{core, apps}.v1. [i.10]		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
ipAddressType	0..1	Service.spec.ipFamilies[0] & cardinal(ipFamilies = 1) ---		Possible correspondence for connection to primary container cluster network when the attribute is part of a VirtualCp IE. ------------------------------------- No correspondence for connection to a secondary container cluster network.
numberOfIpAddress	0..1	-		No correspondence
fixedIpAddress	0..N	- -- (Deployment / StatefulSet)[].metadata.annotations. k8s.v1.cni.cncf.io/networks[].ips	-	No correspondence for connection to primary container cluster network ------------------------------------ Correspondence for connection to secondary container cluster network

5.4.2.1.10 L2AddressData

Table 5.4.2.1.10-1 illustrates a comparison of the attributes the L2AddressData information element as specified in clause 7.1.3.5 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.10-1: Comparison for the L2AddressData information element

Attributes of L2AddressData information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api.*.v1. [i.10]		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
macAddressAssignment	1			No correspondence. See note
NOTE: Boolean specifying whether the mac comes from MANO or VNF, so no correspondence is expected on the CISM side.				

5.4.2.1.11 VirtualNetworkInterfaceRequirements

Table 5.4.2.1.11-1 illustrates a comparison of the attributes the VirtualNetworkInterfaceRequirements information element as specified in clause 7.1.6.6 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.11-1: Comparison for the VirtualNetworkInterfaceRequirements information element

Attributes of VirtualNetworkInterfaceRequirements information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api*.v1. [i.10]		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
name	0..1	-		No correspondence
description	0..1	-		No correspondence. (But could be passed as a pod template annotation)
standardizedNetworkInterfaceRequirements	0..1			
networkInterfaceRequirements	0..1			
nicIoRequirements	0..1			
NOTE: The analysis is not covered by the current version of the present document.				

5.4.2.1.12 VipCpd

VipCpd related analysis is not covered by the current version of the present document.

5.4.2.1.13 VirtualCpd

Table 5.4.2.1.13-1 illustrates a comparison of the attributes of the VirtualCpd information element as specified in clause 7.1.18.2 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.13-1: Comparison for the VirtualCpd information element

Attributes of VirtualCp information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api.core.v1. [i.10]		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
vdu	1..N	Service.spec.selector	1..N	Correspondence
additionnalServiceData	0..N	N/A		See clause 5.4.2.1.14
{Cp-inherited attributes}		N/A		See clause 5.4.2.1.6

5.4.2.1.14 AdditionalServiceData

Table 5.4.2.1.14-1 illustrates a comparison of the attributes of the AdditionalServiceData information element as specified in clause 7.1.18.3 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.14-1: Comparison for the AdditionnalServiceData information element

Attributes of AdditionalServiceData information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api.networking.v1. [i.10]		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
portData	1..N	N/A		See clause 5.4.2.1.15
serviceData	0..1	If VirtualCp.additionalServiceData [ ]. protocol = (http / https), then: Ingress.spec.rules[].host and/or Ingress.spec.rules[].paths[ ].path --		Possible correspondence for connection to primary container cluster network when the attribute is part of a VirtualCp IE. See ETSI GS NFV-SOL 018 [i.27], clause 6.2.3.1 and note. ------------------------------------ No correspondence for connection to a secondary container cluster network.
NOTE: 'http' and 'https' are evoked in ETSI GS NFV-SOL 018 [i.27], but yet not defined as valid values for VirtualCp.additionalService Data.portData.protocol neither in ETSI GS NFV-IFA 011 [i.3] and in ETSI GS NFV-SOL 001 [i.7].				

5.4.2.1.15 ServicePortData

Table 5.4.2.1.15-1 illustrates a comparison of the attributes of the ServicePortData information element as specified in clause 7.1.18.4 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.15-1: Comparison for the ServicePortData information element

Attributes of ServicePortData information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api.core.v1. [i.10]		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
name	1	Service.spec.ports.name		Correspondence
protocol	1	Service.spec.ports.protocol		Correspondence
port	1	Service.spec.ports.port		Correspondence
portConfigurable	1	N/A		See note
NOTE: This property is not destined to be provided to the CISM, but to be held by the VNFM.				

5.4.2.1.16 VnfVirtualLinkDesc

No correspondence can be found since the Kubernetes CISM does not manage such resources. However, extensions to the Kubernetes® API can be used as CCM path for instantiation of a VnfVirtualLink as a cluster resource (e.g. https://access.redhat.com/documentation/fr-fr/openshift_container_platform/4.3/html/networking/cluster-network-operator).

5.4.2.1.17 OsContainerDesc

Table 5.4.2.1.17-1 illustrates a comparison of the attributes the OsContainerDesc information element as specified in clause 7.1.6.17 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.17-1: Comparison for the OsContainerDesc information element

