ETSI GS NFV-IFA 010 V2.3.1 (2017-08)

Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Functional requirements specification

Disclaimer

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

Reference
RGS/NFV-IFA010ed231

Keywords
functional, management, MANO, NFV,
orchestration, requirements, virtualisation

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 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

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

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

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

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.
The content of the PDF version shall not be modified without the written authorization of ETSI.
The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2017.
All rights reserved.

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 protected for the benefit of its Members.
GSM® and the GSM logo are trademarks registered and owned by the GSM Association.

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is 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 IPR Policy, no investigation, 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.

Foreword

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

Modal verbs terminology

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

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

1 Scope

The present document specifies functional requirements for NFV management and orchestration, and general guidelines and requirements for NFV management and orchestration interface design.

The scope of the present document does not cover the functional requirements on interfaces.

2 References

2.1 Normative references

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

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

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

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

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

2.2 Informative references

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

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

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

[i.1] ETSI GS NFV 002: "Network Functions Virtualisation (NFV); Architectural Framework".

[i.2] ETSI GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for main concepts in NFV".

[i.3] ETSI GS NFV 004: "Network Functions Virtualisation (NFV); Virtualisation Requirements".

[i.4] ETSI GS NFV-MAN 001: "Network Functions Virtualisation (NFV); Management and Orchestration".

[i.5] ETSI GS NFV-SWA 001: "Network Functions Virtualisation (NFV); Virtual Network Functions Architecture".

[i.6] ETSI GS NFV-REL 001: "Network Functions Virtualisation (NFV); Resiliency requirements".

[i.7] ETSI GS NFV-INF 001: "Network Functions Virtualisation (NFV); Infrastructure Overview".

[i.8] Recommendation ITU-T Y.3500: "Information technology - Cloud computing - Overview and vocabulary".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ETSI GS NFV 003 [i.2] and the following apply:

NOTE: A term defined in the present document takes precedence over the definition of the same term, if any, in ETSI GS NFV 003 [i.2].

administrative domain: collection of systems and networks operated by a single organization or administrative authority

NOTE: This definition is from ETSI GS NFV-MAN 001 [i.4].

affinity of virtualised network resources: persistent policy that forces Virtual Links (VLs) to share the same physical connectivity

NOTE 1: "Persistent" is used here and in the following definitions to indicate that the affinity remains in effect until a change is requested by the consumer.

NOTE 2: This may be stipulated to ensure the same transmission characteristics (such as delay) for VLs.

anti-affinity of virtualised network resources: persistent policy that forces Virtual Links(VLs) to not share any physical connectivity

NOTE: This may be stipulated to ensure that VLs do not fail at the same time.

area affinity: policy that qualifies an affinity (or anti-affinity) policy with respect to location restrictions

EXAMPLE: The anti-affinity policy of having virtualised compute resources on different compute nodes can be further restricted by mandating to locate the compute nodes on different shelves, racks, bays, sites, geographic areas or similar restriction

NOTE: Anti-affinity can be used to support availability, survivability and performance needs with respect to virtualised resources.

composite network service: network service containing at least one network service

consumable virtualised resource: virtualised resource that can be requested for reservation and/or allocation

NOTE: Virtualised resources comprise compute, network and storage.

EXAMPLE: A volume or object based virtual storage.

infrastructure domain: administrative domain that provides virtualised infrastructure resources such as compute, network, and storage or a composition of those resources via a service abstraction to another administrative domain, and is responsible for the management and orchestration of those resources.

NOTE: This definition is from ETSI GS NFV-MAN 001 [i.4].

infrastructure resource group: logical resource collection grouping virtual resource instances assigned to a tenant along with software images

multi-tenancy: feature where physical,virtual or service resources are allocated in such a way that multiple tenants and their computations and data are isolated from and inaccessible by each another

NOTE: This definition has been specialized from the term "multi-tenancy" as defined in Recommendation ITU-T Y.3500 [i.8].

nested network service: network service that is part of a composite network service

NOTE: A composite network service is a network service containing at least one network service.

network service: composition of network function(s) and/or network service(s), defined by its functional and behavioural specification

NFV-MANO service: one or more capabilities offered via NFV MANO functional blocks invoked using a defined interface

NOTE: This definition has been specialized from the term "cloud service" as defined in Recommendation ITU-T Y.3500 [i.8].

EXAMPLE: The VNFM offers a NFV MANO service for VNF lifecycle management to the NFVO. The NFVO offers a NFV MANO service for Network Service lifecycle management to OSS/BSS functions and uses the NFV-MANO service provided by the VNFM.

NFV-MANO service user: natural person, or entity acting on their behalf, associated with an organization that uses NFV-MANO services

NOTE: This definition has been specialized from the term "cloud service user" as defined in Recommendation ITU-T Y.3500 [i.8].

node affinity for virtualised compute resources: persistent policy that forces virtualised compute resources to be on the same compute node

NOTE 1: "Persistent" is used here and in the following definitions to indicate that the affinity remains in effect until a change is requested by the consumer.

NOTE 2: This is to avoid cases where, for example, virtualised compute resource are initially on the same compute node but then later moved to separate nodes by the provider without any requested policy change from the consumer.

node anti-affinity for virtualised compute resources: persistent policy that forces each virtualised compute resource to be on different compute nodes

node affinity for virtualised storage resources: persistent policy that forces virtualised storage resources to be on the same storage node

node anti-affinity for virtualised storage resources: persistent policy that forces each virtualised storage resources to be on different storage nodes

NS healing: procedure that includes all virtualisation related corrective actions to repair a faulty Network Service (NS) instance including components/functionalities which make up the instance, and have been associated with this fault situation

NOTE 1: In a virtualised environment network service healing focuses only on the virtualised components/functionalities. In case of a NS consisting of virtualised and non-virtualised parts a procedure able to handle both parts is needed. This will be done in connection with components/functionalities that are located outside the virtualised environment.

NOTE 2: "Virtualisation related corrective actions" refers to action(s) toward virtualised resource(s) and associated NS instance.

permitted allowance: constraint in terms of resource capacity, used by NFVO to control resource consumption by VNFMs in relation with VNF Lifecycle Operation Granting

NOTE: Permitted allowances are maintained by the NFVO and might vary in granularity (VNFM, VNF, group of VNFs, NS, etc.).

Physical Network Function Descriptor(PNFD): template that describes the connectivity requirements of Connection Point(s) attached to a Physical Network Function

NOTE: It is used by the NFVO to integrate PNF(s) into a NS.

quota: upper limit on specific types of resources, usually used to prevent excessive resource consumption in the VIM by a given consumer

NOTE: Quota is enforced by the VIM.

resource pool: logical grouping of NFVI hardware and software resources

NOTE 1: A resource pool can be solely based on a certain resource type (e.g. compute, storage, networking) or include a combination of them, and can span zero, one or multiple resource zones.

NOTE 2: An NFVI resource can be part of none, one or more than one resource pool.

resource zone: set of NFVI hardware and software resources logically grouped according to physical isolation and redundancy capabilities or to certain administrative policies for the NFVI

NOTE: The same resource cannot be part of two different resource zones.

EXAMPLE 1: Physical isolation may be achieved for example using a separate power supply, network equipment or physical building sites.

EXAMPLE 2: One example of resource zones are the Availability Zones in OpenStack.

service resource group: logical resource collection that groups a subset of service resource instances assigned to a tenant

NOTE: A service resource group can include NS, VNF, PNF, VNFFG and NFP.

tenant: one or more NFV-MANO service users sharing access to a set of physical,virtual or service resources

NOTE 1: This definition has been specialized from the term "tenant" as defined in Recommendation ITU-T Y.3500 [i.8].

NOTE 2: The "tenant" concept in NFV should not be confused with the "tenant" (aka "project") concept in OpenStack. The OpenStack implementation covers a subset of the overall functionalities required by multi-tenancy in NFV.

virtualised resource migration: process of relocating the virtualised resource from one physical node to another physical node

NOTE: Examples of physical nodes are compute nodes and storage nodes.

VNF healing: procedure that includes all virtualisation-related corrective actions to repair a faulty VNF, and/or its VNFC instances and internal VNF Virtual Link(s)

NOTE: "Virtualisation related corrective actions" refers to the corrective action(s) toward virtualised resources and associated VNF/VNFC instance(s), and/or internal VNF Virtual Link(s).

VNF Lifecycle Operation Granting: permission to perform a VNF lifecycle management operation and the resource management operations necessary to complete it, if any apply

NOTE: There is no guarantee that the necessary resources are available after the grant is given. Information on resource requirements to execute a VNF LCM request is included in the Grant request. Granting of individual resource management operations is not in scope of VNF Lifecycle Operation Granting.

VNF-related Resource Management in direct mode: mode of operation where the VNFM invokes on the VIM Virtualised Resources Management operations

NOTE 1: Resource reservation and quota management operations are out of the scope of this mode of operation, with the exception of query reservations and query quota.

NOTE 2: Virtualised Resources Management operations include allocation, migration, scaling, update, query, operation and termination of virtualised resources.

VNF-related Resource Management in indirect mode: mode of operation where the VNFM invokes on the NFVO Virtualised Resources Management operations and the NFVO in turn invokes them towards the VIM

NOTE 1: Resource reservation and quota management operations are out of the scope of this mode of operation, with the exception of query reservations and query quota.

NOTE 2: Virtualised Resources Management operations include allocation, migration, scaling, update, query, operation and termination of virtualised resources.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GS NFV 003 [i.2] and the following apply:

BSS	Business Support System
CP	Connection Point
DF	Deployment Flavour
EM	(Network) Element Manager
ETRI	Electronics and Telecommunications Research Institute
FB	Functional Block
FPGA	Field Programmable Gate Array
IFA	Interfaces and Architecture Working Group
IP	Internet Protocol
LCM	LifeCycle Management
NEC	Nippon Electric Company
NFP	Network Forwarding Path
NSD	Network Service Descriptor
NUMA	Non Uniform Memory Access
OS	Operating System
OSS	Operation Support System
PCIe	Peripheral Component Interface express
PM	Performance Management
PNFD	Physical Network Function Descriptor
SAP	Service Access Point
URI	Uniform Resource Identifier
VL	Virtual Link
WIM	WAN Infrastructure Manager

4 General Description

4.1 Introduction

Network Functions Virtualisation (NFV) adds new capabilities to communications networks and requires a new set of management and orchestration functions to be added to the current model of operations, administration, maintenance and provisioning. The NFV Management and Orchestration (NFV-MANO) architectural framework has the role to manage the infrastructure and orchestrate the resources needed by the Network Services (NSs) and Virtualised Network Functions (VNFs).