Attributes of OsContainerDesc information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api.core.v1. [i.10]		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
osContainerDescId	1	(Deployment / StatefulSet).spec.template.spec.containers[$index]	1	Correspondence.
name	1	(Deployment / StatefulSet).spec.template.spec.containers[$index].name	1	Correspondence.
description	1	-	-	No correspondence.
requestedCpuResources	0..1	(Deployment / StatefulSet).spec.template.spec.containers[$index].resource.requests.cpu	0..1	Correspondence.
requestedMemory Resources	0..1	(Deployment / StatefulSet).spec.template.spec.containers[$index].resource.requests.memory	0..1	Correspondence. See note 1.
requestedEphemeral StorageResources	0..1	(Deployment / StatefulSet).spec.template.spec.containers[$index].resource.requests.ephemeral_storage	0..1	Correspondence. See note 1.
extendedResource Requests	0..N	(Deployment / StatefulSet).spec.template.spec.containers[$index].resource.requests.$extended_resource_key	0..N	Correspondence. See note 1.
cpuResourceLimit	0..1	(Deployment / StatefulSet).spec.template.spec.containers[$index].resource.limits.cpu		Correspondence. See note 1.
memoryResourceLimit	0..1	(Deployment / StatefulSet).spec.template.spec.containers[$index].resource.limits.memory		Correspondence. See note 1.
ephemeralStorageResource Limit	0..1	(Deployment / StatefulSet).spec.template.spec.containers[$index].limits.ephemeral_storage		Correspondence. See note 1.
hugePageResources	0..1	(Deployment / StatefulSet).spec.template.spec.containers[$index].limits.[].hugepages-$size	0..1	Correspondence.
swImageDesc	1	-	-	See clause 5.4.2.1.23. See notes 2 and 3
	0..1	Everything in: (Deployment/StatefulSet).spec.template.spec.containers[$index] Except : resources, name, image		Suggested correspondence. See note 4.
monitoringParameters	0..N	FFS	FFS	FFS

NOTE 1: 'containers' is a list whose entries are of type 'Container' (K8S). 'containers[$index]' denotes the entry in 'containers', at index $index.

NOTE 2: However, the 'name' attribute of NFV SwImage IE matches the K8S field: (Deployment / StatefulSet).spec.template.spec.containers[$index].image

NOTE 3: Except for 'name and 'resources':

- All fields in K8S 'Container' describe "software configuration data" as defined in ETSI GR NFV-EVE 022 [i.28], clause 3.1.
- Most of them are "virtualization-independent" data as defined in the same clause.
- For most of them, the K8S API description says that they cannot be updated.
- Except maybe for imagePullPolicy, they will all be populated by the VNF vendor, not by the operator.

Using this correspondence to bootData would greatly facilitate any NFV-to-K8S mapping in a VNFM.

NOTE 4: For an assessment of the correspondence with Kubernetes objects, refer to the clauses applicable to the IEs where this attribute is referenced.

5.4.2.1.18 VirtualStorageDesc

Table 5.4.2.1.18-1 illustrates a comparison of the attributes the VirtualStorageDesc information element as specified in clause 7.1.9.4.2 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.18-1: Comparison for the VirtualStorageDesc information element

Attributes of VirtualStorageDesc information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api.core.v1.		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
id	1	PersistentVolumeClaim. metadata.name / StatefulSet.spec. volumeClaimTemplates[]. metadata.name	1 1	Correspondence, when used by Deployment. Correspondence, when used by StatefulSet. See note 1.

typeOfStorage	1	PersistentVolumeClaim.spec.
		volumeMode : {
		if typeOfStorage = 'BLOCK': 'Block'
		else: 'File'
		}
				Full Correspondence if OBJECT is never used.
				No correspondence if typeOfStorage: OBJECT.

blockStorageData	0..1		0..1	Partial Correspondence. See clause 5.4.2.1.19.
objectStorageData	0..1	-	-	No correspondence.
fileStorageData	0..1		0..1	Partial correspondence. See clause 5.4.2.1.20.
nfviMaintenanceInfo	0..1			See note 2.
perVnfcInstance	0..1		1	Correspondence (with the kind of workload manifest). See note 1.

NOTE 1: The VirtualStorageDesc concept globally matches the Kubernetes PersistentVolumeClaim concept:

- When these concepts describe a resource used in a stateless way, perVnfcInstance equals 'false'.
- In this case, the PersistentVolumeClaim manifest is pointed at from a Deployment manifest.
- When these concepts describe a resource used in a stateful way, perVnfcInstance equals 'true'.
- In this case, the PersistentVolumeClaim manifest is included in a StatefulSet manifest as an entry of the 'volumeClaimTemplate' field.

NOTE 2: The analysis is not covered by the current version of the present document.

5.4.2.1.19 BlockStorageData

Table 5.4.2.1.19-1 illustrates a comparison of the attributes the BlockStorageData information element as specified in clause 7.1.9.4.3 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.19-1: Comparison for the BlockStorageData information element

Attributes of BlockStorageData information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api.{core, apps}.v1.		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
sizeOfStorage	1	PersistentVolumeClaim.spec . resources.limits.storage	0..1	Correspondence.
vduStorageRequirements	0..N			See note 2.
rdmaEnabled	0..1	Deployment.spec.template.spec. Containers.resources.limits. $hardware-vendor/rdma:1		This assumes an instance of the rdma device plugin is installed on part of the cluster CIS.
swImageDesc	0..1	N/A	N/A	See note 1.
NOTE 1: As per ETSI GS NFV-IFA 011 [i.3], table 7.1.9.4.3.2-1, this attribute is not present in a VirtualStorageDesc used in a VDU realized by one or a set of OS containers. NOTE 2: The analysis is not covered by the current version of the present document.				

5.4.2.1.20 FileStorageData

Table 5.4.2.1.20-1 illustrates a comparison of the attributes the FileStorageData information element as specified in clause 7.1.9.4.5 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.20-1: Comparison for the FileStorageData information element

Attributes of FileStorageData information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api.core.v1.		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
sizeOfStorage	1	PersistentVolumeClaim.spec. resources.limits.storage	0..1	Correspondence.
fileSystemProtocol	1			See note.
intVirtualLinkDesc	1			See note.
NOTE: The analysis is not covered by the current version of the present document.				

5.4.2.1.21 VnfDf

A VNF flavour is a validated configuration of the VNF as a composition. Thus, a VnfDf describes VNF-level information, which is not intended to be managed by a CISM.