In order to guide the development of the specification of the interfaces exposed between the NFV-MANO Functional Blocks (FBs), it is important to have a clear and consolidated set of functional requirements to be addressed by the NFV-MANO. The present document is providing functional requirements on NFV MANO e.g. VNF lifecycle management (LCM), NS LCM, virtualised resource management, etc.

The functional requirements specified in the present document are mainly derived from functional requirements identified in ETSI GS NFV 002 [i.1], ETSI GS NFV 003 [i.2], ETSI GS NFV 004 [i.3], ETSI GS NFV-MAN 001 [i.4], ETSI GS NFV-SWA 001 [i.5], ETSI GS NFV-REL 001 [i.6] and ETSI GS NFV-INF 001 [i.7] or derived from concepts defined in these documents.

4.2 Overview

In order to provide systematic functional requirements, this document arranges the functional requirements by categorizing the requirements according to key operational functions of NFV-MANO, which are documented in ETSI GS NFV-MAN 001 [i.4].

Key operational function categories which are used to organize the requirements on NFV Orchestrator (NFVO), VNF Manager (VNFM) and Virtualised Infrastructure Manager (VIM) in the present document are listed below:

- Virtualised resource management.

- VNF LCM.
- NS LCM.
- VNF information management.
- NS information management.
- NFV performance management.
- NFV fault management.
- Security considerations.
- Software image management.
- NFV acceleration management.
- Multi-tenancy.

NOTE: This categorization groups related functional requirements together. Actual interface requirements derived from the functional requirements may be grouped differently, and/or individual interface requirements may be placed into a group that is different from the category of the related functional requirement.

5 General functional requirements

5.1 General functional requirements for virtualised resource management

The NFV-MANO architecture shall provide support to permit service providers to partially or fully virtualise the Network Functions (NFs) needed to create, deploy and operate the services they provide. In case of partial virtualisation, performance, management and operations of the non-virtualised NFs shall not be impacted.

The NFV-MANO architecture shall be able to support a NS composed of Physical Network Functions (PNFs) and VNFs implemented across multivendor environments.

The NFV-MANO architecture shall be able to manage NFV Infrastructure (NFVI) resources, in order to provide NSs and related VNFs and PNFs with the resources needed. Management of resources for PNFs shall be restricted to provisioning connectivity, e.g. necessary when a NS instance includes a PNF that needs to connect to a VNF.

The NFV-MANO architecture shall enable the NFVO and the VNFM to manage the virtualised resources needed for LCM of the VNFs. The NFV-MANO architecture shall enable deployments and implementations where:

- the NFVO is the only FB to manage the virtualised resources needed for the LCM of the VNF (**VNF-related Resource Management in indirect mode)**;
- the VNFM is the only FB to manage the virtualised resources needed for the LCM of the VNF **(VNF-related Resource Management in direct mode)**;
- the NFVO and the VNFM, both, manage the virtualised resources needed for the LCM of the VNF.

NOTE: This is a decision per VNFM whether it is the NFVO or the VNFM that manages the virtualised resources.

It is a deployment and implementation decision whether one option or both are deployed and implemented. All VNFs managed by one VNFM shall use the same option for virtualised resource management. The detailed requirements on the NFVO and the VNFM for each case are depicted in clauses 6.1 and 7.1.

In addition to managing the VNF-related virtualised resources as explained above, the NFV-MANO architecture shall enable the NFVO to manage the virtualised resources (i.e. network resources) that are needed for LCM of the NS(s).

Additionally, the NFV-MANO shall enable different models, per resource type, to facilitate availability of resources and to avoid resource contention. It shall be possible for the network operator, on a per NS basis, tenant basis or VNF basis, to select one of the following resource commitment models, or a combination of them:

- **Reservation** model, where resources are committed, but not allocated, to a particular consumer or consumer type. A reservation can have one of the following types (see details in clause A.2.8):

  1) reserving a set of resources considering particular virtualised resource configurations, i.e. reserving a number of virtualised containers, virtual networks, network ports and/or storage volumes;

  2) reserving virtualised resource capacity without considering particular resource configurations, i.e. reserving virtualised resource capacity of compute, storage and network resource types.

- **Quota/Allowance based** model, where the number of resources to be consumed by a particular consumer is limited to a defined amount or a percentage of resources; in this model, resources are committed upon demand from the consumer when a VNF or a NS is instantiated or scaled out, as long as those are within the limits established by the quota/allowance for that consumer or consumer type.

- **On demand**, where resources are committed when a VNF or a NS is instantiated or scaled out, as long as there are available resources for consumption.

The permitted allowance concept should be distinguished from the quota concept:

- Quota: enforced by the VIM. Quotas are usually used to prevent excessive resource consumption in the VIM by a given consumer.

- Permitted allowance: maintained at NFVO level. Permitted allowances might vary in granularity (VNFM, VNF, group of VNFs, NS, etc.) and are used to control resource consumption by VNFMs in relation to the granularity associated with the permitted allowance.

The detailed requirements on the affected FBs are depicted in clauses 6.1, 7.1 and 8.2.

5.2 General functional requirements for multi-tenancy

Multi-tenancy can be applied to all infrastructure and service resources which can be consumed from an NFV system and managed by NFV-MANO.

NOTE 1: The term "resource" as used in the present clause goes beyond the definition of NFV-resource as specified in the NFV Terminology document (ETSI GS NFV 003 [i.2]).

Figure 5.2-1 shows the entities relevant to multi-tenancy for any kind of resources.

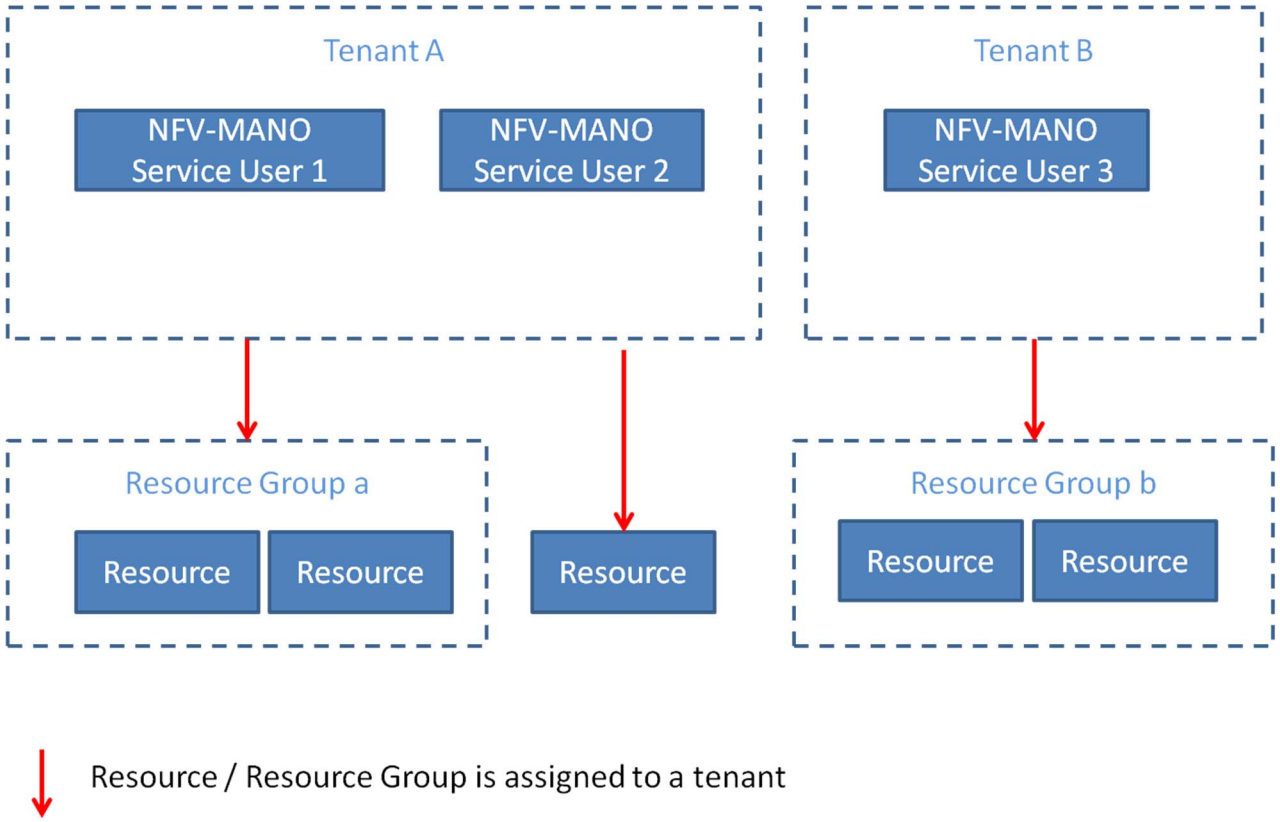

Figure 5.2-1: Entities relevant to multi-tenancy

Each FB may act as multiple tenants on the FBs from which it uses service or infrastructure resources. A service resource e.g. a VNF can be composed from multiple virtual resources from different tenants. Figure 5.2-2 shows an example how a VNFM may use tenants on the VIM.

EXAMPLE: The VNF (Resource Group a) is composed out of virtual resources from Resource Group c. The virtual resources in Resource Group c are assigned to Tenant C. Thus the VNFM has to identify as Tenant C to modify the virtual resources for VNF (Resource Group a). The VNF (Resource Group b) uses virtual resources assigned to Tenant D (Resource Group d) and Tenant E. Therefore the VNFM has to identify as Tenant D or Tenant E or both to modify the virtual resources for VNF (Resource Group b).

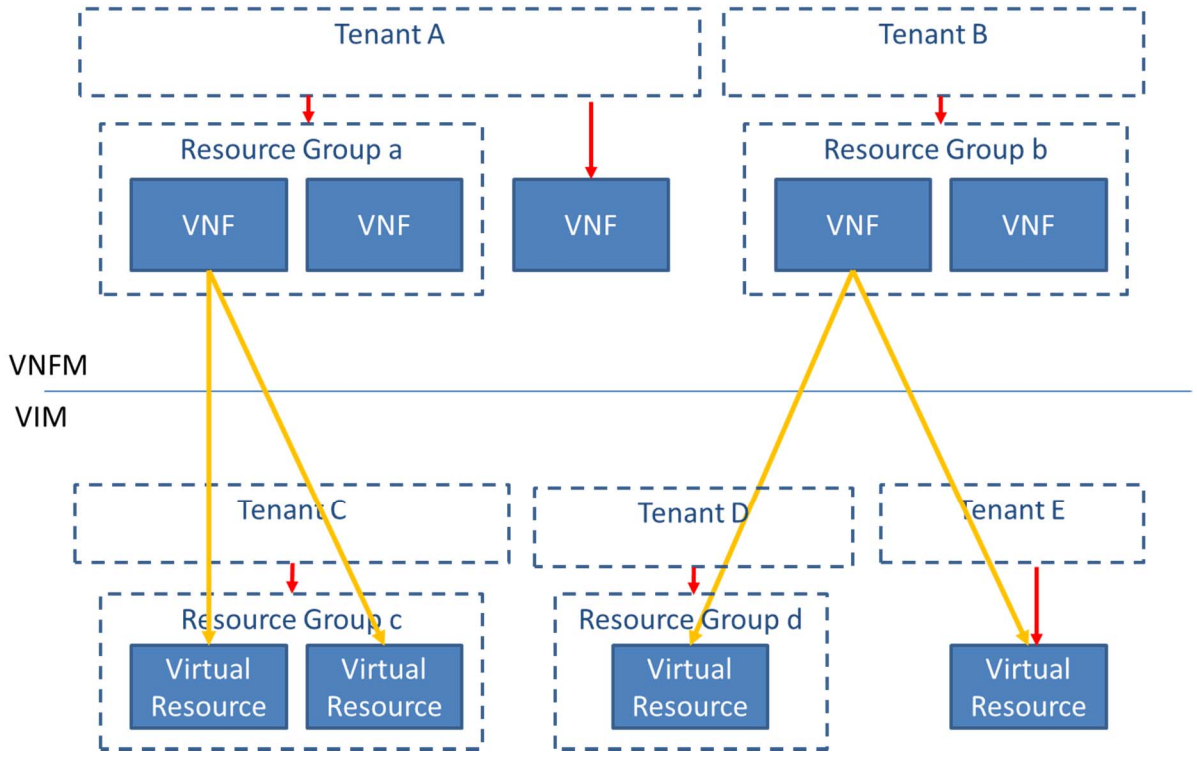

Figure 5.2-2: Example of how a VNFM may use tenants on a VIM

Since multi-tenancy exists for all kinds of service and infrastructure resources which can be used from an NFV-MANO service, tenants can be grouped based in the resources they use:

- A tenant to which virtual resources are assigned is referred to as an infrastructure tenant (Tenant C, D, E).
- A tenant to which VNFs are assigned is referred to as a VNF tenant (Tenant A, B).
- A tenant to which NSs are assigned is referred to as a NS tenant.

A resource group has different meaning for different resources which are being used:

- A resource group can be a "service resource group" containing VNFs, PNFs or NSs instances.
- A resource group can be an "infrastructure resource group" containing a set of virtual resources under the control of a VIM and belonging to a tenant.

6 Functional requirements for NFVO

6.1 Functional requirements for virtualised resource management

6.1.1 Functional requirements for general virtualised resource management

Table 6.1.1-1: Functional requirements for general virtualised resource management

Numbering	Functional requirements description
Nfvo.Gvrm.001	The NFVO shall support orchestration of actions related to virtualised resources managed by one or more VIMs.
Nfvo.Gvrm.002	The NFVO shall support the capability to mitigate conflicts in resource allocation in case of conflicting resource requests.
Nfvo.Gvrm.003	The NFVO shall support the capability to provide deployment-specific configuration information for virtualised resources related to NS.

6.1.2 Functional requirements for VNF-related resource management in indirect mode

Table 6.1.2-1: Functional requirements for VNF-related resource management in indirect mode

Numbering	Functional requirements description
Nfvo.VnfRmpbNfvo.001	When VNF-related Resource Management in indirect mode is applicable, the NFVO shall support the capability to request to the VIM the management of virtualised resources needed for VNFs instantiation, scaling and termination (see notes 1 and 4).
Nfvo.VnfRmpbNfvo.002	When VNF-related Resource Management in indirect mode is applicable, the NFVO shall support the capability to invoke resource management operations toward the VIM as requested by the VNFM.
Nfvo.VnfRmpbNfvo.003	When VNF-related Resource Management in indirect mode is applicable, the NFVO shall support the capability to receive notifications regarding the resources being allocated to or released from specific VNF instances, as well as regarding events and relevant fault reports related to those resources (see notes 1 and 3).
Nfvo.VnfRmpbNfvo.004	When VNF-related Resource Management in indirect mode is applicable, the NFVO shall support the capability to request allocation and update of resources in the different resource commitment models (see note 2).
Nfvo.VnfRmpbNfvo.005	When VNF-related Resource Management in indirect mode is applicable, the NFVO shall support the capability to request to the VIM affinity and anti-affinity policies for the VNF's virtualised resources.(see note 1).
NOTE 1: Virtual resources managed for the LCM of VNFs include compute and storage resources needed for VNF components as well as networking resources needed to ensure intra-VNF connectivity. NOTE 2: Resource commitment models are: reservation model, quota model and on-demand. NOTE 3: Events include NFVI outage and performance related events. NOTE 4: The management of virtualised resources includes allocation, update, scaling, termination, etc. of virtualised resources.	

6.1.3 Functional requirements for VNF-related resource management in direct mode

Table 6.1.3-1: Functional requirements for VNF-related resource management in direct mode

Numbering	Functional requirements description
Nfvo.VnfRmpbVnfm.001	When VNF-related Resource Management in direct mode is applicable, the NFVO shall support the capability to provide appropriate information about VIM to enable the VNFM to access the VIM.

6.1.4 Functional requirements for NS-related resource management performed by the NFVO

Table 6.1.4-1: Functional requirements for VNF-related resource management performed by NFVO

Numbering	Functional requirements description
Nfvo.NsRmpbNfvo.001	The NFVO shall support the capability to issue requests to the VIM in order to allocate resources needed for the connectivity of NSs, identify current resource allocations associated with a particular NS instance, update current resources allocated to the NS instance or release resources that had been allocated to an NS instance (see note 1).
Nfvo.NsRmpbNfvo.002	The NFVO shall support the capability to query to the VIM about the resources that are allocated for the connectivity of the VNF Forwarding Graphs (VNFFGs) of specific NS instances.
Nfvo.NsRmpbNfvo.003	The NFVO shall support the capability to receive notifications of the resources that are allocated to or released from specific NS instances as well as events and relevant fault reports related to those resources (see notes 1 and 2).
NOTE 1: Resources needed for the connectivity of NSs include networks, subnets, ports, addresses, links and forwarding rules, and are used for the purpose of ensuring inter-VNF connectivity. NOTE 2: Events include NFVI outage and performance related events.	

6.1.5 Functional requirements for resource reservation management

Table 6.1.5-1: Functional requirements for resource reservation management

Numbering	Functional requirements description
Nfvo.Rrm.001	The NFVO shall support the capability to request creation, query, update and termination of virtualised resource reservation to corresponding VIM(s) as part of NS LCM, VNF LCM, and VNF lifecycle granting procedures, and during configuration/reconfiguration of resources in the NFVI Point of Presence(s) (NFVI-PoPs).
Nfvo.Rrm.002	The NFVO shall support the capability to consider affinity/anti-affinity rules for resource reservation management.
Nfvo.Rrm.003	The NFVO shall support the capability to receive change notification regarding to virtualised resource reservation.
Nfvo.Rrm.004	When a resource reservation model is used, the NFVO shall support the capability to provide to VNFM resource reservation identification information.

6.1.6 Functional requirements for virtualised resource capacity management

Table 6.1.6-1: Functional requirements for virtualised resource capacity management

Numbering	Functional requirements description
Nfvo.Vrcm.001	The NFVO shall support the capability to maintain information regarding the virtualised resources capacity and its usage at different granularities, including usage per VNFM or per NS (see note 1).
Nfvo.Vrcm.002	The NFVO shall support the capability to query information about resource zones managed by the VIM and about NFVI-PoP(s) administered by the VIM.
Nfvo.Vrcm.003	The NFVO shall support the capability to maintain information regarding the resource zones available on the connected VIMs.
Nfvo.Vrcm.004	The NFVO shall support the capability to retrieve information regarding the virtualised resources capacity and its usage at different granularities and levels, including (not limited to) total per NFVI-PoP and per resource zone (see note 2).
Nfvo.Vrcm.005	The NFVO shall support the capability to synchronize periodically and automatically, or on demand, the virtualised resource capacity information maintained in the NFVO with the information managed by the VIM(s).
Nfvo.Vrcm.006	The NFVO shall support the capability to configure thresholds for setting virtualised resource capacity shortage alarms at different granularities and levels, including (not limited to) per NFVI-PoP and per resource zone.
Nfvo.Vrcm.007	The NFVO shall support the capability to notify about virtualised resource capacity shortage.
Nfvo.Vrcm.008	The NFVO shall support the capability to receive the notification from VIM related to the changes to NFVI capacity information.
NOTE 1: This information can be maintained for multiple uses, including statistics, analytics, granting VNF requests, management of NS, determining placement for VNFs on certain NFVI-PoPs and resource zones, for general network planning, etc. Refer to annex B for further information. NOTE 2: The capacity information can include information related to available, allocated, reserved and total virtualised resource capacity.	