Table 5.4.2.1.21-1 illustrates a comparison of the attributes the VnfDf information element as specified in clause 7.1.8 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.21-1: Comparison for the VnfDf information element

Attributes of VnfDf information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api*.v1.		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
flavourId	1	-	-	No correspondence.
description	1	-	-	No correspondence.
vduProfile	1..N	-	-	No correspondence.
virtualLinkProfile	0..N	-	-	No correspondence.
vipCpProfile	0..N	-	-	No correspondence.
mciopProfile	0..N	-	-	No correspondence.
instantiationLevel	1..N	-	-	No correspondence.
defaultInstantiationLevelId	0..N	-	-	No correspondence.
supportedOperation	0..N	-	-	No correspondence.
vnfLcmOperationConfiguration	1	-	-	No correspondence.
affinityOrAntiAffinityGroup	0..N	-	-	No correspondence.
vnfIndicator	0..N	-	-	No correspondence.

Attributes of VnfDf information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api*.v1.		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
supportedVnfInterface	0..N	-	-	No correspondence.
supportedCoordinationActions	0..N	-	-	No correspondence.
monitoringParameter	0..N	-	-	No correspondence.
scalingAspect	0..N	-	-	No correspondence.
initialDelta	0..N	-	-	No correspondence.
dependencies	0..N	-	-	No correspondence.

5.4.2.1.22 McioConstraintParams as populated using GrantInfo

Table 5.4.2.1.22-1 shows how to map the key/value pairs computed by the VNFM after a Grant Response from the NFVO.

Table 5.4.2.1.22-1 illustrates a comparison of the attributes the McioConstraintsData information element as specified in clause 7.1.6.2 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.22-1: Comparison for the Keys in a Vdu mcioConstraintParams information element

Keys in a Vdu mcioConstraintParams information element in ETSI GS NFV-IFA 011 [i.3]	Attributes of Kubernetes information element In io.k8s.api.apps.v1.	Comments
Keys	**Attribute**	
affinityNfviPop	Simple correspondence of affinityNfviPop: "$key: $value" (Deployment / StatefulSet).spec.template.spec. nodeSelector: "$key: $value" - Complex correspondence: (Deployment / StatefulSet).spec.template.spec. affinity.nodeAffinity. requiredDuringSchedulingIgnoredDuringExecution. nodeSelectorTerms[].matchExpressions: - key: $key operator: In values: - $value	Possible correspondences. The key/value pair "affinityNfviPop: $mcioConstraintValue" is provided by the NFVO in the mcioConstraints. The "$mcioConstraintValue" is in turn composed of two sub-strings corresponding to "$key: $value" and identifies a set of CIS nodes. See notes 1 and 3.
affinityZone	Simple correspondence of affinityZone: "$key: $value": (Deployment / StatefulSet).spec.template.spec. nodeSelector: "$key: $value" - Complex correspondence: (Deployment / StatefulSet).spec.template.spec. affinity.nodeAffinity. requiredDuringSchedulingIgnoredDuringExecution. nodeSelectorTerms[].matchExpressions: - key: $key operator: In values: - $value	Possible correspondences. The key/value pair "affinityZone: $mcioConstraintValue" is provided by the NFVO in the mcioConstraints. The "$mcioConstraintValue" is in turn composed of two sub-strings corresponding to "$key: $value" and identifies a set of CIS nodes. See notes 1 and 3.

Keys in a Vdu mcioConstraintParams information element in ETSI GS NFV-IFA 011 [i.3]	Attributes of Kubernetes information element In io.k8s.api.apps.v1.	Comments
Keys	Attribute	
affinityZoneGroup	Simple correspondence of affinityZoneGroup: $value: (Deployment / StatefulSet).spec.template.spec. nodeSelector: "$key: $value" - Complex correspondence: (Deployment / StatefulSet).spec.template.spec. affinity.nodeAffinity. requiredDuringSchedulingIgnoredDuringExecution. nodeSelectorTerms[].matchExpressions: - key: $key operator: In values: - $value	Possible correspondences. The key/value pair "affinityZoneGroup: $mcioConstraintValue" is provided by the NFVO in the mcioConstraints. The "$mcioConstraintValue" is in turn composed of two sub-strings corresponding to "$key: $value" and identifies a set of CIS nodes. See notes 1 and 3.
affinityNfviNode	Simple correspondence of affinityNfviNode: $value: (Deployment / StatefulSet).spec.template.spec. nodeSelector: "$key: $value" - Complex correspondence: (Deployment / StatefulSet).spec.template.spec. affinity.nodeAffinity. requiredDuringSchedulingIgnoredDuringExecution. nodeSelectorTerms[].matchExpressions: - key: $key operator: In values: - $value	Possible correspondences. The key/value pair "affinityNfviNode: $mcioConstraintValue" is provided by the NFVO in the mcioConstraints. The "$mcioConstraintValue" is in turn composed of two sub-strings corresponding to "$key: $value" and identifies a set of CIS nodes. See notes 1 and 3.
affinityCisNode	Simple correspondence of affinityCisNode: $value: (Deployment / StatefulSet).spec.template.spec. nodeSelector: $key: $value - Complex correspondence: (Deployment / StatefulSet).spec.template.spec. affinity.nodeAffinity. requiredDuringSchedulingIgnoredDuringExecution. nodeSelectorTerms[].matchExpressions: - key: $key operator: In values: - $value	Possible correspondences. The key/value pair "affinityCisNode: $mcioConstraintValue" is provided by the NFVO in the mcioConstraints. The "$mcioConstraintValue" is in turn composed of two sub-strings corresponding to "$key: $value" and identifies a set of CIS nodes. See notes 1 and 3.
antiAffinityNfviPop	(Deployment / StatefulSet).spec.template.spec. affinity.nodeAntiAffinity. requiredDuringSchedulingIgnoredDuringExecution. nodeSelectorTerms[].matchExpressions: - key: $key operator: In values: - $value	Possible correspondences. The key/value pair "antiAffinityNfviPop: $mcioConstraintValue" is provided by the NFVO in the mcioConstraints. The "$mcioConstraintValue" is in turn composed of two sub-strings corresponding to "$key: $value" and identifies a set of CIS nodes. See notes 1 and 3.