6.1.7 Functional requirements for virtualised resource performance management

Table 6.1.7-1: Functional requirements for virtualised resource performance management

Numbering	Functional requirements description
Nfvo.Vrpm.001	The NFVO shall support the capability to invoke the virtualised resource performance management operations on the virtualised resources for the NS(s) it manages (see notes 1 and 2).
Nfvo.Vrpm.002	The NFVO shall support the capability to receive performance information related to virtualised resources for the NS(s) it manages (see note 2).
Nfvo.Vrpm.003	The NFVO shall support the capability to map to the NS(s) the received performance information related to virtualised resources (see note 2).
NOTE 1: The virtualised resource performance management can include: setting threshold conditions on the performance information collected by the VIM for specific virtualised resource(s), creating Performance Management (PM) jobs by specifying different limitations and conditions for collecting and reporting of performance information from specified virtualised resource(s), etc. NOTE 2: The virtualised resources mentioned in the requirements above are those used by the NS, but not used by any of the contained VNF instances, e.g. Virtual Links (VLs) between VNFs.	

6.1.8 Functional requirements for virtualised resource fault management

Table 6.1.8-1: Functional requirements for virtualised resource fault management

Numbering	Functional requirements description
Nfvo.Vrfm.001	The NFVO shall support the capability to collect fault information related to the virtualised resources assigned to NS(s) that it manages.
Nfvo.Vrfm.002	The NFVO shall support the capability to correlate the virtualised network resource fault information with the impacted NS(s) that it is managing.
Nfvo.Vrfm.003	The NFVO shall support the capability to request corrective operations on virtualised network resources to VIM in order to perform NS healing (see note).
NOTE: The virtualised network resources refer to the virtualised resources supporting the connectivity of the NS instance(s).	

6.1.9 Functional requirements for virtualised resource information management

Table 6.1.9-1: Functional requirements for virtualised resource information management

Numbering	Functional requirements description
Nfvo.Vrim.001	The NFVO shall support collection of information on virtualised resource that can be consumed in a VIM or across multiple VIMs.
Nfvo.Vrim.002	The NFVO shall support the capability to forward the information about resource shortage to the Operation Support System (OSS) as soon as it becomes available in the NFVO.
Nfvo.Vrim.003	The NFVO shall support the capability to receive the notifications regarding the changes of the information on consumable virtualised resources that can be provided by the VIM(s).

6.1.10 Functional requirements for Network Forwarding Path (NFP) management

Table 6.1.10-1: Functional requirements for NFP management

Numbering	Functional requirements description
Nfvo.Nfpm.001	The NFVO shall support the capability of requesting management of NFPs.
Nfvo.Nfpm.002	The NFVO should support the capability to provide or update the classification and selection rules applied to a specific NFP instance (see note 1).
Nfvo.Nfpm.003	The NFVO shall support the capability to receive the classification and selection rules applied to NFP(s) from an authorized entity (see note 2).
NOTE 1: The classification and selection rules applied to NFPs can be rules to classify and select NFPs. A NFP is allocated as the default path for specific types of traffic or packets. The rules are provided to VIM by NFVO, and VIM configures those rules in the Network Controllers to enable the Network Controllers to configure corresponding forwarding tables in NFVI network resources. NOTE 2: The classification and selection rules applied to NFPs are optionally included in the NS Descriptor (NSD). In cases when they are not included they can be provided to NFVO later to be assigned to an existing NFP. The authorized entity sending NFP rule to NFVO may include OSS/Business Support System (BSS).	

6.1.11 Functional requirements for quota management

Table 6.1.11-1: Functional requirements for quota management

Numbering	Functional requirements description
Nfvo.Qm.001	The NFVO shall support the capability to request the VIM to create the quota for a consumer of the virtualised resources.
Nfvo.Qm.002	The NFVO shall support the capability to request the VIM to change the quota for a consumer of the virtualised resources.
Nfvo.Qm.003	The NFVO shall support the capability to request the VIM to delete the quota for a consumer of the virtualised resources.
Nfvo.Qm.004	The NFVO shall support the capability to query to the VIM the information of the quota for a consumer of the virtualised resources.
Nfvo.Qm.005	The NFVO shall support the capability to receive change notification regarding virtualised resource quota.
Nfvo.Qm.006	The NFVO may support the capability to provide to the VNFM the information on available quota(s) applicable to this VNFM (see note 1 and note 2).
NOTE 1: The information on available quota(s) allows the VNFM to interact with the VIM to receive information regarding the quota(s) applied to the VNFM or the VNF(s) which the VNFM manages, when VNF-related Resource Management in direct mode is applicable. NOTE 2: The information on available quota(s) allows the VNFM to interact with the NFVO to receive information regarding the quota(s) applied to the VNFM or the VNF(s) which the VNFM manages, when VNF-related Resource Management in indirect mode is applicable.	

6.1.12 Functional requirements related to permitted allowance management

Table 6.1.12-1: Functional requirements related to permitted allowance management

Numbering	Functional requirements description
Nfvo.Pam.001	When an allowance model is used, it shall be possible for the NFVO to maintain and enforce permitted allowance at various granularity levels (VNFM, VNF, NS, etc.).
Nfvo.Pam.002	A permitted allowance shall be expressed either as a defined amount of resources or as percentages of the total available resources per type of resources.
Nfvo.Pam.003	When an allowance model is used, the NFVO shall support the capability to reject any granting requests from VNFM that would cause the corresponding permitted allowance to be exceeded (see note).
Nfvo.Pam.004	When an allowance model is used, the NFVO shall support the capability to manage the overall consumption of resources across all permitted allowances.
Nfvo.Pam.005	When an allowance model is used, the NFVO shall support the capability to provide notification when the permitted allowance reaches its limit.
Nfvo.Pam.006	When an allowance model is used, the NFVO shall support the capability to process a request for permitted allowance extension or permitted allowance reduction.
Nfvo.Pam.007	When an allowance model is used, the NFVO shall support the capability to arbitrate conflict in permitted allowance consumption (see example).
NOTE: NFVO might decide, based on policy, to extend a given allowance reaching its limit.	
EXAMPLE: An example of conflict can be in case when multiple concurrent resource allocations can be foreseen to exceed the allowance.	

6.2 Functional requirements for VNF lifecycle management

6.2.1 Functional requirements for VNF lifecycle management

Table 6.2.1-1: Functional requirements for VNF lifecycle management

Numbering	Functional requirements description
Nfvo.VnfLcm.001	The NFVO shall support the capability to process notifications about VNF lifecycle change.
Nfvo.VnfLcm.002	The NFVO shall support the capability of granting of the LCM requests.
Nfvo.VnfLcm.003	The NFVO shall support the capability to validate the lifecycle operation requests submitted to it, using information specified in the VNF Package.
Nfvo.VnfLcm.004	The NFVO shall support the capability to request changing the state of a VNF instance (see note 1).
Nfvo.VnfLcm.005	When NFVO is the consumer of the VNF LCM operation, the NFVO shall support the capability to query the status of the ongoing LCM operation.
Nfvo.VnfLcm.006	The NFVO shall support the capability to query information about a VNF instance.
Nfvo.VnfLcm.007	The NFVO shall support the capability to request the creation and deletion of the identifier of a VNF instance.
Nfvo.VnfLcm.008	The NFVO shall support the capability to request VNFM to conduct error handling operation(s) after the VNF life cycle operation occurrence fails(see notes 2 and 3).
NOTE 1: Change state refers to start and stop a VNF instance/VNF Component (VNFC) instances(s). These operations are complementary to instantiate/create a VNF or terminate a VNF. NOTE 2: It is up to the protocol design stage to design the detail error handling operation(s). NOTE 3: It depends on the VNF capabilities and is declared in the VNFD whether and how the operation(s) are supported by a particular VNF.	

6.2.2 Functional requirements for VNF instantiation

Table 6.2.2-1: Functional requirements for VNF instantiation

Numbering	Functional requirements description
Nfvo.VnfI.001	The NFVO shall support the capability to request the instantiation of a VNF instance.
Nfvo.VnfI.002	The NFVO shall support the capability to send to the VNFM, as part of the VNF instantiation request, input parameters specific for the VNF instance being instantiated.

6.2.3 Functional requirements for VNF scaling

NOTE: The LCM operations that expand or contract a VNF instance include scale in, scale out, scale up, scale down. Not all VNFs support all these operations, which implies that the set of operations that a VNFM will be able to perform on a VNF instance will depend on the VNF capabilities.

Table 6.2.3-1: Functional requirements for VNF scaling

Numbering	Functional requirements description
Nfvo.VnfS.001	The NFVO shall support the capability to request expanding the capacity of a VNF instance (see note 1).
Nfvo.VnfS.002	The NFVO shall support the capability to request contracting the capacity of a VNF instance (see note 2).
NOTE 1: Expansion can either be performed by scaling out or scaling up. NOTE 2: Contraction can either be performed by scaling in or scaling down.	

6.2.4 Functional requirements for VNF termination

Table 6.2.4-1: Functional requirements for VNF termination

Numbering	Functional requirements description
Nfvo.VnfT.001	The NFVO shall support the capability to request the termination of a VNF instance.
Nfvo.VnfT.002	The NFVO shall support the capability to check the dependencies between VNF instances before granting the termination of a particular VNF instance.

6.3 Functional requirements for NS lifecycle management

6.3.1 Functional requirements for NS lifecycle management

Table 6.3.1-1: Functional requirements for NS lifecycle management

Numbering	Functional requirements description
Nfvo.NsLcm.001	The NFVO shall ensure the integrity of data related to the NS instances (e.g. descriptors, software images, records, etc.) against loss and corruption from hardware/software failures and against tampering with such data by unauthorized parties.
Nfvo.NsLcm.002	The NFVO shall support the capability to use the deployment information from the NSD for the NS LCM.
Nfvo.NsLcm.003	The NFVO shall support the capability to notify about the following events related to NS lifecycle changes: • The start of the lifecycle procedure. • The result of the lifecycle procedure.
Nfvo.NsLcm.004	The NFVO shall support the capability to execute scheduled NS lifecycle operations.
Nfvo.NsLcm.005	The NFVO shall support the capability to manage the connectivity between the VNFs, nested NS(s) and PNF(s) that are part of the NS.
Nfvo.NsLcm.006	The NFVO shall support the capability to provide the status of a NS LCM operation in response to a request.

6.3.2 Functional requirements for NS instantiation

Table 6.3.2-1: Functional requirements for NS instantiation

Numbering	Functional requirements description
Nfvo.NsI.001	The NFVO shall support the capability to manage the instantiation of a NS instance.
Nfvo.NsI.002	The NFVO shall support the capability to invoke the instantiation of the constituent VNFs for a NS.
Nfvo.NsI.003	The NFVO shall support the capability to invoke the creation of the constituent VLs for a NS.
Nfvo.NsI.004	The NFVO shall support the capability to create VNFFG(s) for a NS (see note 1).
Nfvo.NsI.005	The NFVO shall support the capability to instantiate a NS which includes existing VNF instances (see note 2).
NOTE 1: The VNFFG(s) of a NS can include PNF(s). NOTE 2: The VNF descriptors (VNFDs) of the existing VNF instances shall be referenced from the NSD of the NS instance being instantiated. The existing VNF instances may need to be modified as part of NS instantiation.	

6.3.3 Functional requirements for NS scaling

Table 6.3.3-1: Functional requirements for NS scaling

Numbering	Functional requirements description
Nfvo.NsS.001	The NFVO shall support the capability to manage the expansion of a NS instance (see note 1).
Nfvo.NsS.002	The NFVO shall support the capability to manage the contraction of a NS instance (see note 2).
Nfvo.NsS.003	The NFVO shall support the capability to request to scale a VNF instance as part of the expansion/contraction of a NS instance.
Nfvo.NsS.004	The NFVO shall support the capability to evaluate the impact on NS instance(s) it manages when scaling needs to be performed on a component instance (i.e. a VNF or nested NS) shared or not.
NOTE 1: Expansion can either be performed by increasing the number of the existing VNF instance(s) or expansion of the existing VNF instance(s).	
NOTE 2: Contraction can either be performed by decreasing the number of the existing VNF instance(s) or contraction of the existing VNF instance(s).	

6.3.4 Functional requirements for NS updating

Table 6.3.4-1: Functional requirements for NS updating

Numbering	Functional requirements description
Nfvo.NsU.001	The NFVO shall support the capability to manage the update of a NS instance.
Nfvo.NsU.002	The NFVO shall support the capability to add new VNF(s)/VL(s)/VNFFG(s)/PNF(s)/Nested NS(s)/Service Access Point(s) (SAPs) to an existing NS in order to perform NS update.
Nfvo.NsU.003	The NFVO shall support the capability to remove the VNF(s)/VL(s)/VNFFG(s)/PNF(s)/Nested NS(s)/SAP(s) from an existing NS in order to perform NS update.
Nfvo.NsU.004	The NFVO shall support the capability to update the existing VNF(s)/VL(s)/VNFFG(s) involved in an existing NS (see note 1).
Nfvo.NsU.005	The NFVO shall support the capability to add existing VNF instance(s) to an existing NS (see note 2).
NOTE 1: The operation of updating the existing VNF(s) involved in an existing NS is embedded in the fine grained NS LCM operation, and can include: changing the Deployment Flavour (DF) of VNF instances, changing the operational state of a VNF instance, modifying VNF information data, modifying VNF configuration data.	
NOTE 2: The VNFDs of the existing VNF instances shall be referenced from the NSD of the NS instance being updated. The existing VNF instance(s) may need to be modified as part of NS update.	

6.3.5 Functional requirements for NS termination

Table 6.3.5-1: Functional requirements for NS termination

Numbering	Functional requirements description
Nfvo.NsT.001	The NFVO shall support the capability to terminate a NS instance.
Nfvo.NsT.002	The NFVO shall support the capability to request the termination of VNF instance(s) in order to perform NS termination.
Nfvo.NsT.003	The NFVO shall support the capability to retain a VNF instance currently used by another NS instance (i.e. other than the NS being terminated) when performing NS termination.
Nfvo.NsT.004	The NFVO shall support the capability to return information about retained VNF instance(s) used by another NS instance (i.e. other than the NS being terminated) when performing NS termination.

6.4 Functional requirements for VNF configuration management

Configuration parameters referred in this clause include those set at initial configuration and any other configurable parameter declared in the VNFD.

Table 6.4-1: Functional requirements for VNF configuration management

Numbering	Functional requirements description
Nfvo.VnfCm.001	The NFVO shall support the capability to invoke a request to set initial configuration parameters for a VNF instance.
Nfvo.VnfCm.002	The NFVO shall support the capability to invoke a request to update configuration parameters for a VNF instance.

6.5 Functional requirements for VNF information management

6.5.1 Functional requirements for VNF Package management

Table 6.5.1-1: Functional requirements for VNF Package management

Numbering	Functional requirements description
Nfvo.VnfPkgm.001	The NFVO shall support the capability of management of VNF Packages (see note 1).
Nfvo.VnfPkgm.002	The NFVO shall support the capability to verify the integrity and authenticity of the VNF Package.
Nfvo.VnfPkgm.003	The NFVO shall support the capability to verify that all mandatory information in the VNF Package is present and complies with the standard for this information.
Nfvo.VnfPkgm.004	The NFVO shall support the capability to notify about the state changes of the VNF Package.
Nfvo.VnfPkgm.005	The NFVO shall support the capability to validate the integrity and authenticity of the VNFD in the VNF Package.
Nfvo.VnfPkgm.006	The NFVO shall support the capability to notify about the on-boarding of the VNF Package.
Nfvo.VnfPkgm.007	The NFVO shall support the capability to request modifying the VNF instance information in the VNFM to refer to a different VNF Package when no conflicts exist between the previous and the newly referred VNF Package (see note 2).
Nfvo.VnfPkgm.008	The NFVO shall support the capability to allow on-boarding of different VNF Packages of a VNF.
NOTE 1: The VNF Packages management can include on-boarding, enable/disable, query and delete of VNF Packages. NOTE 2: A related use case is to keep NFV MANO in sync about a VNF application software modification (see clause 5.7 of ETSI GS NFV-IFA 011 [1])	

6.5.2 Functional requirements for VNF instance information management

Table 6.5.2-1: Functional requirements for VNF instance information management

Numbering	Functional requirements description
Nfvo.VnfIIm.001	The NFVO shall support the capability to query information on the mapping relationship between the VNF instance(s) and associated virtualised resource.

6.6 Functional requirements for NS information management

6.6.1 Functional requirements for NSD management

Table 6.6.1-1: Functional requirements for NSD management

Numbering	Functional requirements description
Nfvo.NsDtm.001	The NFVO shall support the capability of management of NSD (see note).
Nfvo.NsDtm.002	The NFVO shall support the capability to verify the integrity of the provided NSD.
Nfvo.NsDtm.003	The NFVO shall support the capability to verify that all mandatory information in the NSD is present and complies with the standard for this information.
Nfvo.NsDtm.004	The NFVO shall support the capability to report information related to the operation result of NSD.
Nfvo.NsDtm.005	The NFVO shall support the capability to perform version control of on-boarded NSDs.
NOTE: The NSD management can include on-boarding, update, enable/disable, query and delete of NSD.	

6.6.2 Functional requirements for NS instance information management

Table 6.6.2-1: Functional requirements for NS instance information management

Numbering	Functional requirements description
Nfvo.NsIim.001	The NFVO shall support the capability to receive run-time data related to NS instances that it has created (see note).
NOTE: Run-time data of NS instance can be information related to the run-time virtualised resource allocated to a NS instance, such as performance measurements related to resources of this instance or the VNF instance within this NS instance, resource reservation information for NFVI resources reserved for this NS instance, etc.	

6.6.3 Functional requirements for PNF Descriptor (PNFD) management

Table 6.6.3-1: Functional requirements for PNFD management

Numbering	Functional requirements description
Nfvo.PnfDtm.001	The NFVO shall support the capability of management of PNFD (see note).
Nfvo.PnfDtm.002	The NFVO shall support the capability to verify the integrity of the provided PNFD.
Nfvo.PnfDtm.003	The NFVO shall support the capability to verify that all mandatory information in the PNFD is present and complies with the standard for this information.
Nfvo.PnfDtm.004	The NFVO shall support the capability to report information related to the operation result of PNFD management.
Nfvo.PnfDtm.005	The NFVO shall support the capability to perform version control of on-boarded PNFD(s).
NOTE: The PNFD management can include on-boarding, update, query and delete of PNFD.	

6.7 Functional requirements for NS performance management

Table 6.7-1: Functional requirements for NS performance management

Numbering	Functional requirements description
Nfvo.NsPm.001	The NFVO shall support the capability of performance management of NSs.
Nfvo.NsPm.002	The NFVO shall support the capability to notify availability of performance information on the NSs it manages (see note).
Nfvo.NsPm.003	In response to a query, the NFVO shall support the capability to provide the information about active PM jobs which match the filter criteria.
NOTE: Performance information on a given NS results from either collected performance information of the virtualised resources impacting the connectivity of this NS instance or VNF performance information issued by the VNFM for the VNFs that is part of this NS instance. The latter performance information also results from collected performance information of the virtualised resources that are mapped to this VNF instance.	

6.8 Functional requirements for VNF fault management

6.8.1 Functional requirements for virtualisation-related fault management

Table 6.8.1-1: Functional requirements for virtualisation-related fault management

Numbering	Functional requirements description
Nfvo.VirFm.001	The NFVO shall support the capability to request VNF healing to VNFM.
Nfvo.VirFm.002	The NFVO shall support the capability to collect notifications about alarms on a VNF instance as a consequence of state change in the virtualised resources used by the VNF.