Keys in a Vdu mcioConstraintParams information element in ETSI GS NFV-IFA 011 [i.3]	Attributes of Kubernetes information element In io.k8s.api.apps.v1.	Comments
Keys	Attribute	
antiAffinityZone	(Deployment / StatefulSet).spec.template.spec. affinity.nodeAntiAffinity. requiredDuringSchedulingIgnoredDuringExecution. nodeSelectorTerms[].matchExpressions: - key: $key operator: In values: - $value	Possible correspondences. The key/value pair "antiAffinityZone: $mcioConstraintValue" is provided by the NFVO in the mcioConstraints. The "$mcioConstraintValue" is in turn composed of two sub-strings corresponding to "$key: $value" and identifies a set of CIS nodes. See notes 1 and 3.
antiAffinityZoneGroup	(Deployment / StatefulSet).spec.template.spec. affinity.nodeAntiAffinity. requiredDuringSchedulingIgnoredDuringExecution. nodeSelectorTerms[].matchExpressions: - key: $key operator: In values: - $value	Possible correspondences. The key/value pair "antiAffinityZoneGroup: $mcioConstraintValue" is provided by the NFVO in the mcioConstraints. The "$mcioConstraintValue" is in turn composed of two sub-strings corresponding to "$key: $value" and identifies a set of CIS nodes. See notes 1 and 3.
antiAffinityNfviNode	(Deployment / StatefulSet).spec.template.spec. affinity.nodeAntiAffinity. requiredDuringSchedulingIgnoredDuringExecution. nodeSelectorTerms[].matchExpressions: - key: $key operator: In values: - $value	Possible correspondences. The key/value pair "antiAffinityNfviNode: $mcioConstraintValue" is provided by the NFVO in the mcioConstraints. The "$mcioConstraintValue" is in turn composed of two sub-strings corresponding to "$key: $value" and identifies a set of CIS nodes. See notes 1 and 3.
antiAffinityCisNode	(Deployment / StatefulSet).spec.template.spec. affinity.nodeAntiAffinity. requiredDuringSchedulingIgnoredDuringExecution. nodeSelectorTerms[].matchExpressions: - key: $key operator: In values: - $value	Possible correspondences. The key/value pair "antiAffinityCisNode: $mcioConstraintValue" is provided by the NFVO in the mcioConstraints. The "$mcioConstraintValue" is in turn composed of two sub-strings corresponding to "$key: $value" and identifies a set of CIS nodes. See notes 1 and 3.

Keys in a Vdu mcioConstraintParams information element in ETSI GS NFV-IFA 011 [i.3]	**Attributes of Kubernetes information element In io.k8s.api.apps.v1.**	**Comments**
Keys	**Attribute**	
localAffinityNfviPop	(Deployment / StatefulSet).spec.template.spec. affinity.podAffinity. requiredDuringSchedulingIgnoredDuringExecution[]. labelSelector.matchExpressions: - key: $key operator: In values: - $value topologyKey: etsi.org/nfviPop	Possible correspondences. This example assumes that "etsi.org/nfviPop" is provided as value in the key/value pair entry with key "localAffinityNfviPop" of the "mcioConstraints" in the granting response provided by the NFVO. See notes 1, 4 and 5.
localAffinityZone	(Deployment / StatefulSet).spec.template.spec. affinity.podAffinity. requiredDuringSchedulingIgnoredDuringExecution[]. labelSelector.matchExpressions: - key: $key operator: In values: - $value topologyKey: etsi.org/zone	Possible correspondences. This example assumes that "etsi.org/zone" is provided as value in the key/value pair entry with key "localAffinityZone" of the "mcioConstraints" in the granting response provided by the NFVO. See notes 1, 4 and 5.
localAffinityZoneGroup	(Deployment / StatefulSet).spec.template.spec. affinity.podAffinity. requiredDuringSchedulingIgnoredDuringExecution[]. labelSelector.matchExpressions: - key: $key operator: In values: - $value topologyKey: etsi.org/zoneGroup	Possible correspondences. This example assumes that "etsi.org/zoneGroup" is provided as value in the key/value pair entry with key "localAffinityZoneGroup" of the "mcioConstraints" in the granting response provided by the NFVO. See notes 1, 4 and 5.
localAffinityNfviNode	(Deployment / StatefulSet).spec.template.spec. affinity.podAffinity. requiredDuringSchedulingIgnoredDuringExecution[]. labelSelector.matchExpressions: - key: $key operator: In values: - $value topologyKey: etsi.org/nfviNode	Possible correspondences. This example assumes that "etsi.org/nfviNode" is provided as value in the key/value pair entry with key "localAffinityNfviNode" of the "mcioConstraints" in the granting response provided by the NFVO. See notes 1, 4 and 5.
localAffinityCisNode	(Deployment / StatefulSet).spec.template.spec. affinity.podAffinity. requiredDuringSchedulingIgnoredDuringExecution[]. labelSelector.matchExpressions: - key: $key operator: In values: - $value topologyKey: etsi.org/cisNode	Possible correspondences. This example assumes that "etsi.org/cisNode" is provided as value in the key/value pair entry with key "localAffinityCisNode" of the "mcioConstraints" in the granting response provided by the NFVO. See notes 1, 4 and 5.