6.9 Functional requirements for NS fault management

Table 6.9-1: Functional requirements for NS fault management

Numbering	Functional requirements description
Nfvo.NsFm.001	The NFVO shall support the capability to provide notifications of fault information related to the NSs it manages (see notes 1 and 2).
Nfvo.NsFm.002	The NFVO shall support the capability to provide fault information on the NSs it manages (see notes 1 and 2).
Nfvo.NsFm.003	The NFVO shall support the capability to provide notifications of changes in fault information related to the NSs it manages (see notes 1 and 2).
Nfvo.NsFm.004	The NFVO shall support the capability to perform automated or on-demand healing on the NSs it manages.
Nfvo.NsFm.005	The NFVO shall support the capability to notify the errors during NS lifecycle procedure.
Nfvo.NsFm.006	The NFVO shall support the capability to evaluate the impact on NS instance(s) it manages when NS healing needs to be performed on a component instance (i.e. a VNF or nested NS) shared or not.
NOTE 1: Fault information on a given NS results from either a collected virtualised resource fault impacting the connectivity of the NS instance or a VNF alarm (see clause 7.6) issued by the VNFM for a VNF that is part of this NS instance. NOTE 2: Fault information on a given NS instance can include the information related to the alarm (e.g. alarm created, alarm cleared, etc.), alarm causes and identification of this NS instance and fault information concerning the virtualised resources supporting the constituent VNFs for this NS instance and the virtualised resources supporting the connectivity of this NS instance.	

6.10 Functional requirements for infrastructure resource management

Table 6.10-1: Functional requirements for infrastructure resource management

Numbering	Functional requirements description
Nfvo.Irm.001	The NFVO shall support the capability to collect the information about NFVI-PoPs, such as network connectivity endpoints and geographical locations (see note).
NOTE: This information may be used by the NFVO for building and keeping NFVI-PoP topology information.	

6.11 Functional requirements for security consideration

Table 6.11-1: Functional requirements for security consideration

Numbering	Functional requirements description
Nfvo.Sc.001	The NFVO shall support the capability to validate that the received message is from an authenticated and authorized consumer.
Nfvo.Sc.002	The NFVO shall support the capability to verify the integrity of the received message.
Nfvo.Sc.003	The NFVO shall support the capability to encrypt the sent message or decrypt the received message using negotiated key and algorithm to or from an authenticated and authorized consumer or producer.

6.12 Functional requirements for software image management

NOTE: The software image(s) is/are at virtualisation container level, e.g. Virtual Machine (VM) images.

Table 6.12-1: Functional requirements for software image management

Numbering	Functional requirements description
Nfvo.Sim.001	The NFVO shall support the capability to distribute the software image(s) to one or more VIMs.
Nfvo.Sim.002	The NFVO shall support the capability to query the VIM for information on the software images.
Nfvo.Sim.003	The NFVO shall support the capability to invoke software image deletion request to VIM on those software image(s) which were distributed by the NFVO and managed by VIM.
Nfvo.Sim.004	The NFVO shall support the capability to invoke updating the user-defined metadata for the selected software images which were distributed by the NFVO and managed by VIM (see note).
NOTE: The metadata may, but need not come from VNF Package.	

6.13 Functional requirements for NFV acceleration management

Table 6.13-1: Functional requirements for NFV acceleration management

Numbering	Functional requirements description
Nfvo.NfvAm.001	When VNF-related Resource Management in indirect mode is applicable, the NFVO shall support the capability to request to the VIM the allocation and release of necessary acceleration resources to meet acceleration capability requirement(s) of the VNFs (see note).
Nfvo.NfvAm.002	The NFVO shall support the capability to retrieve acceleration capability requirement(s) of the VNF from the VNFD (see note).
Nfvo.NfvAm.003	The NFVO shall support the capability to receive acceleration capability information from VIM (see note).
Nfvo.NfvAm.004	The NFVO shall support the capability to query acceleration capability information from VIM (see note).
Nfvo.NfvAm.005	The NFVO shall support the capability to select a VIM that has enough available acceleration capabilities to support acceleration capability requirement(s) of the VNF (see note).
NOTE: The acceleration capabilities can include type, capacity, Non-Uniform Memory Architecture (NUMA) support, etc.	

6.14 Functional requirements for multi-tenancy

Table 6.14-1: Functional requirements for multi-tenancy

Numbering	Functional requirements description
Nfvo.Mtm.001	The NFVO shall support the capability of management of NS tenants (see note 1).
Nfvo.Mtm.002	The NFVO may support the capability of management of infrastructure tenants (see note 1) and mapping of such infrastructure tenants to the VIM managed infrastructure tenants in case VNF-related resource management in indirect mode is applicable.
Nfvo.Mtm.003	The NFVO shall support the capability to assign on-boarded VNF Packages and NSDs to one or more NS tenants (see note 2).
Nfvo.Mtm.004	The NFVO shall support the capability to on-board VNF Packages and NSDs for a tenant.
Nfvo.Mtm.005	The NFVO shall allow a tenant to instantiate VNFs and NSs using VNF Packages and NSD s assigned to this tenant or shared VNF Packages and NSDs.
Nfvo.Mtm.006	The NFVO shall support the capability to limit the scope of operations only to the service and infrastructure resource groups assigned to the requesting tenant.
NOTE 1: The management of tenants include: • create, read, update, delete tenants; • associate a tenant with a single or multiple consumer of the Os-Ma interface, defining also the role; • associate a tenant to a "service resource group", i.e. to a collection of NSs; • associate a tenant to "infrastructure resource group" managed by a VIM or to multiple "infrastructure resource groups" managed by different VIMs; • manage the association of a tenant and a VNFM if a VNF specific VNFM is used. NOTE 2: A VNF Package or NSD which is assigned to a single tenant is commonly referred to as a private VNF Package or NSD of this tenant. A VNF Package or NSD which is assigned to all tenants is commonly referred to as a public VNF Package or NSD. A VNF Package or NSD which is assigned to more than one tenant is commonly referred to as a shared VNF Package or NSD.	

7 Functional requirements for VNFM

7.1 Functional requirements for virtualised resource management

7.1.1 Functional requirements for virtualised resource management

Table 7.1.1-1: Functional requirements for virtualised resource management

Numbering	Functional requirements description
Vnfm.Vrm.001	The VNFM shall support providing deployment-specific configuration information for virtualised resource related to VNF instance(s).
Vnfm.Vrm.002	The VNFM shall support the capability to maintain the mapping between a VNF instance and the virtualised resources of the VNF instance (see note).
Vnfm.Vrm.003	The VNFM shall support the capability to request resource allocation for VNF instance that meet the requirements specified by the VNF provider.
NOTE: The VNFM maintains the mapping between virtualised resources and the VNF in order to, for example: • Map virtualised resources fault information, performance information and change notifications to corresponding VNFCs. • Request management of virtualised resources to support current instantiated VNFCs, instantiate new VNFCs, terminate existing instantiated VNFCs, and internal connectivity in the VNF (VLs and connection points (CPs)).	

7.1.2 Functional requirements for VNF-related resource management in indirect mode

Table 7.1.2-1: Functional requirements for VNF-related resource management in indirect mode

Numbering	Functional requirements description
Vnfm.VnfRmpbNfvo.001	When VNF-related Resource Management in indirect mode is applicable, the VNFM shall support the capability to request to NFVO the management of virtualised resources needed for VNFs instantiation, scaling and termination (see note).
Vnfm.VnfRmpbNfvo.002	When VNF-related Resource Management in indirect mode is applicable, the VNFM shall support the capability to invoke resource management requests towards the NFVO to allocate resources that meet the requirements specified by the VNF provider.
NOTE: The management of virtualised resources includes allocation, update, scaling, termination, etc. of virtualised resources.	

7.1.3 Functional requirements for VNF-related resource management in direct mode

Table 7.1.3-1: Functional requirements for VNF-related resource management in direct mode

Numbering	Functional requirements description
Vnfm.VnfRmpbVnfm.001	When VNF-related Resource Management in direct mode is applicable, the VNFM shall support the capability to request to the VIM the management of virtualised resources needed for VNFs instantiation, scaling and termination (see notes 1 and 4).
Vnfm.VnfRmpbVnfm.002	When VNF-related Resource Management in direct mode is applicable, the VNFM shall support the capability to query to the VIM about the resources being allocated to VNF instances it manages (see note 1).
Vnfm.VnfRmpbVnfm.003	When VNF-related Resource Management in direct mode is applicable, the VNFM shall support the capability to receive notifications regarding the resources being allocated to or released from specific VNF instances, as well as regarding events and relevant fault reports related to those resources (see notes 1 and 3).
Vnfm.VnfRmpbVnfm.004	When VNF-related Resource Management in direct mode is applicable, the VNFM shall support the capability to request allocation and update of resources in the different resource commitment models (see notes 2 and 5).
Vnfm.VnfRmpbVnfm.005	When VNF-related Resource Management in direct mode is applicable, the VNFM shall support the capability to request to the VIM affinity and anti-affinity policies for the VNF's virtualised resources (see note 1).
Vnfm.VnfRmpbVnfm.006	When VNF-related Resource Management in direct mode is applicable and a resource reservation model is used, the VNFM shall support the capability to use resource reservation identification information obtained from the NFVO to request allocation of virtualised resources for a VNF.
Vnfm.VnfRmpbVnfm.007	When VNF-related Resource Management in direct mode is applicable, the VNFM shall support the capability to obtain appropriate information to enable the VNFM to access the VIM.
NOTE 1: Virtual resources managed for the LCM of VNFs include compute and storage resources needed for VNF components as well as networking resources needed to ensure intra-VNF connectivity. NOTE 2: Resource commitment models are: reservation model, quota model and on-demand. NOTE 3: Events include NFVI outage and performance related events. NOTE 4: The management of virtualised resources includes allocation, update, scaling, termination, etc. of virtualised resources. NOTE 5: This does not imply that the VNFM can manage resource reservations and quotas, which are NFVO's prerogatives.	

7.1.4 Functional requirements for resource reservation management

Table 7.1.4-1: Functional requirements for resource reservation management

Numbering	Functional requirements description
Vnfm.Rrm.001	The VNFM shall support the capability to receive change notification regarding virtualised resource reservation.
Vnfm.Rrm.002	The VNFM shall support the capability to query information regarding virtualised resource reservation.

7.1.5 Functional requirements for virtualised resource performance management

Table 7.1.5-1: Functional requirements for virtualised resource performance management

Numbering	Functional requirements description
Vnfm.Vrpm.001	The VNFM shall support the capability to invoke the virtualised resource performance management operations on the virtualised resources for the VNF instance(s) it manages (see note).
Vnfm.Vrpm.002	The VNFM shall support the capability to receive performance information related to virtualised resources for the VNF instance(s) it manages.
Vnfm.Vrpm.003	The VNFM shall support the capability to map to the VNF instances the received performance information related to virtualised resources.
NOTE: The virtualised resource performance management can include setting threshold conditions on the performance information collected by the VIM for specific virtualised resource(s), creating PM jobs by specifying different limitations and conditions for collecting and reporting of performance information from specified virtualised resource(s), etc.	

7.1.6 Functional requirements for virtualised resource fault management

Table 7.1.6-1: Functional requirements for virtualised resource fault management

Numbering	Functional requirements description
Vnfm.Vrfm.001	The VNFM shall support the capability to collect fault information related to the virtualised resources assigned to VNF instance(s) that it manages.
Vnfm.Vrfm.002	The VNFM shall support the capability to correlate virtualised resource fault information with the impacted VNF(C) instance(s) that it manages.

7.1.7 Functional requirements for virtualised resource information management

Table 7.1.7-1: Functional requirements for virtualised resource information management

Numbering	Functional requirements description
Vnfm.Vrim.001	The VNFM should support the capability to query information regarding consumable virtualised resources that can be provided by the VIM.
Vnfm.Vrim.002	The VNFM shall support the capability to receive the notifications regarding the changes of the information on consumable virtualised resources that can be provided by the VIM.

7.1.8 Functional requirements for quota management

Table 7.1.8-1: Functional requirements for quota management

Numbering	Functional requirements description
Vnfm.Qm.001	The VNFM should support the capability to query the information on the quota(s) that apply to this VNFM or to the VNF(s) that this VNFM manages.
Vnfm.Qm.002	The VNFM should support the capability to receive change notification regarding the quota constraint(s) that apply to this VNFM or to the VNF that this VNFM manages.
Vnfm.Qm.003	The VNFM may support the capability to receive information from NFVO on available quota(s) applicable to this VNFM (see notes 1 and note 2).
NOTE 1: The information on available quota(s) allows the VNFM to interact with the VIM to receive information regarding the quota(s) applied to the VNFM or the VNF(s) which the VNFM manages, when VNF-related Resource Management in direct mode is applicable. NOTE 2: The information on available quota(s) allows the VNFM to interact with the NFVO to receive information regarding the quota(s) applied to the VNFM or the VNF(s) which the VNFM manages, when VNF-related Resource Management in indirect mode is applicable.	

7.1.9 Functional requirements related to permitted allowance management

Table 7.1.9-1: Functional requirements related to permitted allowance management

Numbering	Functional requirements description
Vnfm.Pam.001	When an allowance model is used, the VNFM shall support the capability to notify its resource consumption.

7.2 Functional requirements for VNF lifecycle management

7.2.1 Functional requirements for VNF lifecycle management

NOTE: Not all VNFs support all the VNF lifecycle operations which associate with the capabilities defined in this document. For any given VNF, the VNFM will only be able to perform those operations that are supported by that VNF.

Table 7.2.1-1: Functional requirements for VNF lifecycle management

Numbering	Functional requirements description
Vnfm.VnfLcm.001	The VNFM shall support the capability to notify about the following events related to VNF lifecycle changes: • the start of the lifecycle procedure; • the end and the result of the lifecycle procedure, including errors during the procedure, if any.
Vnfm.VnfLcm.002	The VNFM shall support the capability to notify about the type of VNF lifecycle change, the addition/deletion of VNFCs, and about the changes on virtualised resources associated to VNFC(s) as result of the VNF lifecycle change.
Vnfm.VnfLcm.003	The VNFM shall support the capability to notify about virtual networks and CPs that are added/deleted as part of the VNF lifecycle operation.
Vnfm.VnfLcm.004	The VNFM shall support the capability to validate the lifecycle operation requests it processes, using information specified in the VNF Package.
Vnfm.VnfLcm.005	The VNFM shall support the capability to change the state of a VNF instance/VNFC instance(s) (see note 1).
Vnfm.VnfLcm.006	The VNFM shall support the capability to use the deployment information from the VNFD for the VNF LCM.
Vnfm.VnfLcm.007	The VNFM shall support the capability to provide the status of a VNF LCM operation in response to a query.
Vnfm.VnfLcm.008	The VNFM shall support the capability to request an operation granting before executing the VNF lifecycle operation procedure, in procedures that can require changes in terms of resources usage or impact NS management (see note 2).
Vnfm.VnfLcm.009	The VNFM shall support the capability to switch the DF of a VNF instance.
Vnfm.VnfLcm.010	The VNFM shall support the capability to create and delete the identifier of the VNF instance which it manages.
Vnfm.VnfLcm.011	The VNFM shall support the capability to conduct VNF error handling operation(s) after the VNF life cycle operation occurrence fails(see notes 3 and 4).
NOTE 1: Change state refers to start and stop a VNF instance/VNFC instance(s). These operations are complementary to instantiate/create a VNF, or terminate a VNF. NOTE 2: This includes procedures related to instantiation, scaling, healing, and termination of VNF instances. NOTE 3: It is up to the protocol design stage to design the detail error handling operation(s). NOTE 4: It depends on the VNF capabilities and is declared in the VNFD whether and how the operation(s) are supported by a particular VNF.	

7.2.2 Functional requirements for VNF instantiation

Table 7.2.2-1: Functional requirements for VNF instantiation

Numbering	Functional requirements description
Vnfm.VnfI.001	The VNFM shall support the capability to manage the instantiation of a VNF instance.
Vnfm.VnfI.002	The VNFM shall support the capability to request VIM to allocate resources for the VNF instance being instantiated.
Vnfm.VnfI.003	The VNFM shall support the capability to configure deployment specific parameters for the VNF instance being instantiated.
Vnfm.VnfI.004	The VNFM shall support the capability to store the information of the allocated resources and configured deployment specific parameters for the instantiated VNF.

7.2.3 Functional requirements for VNF scaling

NOTE: The LCM operations that expand or contract a VNF instance include scale in, scale out, scale up, scale down. Not all VNFs support all these operations, which implies that the set of operations that a VNFM will be able to perform on a VNF instance will depend on the VNF capabilities.

Table 7.2.3-1: Functional requirements for VNF scaling

Numbering	Functional requirements description
Vnfm.VnfS.001	The VNFM shall support the capability to manage the expansion of the capacity of a VNF instance (see note 1).
Vnfm.VnfS.002	The VNFM shall support the capability to manage the contraction of the capacity of a VNF instance (see note 2).
Vnfm.VnfS.003	The VNFM shall support the capability to manage the scaling out/in of a VNF instance in order to perform expansion/contraction.
Vnfm.VnfS.004	The VNFM shall support the capability to expand/contract a VNF instance based on a request from the VNF instance or its Element Manager (EM) if it exists.
Vnfm.VnfS.005	The VNFM shall support the capability to expand/contract a VNF instance based on a request from NFVO.
Vnfm.VnfS.006	The VNFM should support the capability to monitor the state of a VNF instance and trigger its expansion/contraction when certain conditions are met.
NOTE 1: Expansion can either be performed by scaling out or scaling up, but only the former is required in the present release. NOTE 2: Contraction can either be performed by scaling in or scaling down, but only the former is required in the present release.	

7.2.4 Functional requirements for VNF termination

Table 7.2.4-1: Functional requirements for VNF termination

Numbering	Functional requirements description
Vnfm.VnfT.001	The VNFM shall support the capability to terminate a VNF instance.

7.3 Functional requirements for VNF configuration management

Configuration parameters referred in this clause include those set at initial configuration and any other configurable parameter declared in the VNFD.

Table 7.3-1: Functional requirements for VNF configuration management

Numbering	Functional requirements description
Vnfm.VnfCm.001	The VNFM shall support the capability to set initial configuration parameters for a VNF/VNFC instance.
Vnfm.VnfCm.002	The VNFM shall support the capability to update configuration parameters for a VNF/VNFC instance.

7.4 Functional requirements for VNF information management

7.4.1 Functional requirements for VNF Package management

Table 7.4.1-1: Functional requirements for VNF Package management

Numbering	Functional requirements description
Vnfm.VnfPkgm.001	The VNFM shall support the capability to obtain details of available VNF Packages of the VNFs which it manages.
Vnfm.VnfPkgm.002	The VNFM shall support the capability to receive notifications as a result of on-boarding of VNF Packages.
Vnfm.VnfPkgm.003	The VNFM shall support the capability to receive notifications as a result of changes on VNF Package states.

7.4.2 Functional requirements for VNF instance information management

Table 7.4.2-1: Functional requirements for VNF instance information management

Numbering	Functional requirements description
Vnfm.VnfIim.001	The VNFM shall support the capability to receive run-time data related to VNF instances that it has created (see note 1).
Vnfm.VnfIim.002	The VNFM shall support the capability to provide information on the mapping relationship between the VNF instance(s) and associated virtualised resource in response to the query.
Vnfm.VnfIim.003	The VNFM shall support the capability to modify the VNF instance information to refer to a different VNF Package (see note 2).
NOTE 1: Run-time data of VNF instance can be information from VIM related to the virtualised resource allocated to a run-time VNF instance, such as VNF instance address, record of significant VNF lifecycle event, etc. NOTE 2: A related use case is to keep NFV-MANO in sync about a VNF application software modification (see clause 5.7 of ETSI GS NFV-IFA 011 [1]).	