Keys in a Vdu mcioConstraintParams information element in ETSI GS NFV-IFA 011 [i.3]	Attributes of Kubernetes information element In io.k8s.api.apps.v1.	Comments
Keys	**Attribute**	
localAntiAffinityNfviPop	(Deployment / StatefulSet).spec.template.spec. affinity.podAntiAffinity. requiredDuringSchedulingIgnoredDuringExecution[]. labelSelector.matchExpressions: - key: $key operator: In values: - $value topologyKey: etsi.org/nfviPop	Possible correspondences. This example assumes that "etsi.org/nfviPop" is provided as value in the key/value pair entry with key "localAntiAffinityNfviPop" of the "mcioConstraints" in the granting response provided by the NFVO. See notes 1, 4 and 5.
localAntiAffinityZone	(Deployment / StatefulSet).spec.template.spec. affinity.podAntiAffinity. requiredDuringSchedulingIgnoredDuringExecution[]. labelSelector.matchExpressions: - key: $key operator: In values: - $value topologyKey: etsi.org/zone	Possible correspondences. This example assumes that "etsi.org/zone" is provided as value in the key/value pair entry with key "localAntiAffinityZone" of the "mcioConstraints" in the granting response provided by the NFVO. See notes 1, 4 and 5.
localAntiAffinityZoneGroup	(Deployment / StatefulSet).spec.template.spec. affinity.podAntiAffinity. requiredDuringSchedulingIgnoredDuringExecution[]. labelSelector.matchExpressions: - key: $key operator: In values: - $value topologyKey: etsi.org/zoneGroup	Possible correspondences. This example assumes that "etsi.org/zoneGroup" is provided as value in the key/value pair entry with key "localAntiAffinityZoneGroup" of the "mcioConstraints" in the granting response provided by the NFVO. See notes 1, 4 and 5.
localAntiAffinityNfviNode	(Deployment / StatefulSet).spec.template.spec. affinity.podAntiAffinity. requiredDuringSchedulingIgnoredDuringExecution[]. labelSelector.matchExpressions: - key: $key operator: In values: - $value topologyKey: etsi.org/nfviNode	Possible correspondences. This example assumes that "etsi.org/nfviNode" is provided as value in the key/value pair entry with key "localAntiAffinityNfviNode" of the "mcioConstraints" in the granting response provided by the NFVO. See notes 1, 4 and 5.
localAntiAffinityCisNode	(Deployment / StatefulSet).spec.template.spec. affinity.podAntiAffinity. requiredDuringSchedulingIgnoredDuringExecution[]. labelSelector.matchExpressions: - key: $key operator: In values: - $value topologyKey: etsi.org/cisNode	Possible correspondences. This example assumes that "etsi.org/cisNode" is provided as value in the key/value pair entry with key "localAntiAffinityCisNode" of the "mcioConstraints" in the granting response provided by the NFVO. See notes 1, 4 and 5.

Keys in a Vdu mcioConstraintParams information element in ETSI GS NFV-IFA 011 [i.3]	**Attributes of Kubernetes information element In io.k8s.api.apps.v1.**	**Comments**
Keys	**Attribute**	
nodeAdditionalCapability Ssd	(Deployment / StatefulSet).spec.template.spec. nodeSelector: etsi.org/ nodeAdditionalCapability: ssd	Possible correspondence.
nodeAdditionalCapability Dpdk	'nodeAdditionalCapabilityDpdk' can be mapped onto: Deployment.spec.template.spec. containers.resources.limits: $hardware-vendor/$resource: 1	Correspondence if the cluster is running a device plugin that advertises the resource, e.g. as '$hardware-vendor/$resource'. See note 2.
nodeAdditionalCapability Sriov	'nodeAdditionalCapabilitySriov' can be mapped onto: Deployment.spec.template.spec. containers.resources.limits: $hardware-vendor/$resource: 1	Correspondence if the cluster is running a device plugin that advertises the resource, e.g. as '$hardware-vendor/$resource'. See note 2.
nodeAdditionalCapability Gpu	'nodeAdditionalCapabilityGpu' can be mapped onto: Deployment.spec.template.spec. containers.resources.limits: $hardware-vendor/$resource: 1	Correspondence if the cluster is running a device plugin that advertises the resource, e.g. as '$hardware-vendor/$resource'. See note 2.
nodeAdditionalCapability Fpga	'nodeAdditionalCapabilityFpga' can be mapped onto: Deployment.spec.template.spec. containers.resources.limits: $hardware-vendor/$resource: 1	Correspondence if the cluster is running a device plugin that advertises the resource, e.g. as '$hardware-vendor/$resource'. See note 2.
nodeAdditionalCapability CpuPin	'nodeAdditionalCapabilityCpuPin' can be mapped onto: Deployment.spec.template.spec. containers.resources.limits: $hardware-vendor/$resource: 1	Correspondence if the cluster is running a device plugin that advertises the resource, e.g. as '$hardware-vendor/$resource'. See note 2.
nodeCapabilityLogicalNuma		See note 6.
nodePool		See note 6.

NOTE 1: The term 'requiredDuringSchedulingIgnoredDuringExecution' can be substituted with 'preferredDuringSchedulingIgnoredDuringExecution', in which case if a matching node is not available, the scheduler still schedules the Pod. However, no NFV IE currently allow to select one of the 2 options on a per VNF instance basis, or on a per VNF instance's mcio parameter basis.

NOTE 2: The correspondence assumes that at VNF instantiation, in the grant response, the'GrantInfo.mcioConstraints'Attribute, contains the key 'nodeAdditionalCapability*' with the value $hardware-vendor/$resource.

NOTE 3: The concerned CIS nodes have been marked accordingly beforehand through CCM.

NOTE 4: The short names of scopes for affinity or anti-affinity defined by NFV are here 'frozen' as NFV-defined (e.g. in an NFV registry) Kubernetes label keys, e.g. etsi.org/nfviPop, based on the Kubernetes model (which one proposes only 2 scopes: topology.kubernetes.io/region and topology.kubernetes.io/zone).

NOTE 5: A key-value pair ($key: $value) identifies all the VNFC instances based on the same Vdu. Kubernetes pods are schedulable in the scope, e.g. etsi.org/nfviPop, or not, depending on whether other pods with the same label ($key: $value) are also scheduled in the scope. This assumes that the VNFM knows how to generate a K8s label for all the replicas of a same Kubernetes Deployment, i.e. for all the VNFC instances based on the same Vdu (for a given VNF instance).

NOTE 6: The analysis is not covered by the current version of the present document.

5.4.2.1.23 SwImageDesc

Table 5.4.2.1.23-1 illustrates a comparison of the attributes the SwImageDesc information element as specified in clause 7.1.6.5 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.23-1: Comparison for the SwImageDesc information element

Attributes of SwImageDesc information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api*.v1.		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
id	1	(Deployment / StatefulSet).spec. template.spec.containers.image where the value of 'image' attribute includes a registry hostname.		Possible correspondence. e.g. my_telco.registry. example:666/image_name)
Name	1	Part of (Deployment / StatefulSet).spec. template.spec.containers.image after value of cirConnectionInfo in Grant and before value of SwImageDesc.version		Correspondence as defined in ETSI GS NFV-SOL 018 [i.27] Table 6.2.2.1-5 where the image value is a compound value of the following: - cirConnectionInfo in Grant of ETSI GS NFV-SOL 003 [i.8] - name in SwImage of ETSI GS NFV-SOL 001 [i.7] - version in SwImage of ETSI GS NFV-SOL 001 [i.7] e.g. my_telco.registry. example:666/image01:22.04
version	1	Part of (Deployment / StatefulSet).spec. template.spec.containers.image after value of SwImageDesc.name		Correspondence as defined in ETSI GS NFV-SOL 018 [i.27] Table 6.2.2.1-5 where the image value is a compound value of the following: - cirConnectionInfo in Grant of ETSI GS NFV-SOL 003 [i.8] - name in SwImage of - version in SwImage of ETSI GS NFV-SOL 001 [i.7] e.g. my_telco.registry. example:666/image01:22.04
checksum	0..1	-		No correspondence.
containerFormat	1	-		No correspondence.
diskFormat	0..1	-		No correspondence.
minDisk	0..1	-		No correspondence.
minRam	0..1	-		No correspondence.
size	0..1	-		No correspondence.
swImage	1	-		No correspondence.
operatingSystem	0..1	-		No correspondence.
supportedVirtualisationEnvironments	0..N	-		No correspondence.

5.4.2.1.24 SecurityGroupRule

Table 5.4.2.1.24-1 illustrates a comparison of the attributes the SecurityGroupRule information element as specified in clause 7.1.6.9 in ETSI GS NFV-IFA 011 [i.3] with properties in the Kubernetes resource object manifests as specified in [i.10].

Table 5.4.2.1.24-1: Comparison for the SecurityGroupRule information element

Attributes of SecurityGroupRule information element in ETSI GS NFV-IFA 011 [i.3]		Attributes of Kubernetes information element In io.k8s.api.networking.v1.		Comments
Attribute	**Cardinality**	**Attribute**	**Cardinality**	
securityGroupRuleId	1	NetworkPolicy.metadata.name and $index in NetworkPolicy.spec.(egress / ingress)[$index]	1	Complex correspondence. See note.
description	0..1	-		No formal correspondence. But could be stored, e.g. in NetworkPolicy.metadata.as 'annotations. description$index.' (i.e. concatenation of word 'description' and value of $index).
direction	0..1	For an egress rule: NetworkPolicy.spec.policyTypes: egress		Rules will be grouped for egress or ingress or ingress and egress.
etherType	0..1	-		No correspondence. However, if the cidr is known, 'NetworkPolicy. spec.egress[$index].to[ ].ipBlock could be used as a subset of the etherType.
protocol	0..1	NetworkPolicy.spec.egress[$index].ports[ ].protocol		Correspondence.
portRangeMin	0..1	NetworkPolicy.spec.egress[$index].ports[ ].port		Correspondence.
portRangeMax	0..1	NetworkPolicy.spec.egress[$index].ports[ ].endPort		Correspondence.
NOTE: A K8S NetworkPolicy is a set of egress or ingress or bidirectional rules.				

6 Recommendations

6.1 Overview

The present clause documents recommendations about potential enhancements, changes, or clarifications to existing ETSI NFV specifications. The recommendations are derived based on the analysis between the related open source projects and the ETSI NFV specifications, and the potential solutions to resolve the identified gaps, as described in clause 5.

The recommendations are categorized and elaborated as follows:

- recommendations related to VNFD (refer to clause 6.2);
- recommendations related to VNF LCM interface (refer to clause 6.3);
- recommendations related to VNF PM interface (refer to clause 6.4);
- recommendations related to VNF FM interface (refer to clause 6.5);
- recommendations related to Grant interface (refer to clause 6.6).

6.2 Recommendations related to VNFD

The present clause provides recommendations related to VNFD.

Table 6.2-1 provides the recommendations related to VNFD.

Table 6.2-1: Recommendations related to VNFD

Identifier	Recommendation description	Comments
vnfd.002	It is recommended that a requirement be specified to state that for containerized VNF in which cases virtualComputeDesc, osContainerDesc, virtualStorageDesc and swImageDesc are not mandatory to be present.	See solution SOL 1-1b in clause 5.2.2.6.1.2 and Gap#1 of ASD in clause 5.2.2.6.1.

6.3 Recommendations related to VNF LCM interface

The present clause provides recommendations related to VNF LCM interface.

Table 6.3-1 provides the recommendations related to VNF LCM interface.

Table 6.3-1: Recommendations related to VNF LCM interface

Identifier	Recommendation description	Comments
vnfLcm.001	It is recommended that a requirement be specified to be able to support the externalIP value assignment through VNF LCM in ETSI GS NFV-SOL 003 [i.8].	See solution SOL 3-1 in clause 5.2.2.6.3.2 and Gap#3 of ASD in clause 5.2.2.6.3.
vnfLcm.002	It is recommended that a requirement be specified for Heal VNF operation of the VNF LCM interface produced by the VNFM to support determining which network and/or storage resource to heal.	See Gap#1 in clause 5.3.2.6.1. Heal VNF operation already supports indicating which compute resource to heal through the "vnfcInstanceId" attribute of ETSI GS NFV-SOL 002 [i.31].
vnfLcm.003	It is recommended that a requirement be specified for Change Current VNF package operation of the VNF LCM interface produced by the VNFM to support determining upgrade type to realize, i.e. Blue-Green upgrade, Rolling upgrade, based on mapping information between source of VNF/VNFC instance to destination of VNF/VNFC instance from enhanced VnfPackageChangeInfo in VNFD.	See Gap#1 in clause 5.3.2.6.1. "componentMapping" attribute of VnfPackageChangeInfo in VNFD can describes mapping source VNFD and destination VNFD in upgrade process.
vnfLcm.004	It is recommended that a requirement be specified for the VNF LCM interface produced by the VNFM to NFVO to support managing VNFC (component) and VL (network), if the operation can change both kind of resources.	See Gap#1 in clause 5.3.2.6.1. VNF LCM request from EM can support VNFC, and VNF LCM request from NFVO can support VL.
vnfLcm.005	It is recommended that a requirement be specified for the VNF LCM interface produced by the VNFM to support providing metadata in instantiatedVnfInfo to store information related to VNF LCM operation after successful instantiation.	See Gap#1 in clause 5.3.2.6.1.

6.4 Recommendations related to VNF PM interface

The present clause provides recommendations related to VNF PM interface.

Table 6.4-1 provides the recommendations related to VNF PM interface.

Table 6.4-1: Recommendations related to VNF PM interface

Identifier	Recommendation description	Comments
vnfPm.001	It is recommended that a requirement be specified for create PM job operation of the VNF PM interface produced by the VNFM to support indicating access information to monitoring component in CISM/VIM/PaaS.	See Gap#2 in clause 5.3.2.6.2. Access information to monitoring component, such as a kind of VimConnectionInfo, can be configured into the VNFM using NFV-MANO management interfaces [i.24].

6.5 Recommendations related to VNF FM interface

The present clause provides recommendations related to VNF FM interface.

Table 6.5-1 provides the recommendations related to VNF FM interface.

Table 6.5-1: Recommendations related to VNF FM interface

Identifier	Recommendation description	Comments
vnfFm.001	It is recommended that a requirement be specified for the VNF FM interface and VNF Indicator produced by the VNFM to NFVO to support managing VNFC (component) and VL (network), if the alarm can correlate both kind of resources.	See Gap#1 in clause 5.3.2.6.1. VNF FM request from EM can support VNFC, and VNF FM request from NFVO can support VL.
vnfFm.002	It is recommended that a requirement be specified for subscribe operation of the VNF FM interface produced by the VNFM to support indicating access information to monitoring component in CISM/VIM/PaaS.	Same recommendation is useful as vnfPm.001 of recommendation related to VNF PM in clause 6.4. Access information to monitoring component, such as a kind of VimConnectionInfo, can be configured into the VNFM using NFV-MANO management interfaces [i.24].

6.6 Recommendations related to Grant interface

The present clause provides recommendations related to Grant interface.

Table 6.6-1 provides the recommendations related to Grant interface.

Table 6.6-1: Recommendations related to Grant interface

Identifier	Recommendation description	Comments
grant.001	It is recommended that a requirement be specified to be able to support containing resource requirement information in the Grant VNF Lifecycle operation in ETSI GS NFV-IFA 007 [i.4].	See solution SOL 1-1b in clause 5.2.2.6.1.2 and Gap#1 of ASD in clause 5.2.2.6.1. To minimize impacts to the NFVO, the modeling of resource requirements on the granting interface are recommended to follow the same modeling as specified already in the referenced VNFD specification.

Annex A: Change History

Date	Version	Information about changes
0.0.1	2022.9	Initial version with the approved skeleton. NFVIFA(22)000639r1_skeleton_of_new_report_IFA051
0.0.2	2022.9	Implemented: NFVIFA(22)000645r1_IFA051_adding_scope, NFVIFA(22)000661r1_IFA051_adding_introduction_of_current_VNF_management NFVIFA(22)000663r2_IFA051_ASD_model_introduction NFVIFA(22)000664r3_IFA051_comparison_between_VNFD_and_ASD_information_model NFVIFA(22)000665r2_IFA051_comparison_between_vnfExtCpd_and_asdExtCpd_information_model NFVIFA(22)000666r2_IFA051_comparison_between_mciopProfile_and_deploymentItems_information_model
0.0.3	2022.11	Implemented: NFVIFA(22)000799r1_IFA051_comparison_of_VNF_instantiate_operation, NFVIFA(22)000836r1_IFA051_comparison_of_OpenStack_Tacker_VNFD
0.0.4	2022.12	Implemented: NFVIFA(22)000798r2_IFA051_comparison_of_identifier_creation_operation, NFVIFA(22)000826r2_IFA051_-_Gap_Analysis_with_Kubernetes_-_VNFD, NFVIFA(22)000857r1_IFA051_comparison_of_OpenStack_Tacker_VNF_LCM_API_for_instantiate_VNF, NFVIFA(22)000922r1_IFA051_comparison_of_OpenStack_Tacker_VNF_LCM_API_for_terminate_VNF NFVIFA(22)000923r1_IFA051_comparison_of_OpenStack_Tacker_VNF_LCM_API_for_Query_
0.0.5	2023.01	Implemented: NFVIFA(22)000662r3_IFA051_adding_introduction_of_ONAP_ASD, NFVIFA(22)000885r3_IFA051_update_comparison_table_5_2_2_1_4-1, NFVIFA(22)000886r1_IFA051_adding_gaps_based_on_comparison_with_ONAP_ASD, NFVIFA(22)000953_IFA051_comparison_of_OpenStack_Tacker_VNF_LCM_API_for_Scale_VNF, NFVIFA(22)000954_IFA051_comparison_of_OpenStack_Tacker_VNF_LCM_API_for_Heal_VNF, NFVIFA(22)000955_IFA051_comparison_of_OpenStack_Tacker_VNF_LCM_API_for_Change_Current_External_VNF_Connectivity, NFVIFA(22)000958_IFA051_comparison_of_OpenStack_Tacker_VNF_LCM_API_for_Change_Current_VNF_Package, NFVIFA(22)000980r1_IFA051_comparison_of_OpenStack_Tacker_VNF_LCM_API_for_Modify_VNF NFVIFA(22)000797r3_IFA051_adding_introduction_of_VNF_LCM_related_work_in_ONAP_ASD
0.0.6	2023.02	Implemented: NFVIFA(22)000977r4_IFA051_-_Gap_Analysis_with_Kubernetes_-_VNFD_-_Further_improvements NFVIFA(23)000045r2_IFA051_comparison_of_OpenStack_Tacker_VNF_PM_API, NFVIFA(23)000059_IFA051_comparison_of_OpenStack_Tacker_VNF_FM_API, NFVIFA(23)000061_IFA051_comparison_of_Grant_of_OpenStack_Tacker, NFVIFA(23)000064_IFA051_introduction_and_gaps_of_OpenStack_Tacker, NFVIFA(23)000065r1_IFA051_adding_FM_related_comparison_with_ONAP_ASD, NFVIFA(23)000066r1_IFA051_adding_PM_related_comparison_with_ONAP_ASD, NFVIFA(23)000067r2_IFA051_adding_new_gap_related_to_input_parameter_mapping_for_ONAP_ASD,
0.0.7	2023.03	Implemented: NFVIFA(23)000169r2_IFA051_adding_potential_solutions_for_ASD_gap#1, NFVIFA(23)000170r1_IFA051_update_ASD_comparison
0.0.8	2023.04	Implemented: NFVIFA(23)000215r2_adding_solution_evaluation_for_ONAP_ASD_gap_1, NFVIFA(23)000297r1_IFA051_adding_overview, NFVIFA(23)000298r1_IFA051_adding_Grant_operation_comparison_for_ASD, NFVIFA(23)000308_IFA051_adding_comparison_for_terminate_AS_for_for_ASD, NFVIFA(23)000299r1_IFA051_adding_gaps_related_to_VnfLcm_for_ASD

Date	Version	Information about changes
0.0.9	2023.05	Implemented: NFVIFA(23)000316_IFA051_Resolve_EN_of_additionalParams_in_Tacker, NFVIFA(23)000322r1_IFA051_gaps_related_to_VnfLcm_for_ASD
0.0.10	2023.06	Implemented: NFVIFA(23)000372r1_IFA051_potential__solution_for_ASD_Gap_2, NFVIFA(23)000373r3_IFA051_resolving_editor_notes_FM_PM_Grant, NFVIFA(23)000374r3_IFA051_resolving_editor_notes_for_ASD_Gap_1, NFVIFA(23)000422_IFA051_Naming_alignment_across_clause_titles, NFVIFA(23)000425r1_IFA051_potential__solution_for_ASD_Gap_3, NFVIFA(23)000427_IFA051_recommendation_for_VnfLcm_interface, NFVIFA(23)000453r1_IFA051_Improvements_on_potential_solutions_for_Gap_1_in_ONAP_ASD_ analysis, NFVIFA(23)000353_IFA051_Delete_unnecessary_EN_and_fix_bugs_in_Tacker
0.1.0	2023.06	Implemented: NFVIFA(23)000426r2_IFA051_recommendation_for_VNFD, NFVIFA(23)000451r2_IFA051_recommendation_for_Grant_interface, NFVIFA(23)000497r1_IFA051_adding_release_version_number_in_reference, NFVIFA(23)000498_IFA051_resolve_editor_notes_for_ASD, NFVIFA(23)000499r1_IFA051_update_solution_evaluation_for_ASD, NFVIFA(23)000504_IFA051_Resolve_EN_of_gaps_in_Tacker
0.1.1	2023.07	Implemented: NFVIFA(23)000529r1_IFA051_Add_recommendation_of_VNF_LCM_from_gap_of_Tacker, NFVIFA(23)000530r2_IFA051_Add_recommendation_of_VNF_PM_from_gap_of_Tacker, NFVIFA(23)000543_IFA051_reference_corrections, NFVIFA(23)000544r1_IFA051_adding_recommendation_overview, NFVIFA(23)000546r1_IFA051_adding_overview_for_Kubernetes, NFVIFA(23)000548r1_IFA051_editor_notes_resolve_for_Kubernetes.
0.2.0	2023.08	Implemented: NFVIFA(23)000556r1_IFA051_Add_recommendation_of_VNF_FM_from_gap_of_Tacker, NFVIFA(23)000606_IFA051_resolving_editor_notes_for_ASD, NFVIFA(23)000607_IFA051_resolving_editor_notes_for_Kubernetes, NFVIFA(23)000614_IFA051_update_latest_reference_and_figure_of_Tacker, NFVIFA(23)000618_IFA051_remove_unnecessary_annex

History

Document history		
V5.1.1	October 2023	Publication