7.5 Functional requirements for VNF performance management

Table 7.5-1: Functional requirements for VNF performance management

Numbering	Functional requirements description
Vnfm.VnfPm.001	The VNFM shall support the capability to notify the availability of VNF performance information, resulting from virtualised resources performance information, on the VNFs it manages (see note).
NOTE: Performance information on a given VNF results from collected performance information of the virtualised resources that are mapped to this VNF instance.	

7.6 Functional requirements for VNF fault management

7.6.1 Functional requirements for virtualised resource-related VNF fault management

Table 7.6.1-1: Functional requirements for virtualised resource-related VNF fault management

Numbering	Functional requirements description
Vnfm.VrVnfFm.001	The VNFM shall support the capability to provide notifications of virtualised resource-related fault information on the VNFs it manages (see notes 1 and 2).
Vnfm.VrVnfFm.002	The VNFM shall support the capability to provide virtualised resource-related fault information on the VNFs it manages (see notes 1 and 2).
Vnfm.VrVnfFm.003	The VNFM shall support the capability to provide notifications of changes in virtualised resource-related fault information related to the VNFs it manages (see notes 1 and 2).
Vnfm.VrVnfFm.004	The VNFM shall support the capability to notify about alarms on VNF and any of its VNFCs as a consequence of state changes in the virtualised resources used by the VNF and its VNFCs.
Vnfm.VrVnfFm.005	The VNFM shall support the capability to request corrective operations on virtualised resources to VIM in order to perform VNF healing.
Vnfm.VrVnfFm.006	The VNFM shall support the capability to assign unique identifiers to the virtualised resource-related fault information on the VNFs it manages (see note 3).
Vnfm.VrVnfFm.007	The VNFM shall support the capability to keep the alarm record(s) in the alarm list unless the criteria (see note 4) is met.
NOTE 1: Virtualised resource-related fault information on a given VNF results from a collected virtualised resource fault impacting the corresponding VNF/VNFC instance. NOTE 2: Virtualised resource-related fault information on a given VNF instance can includes the information related to the alarm (e.g. alarm created, alarm cleared, etc.), alarm causes and identification of this VNF instance and fault details related to the virtualised resources assigned to this VNF/VNFC instance. NOTE 3: Two alarms that are produced by the same VNFM cannot have the same identifier. NOTE 4: The criteria to be met before an alarm record can be removed from alarm list is: the alarm acknowledgement state is "acknowledged" and the perceived severity is "cleared".	

7.6.2 Functional requirements for virtualisation-related fault management

Table 7.6.2-1: Functional requirements for virtualisation-related fault management

Numbering	Functional requirements description
Vnfm.VirFm.001	The VNFM shall support the capability to perform on-demand VNF healing on the VNF(s) it manages.
Vnfm.VirFm.002	The VNFM shall support the capability to perform automated VNF healing on the VNF(s) it manages.

7.7 Functional requirements for security consideration

Table 7.7-1: Functional requirements for security consideration

Numbering	Functional requirements description
Vnfm.Sc.001	The VNFM shall support the capability to validate that the received message is from an authenticated and authorized consumer.
Vnfm.Sc.002	The VNFM shall support the capability to verify the integrity of the received message.
Vnfm.Sc.003	The VNFM shall support the capability to encrypt the sent message or decrypt the received message using negotiated key and algorithm to or from an authenticated and authorized consumer or producer.

7.8 Functional requirements for software image management

NOTE: The software image(s) is/are at virtualisation container level, e.g. VM images.

Table 7.8-1: Functional requirements for software image management

Numbering	Functional requirements description
Vnfm.Sim.001	The VNFM shall support the capability to query the VIM for information of the software images.

7.9 Functional requirements for NFV acceleration management

Table 7.9-1: Functional requirements for NFV acceleration management

Numbering	Functional requirements description
Vnfm.NfvAm.001	When VNF-related Resource Management in direct mode is applicable, the VNFM shall support the capability to request to the VIM the allocation and release of necessary acceleration resources to meet the acceleration capability requirement(s) of the VNFs (see note).
Vnfm.NfvAm.002	When VNF-related Resource Management in indirect mode is applicable, the VNFM shall support the capability to request to the NFVO the allocation and release of necessary acceleration resources to meet the acceleration capability requirement(s) of the VNFs (see note).
Vnfm.NfvAm.003	The VNFM shall support the capability to retrieve acceleration capability requirement(s) of the VNF from the VNFD (see note).
NOTE: The acceleration capabilities can include type, capacity, NUMA support, etc.	

7.10 Functional requirements for multi-tenancy

Table 7.10-1: Functional requirements for multi-tenancy

Numbering	Functional requirements description
Vnfm.Mtm.001	When a VNFM supports multi-tenancy it shall support the capability of management of VNF tenants (see note).
Vnfm.Mtm.002	The VNFM shall support the capability to limit the scope of operations only to the service resource groups assigned to the requesting VNF tenant.
NOTE: A VNFM may be private for a tenant or it can support multi-tenancy. The management of tenants for a VNFM supporting multi-tenancy include: • create, read, update, delete tenants; • associate a tenant to a "service resource group", i.e. to a collection of VNFs; • associate a tenant to "infrastructure resource groups" managed by a VIM or to multiple "infrastructure resource groups" managed by different VIMs.	

7.11 Functional requirements for VNF indicator management

Table 7.11-1: Functional requirements for VNF indicator management

Numbering	Functional requirements description
VNFM_NFV_IND.001	The VNFM shall support the capability to receive notifications of VNF indicator value changes for the VNFs it manages (see note).
VNFM_NFV_IND.002	The VNFM shall support the capability to retrieve VNF indicator values, for the VNFs it manages, from the corresponding VNF/EM (see note).
NOTE: Indicators are information supplied by the VNF or the EM to provide some indication on the VNF behaviour. VNFM can use these indicators in conjunction with virtualised resource data to perform auto-scaling decisions.	

8 Functional requirements for VIM

8.1 General considerations

The following statement on the scope of VIM applies to all VIM related requirements:

- The VIM is responsible for controlling and managing the NFVI compute, storage and network resources of an operator's NFVI-PoP or a subset thereof (see note).

NOTE: This does not limit the possibility of VIM implementations capable of managing multiple NFVI-PoPs in any way.

8.2 Functional requirements for virtualised resource management

8.2.1 Functional requirements for virtualised resource management

Table 8.2.1-1: Functional requirements for virtualised resource management

Numbering	Functional requirements description
Vim.Vrm.001	The VIM shall support NFVI resource management within its area of responsibility (see note 1).
Vim.Vrm.002	The VIM shall support the capability of resource reservation management (see note 2).
Vim.Vrm.003	The VIM shall support the capability of quota based resource management.
Vim.Vrm.004	The VIM shall support the capability to correlate allocated and reserved virtualised resources with changes on underlying hardware/software resources due to maintenance, operation and management of the NFVI, and change the state of the allocated and reserved virtualised resources accordingly.
Vim.Vrm.005	The VIM shall support the capability to notify changes about allocated and reserved virtualised resources.
Vim.Vrm.006	The VIM shall support the capability to enforce affinity and anti-affinity policies for NFVI resource management
Vim.Vrm.007	VIM shall support the capability to receive the virtualised resource management requests from VNFM and/or NFVO, and conduct the corresponding resource management operations.
NOTE 1: NFVI resource management includes allocation, termination, update, etc. of virtualised resources. NOTE 2: The management can include the creation, update, query and termination of resource reservation(s).	

8.2.2 Functional requirements for resource reservation management

Table 8.2.2-1: Functional requirements for resource reservation management

Numbering	Functional requirements description
Vim.Rrm.001	The VIM shall support the capability to manage resources according to different resource commitment models, as follows: • Reservation model; • Quota model; • On demand.
Vim.Rrm.002	When a reservation model is used, the VIM shall support the capability to ensure that resources are allocated or updated from a resource reservation, when processing virtualised resource allocation or update requests.
Vim.Rrm.003	When a reservation model is used, the VIM shall support the capability to infer information about what reservation is applicable by using input information received with the allocation or update request.
Vim.Rrm.004	When a reservation model is used and explicit reservation identification is indicated, the VIM shall support the capability to use such information to map to the applicable resource reservation.
Vim.Rrm.005	When a reservation model is used and explicit reservation identification is not indicated, the VIM shall support the capability to map to the applicable reservation by using other information such as consumer/tenant identification (see note).
Vim.Rrm.006	The VIM shall support the capability to consider affinity/anti-affinity rules for resource reservation management.
Vim.Rrm.007	The VIM shall support the capability to notify the change regarding to virtualised resource reservation.
NOTE: Note that in this case of so-called "implicit reservation identification", the reservation identified has been reserved by the NFVO as a single bulk of resources, and successive allocations consume from that bulk.	

8.2.3 Functional requirements for virtualised resource capacity management

Table 8.2.3-1: Functional requirements for virtualised resource capacity management

Numbering	Functional requirements description
Vim.Vrcm.001	The VIM shall support the capability to collect and maintain information regarding the capacity of the NFVI it manages.
Vim.Vrcm.002	The VIM shall support the capability to provide information related to available, allocated, reserved and all virtualised resource capacity.
Vim.Vrcm.003	The VIM shall support the capability to provide the notification of the change(s) related to the capacity of the virtualised resource which are managed by it.
Vim.Vrcm.004	The VIM shall support the capability to provide information about NFVI-PoP(s) it administers, such as network connectivity endpoints and geographical location.
Vim.Vrcm.005	The VIM shall support the capability to provide information about resource zones in the NFVI that it manages.

8.2.4 Functional requirements for virtualised resource performance management

Table 8.2.4-1: Functional requirements for virtualised resource performance management

Numbering	Functional requirements description
Vim.Vrpm.001	The VIM shall support the capability to collect performance information related to virtualised resources (see note 1).
Vim.Vrpm.002	The VIM shall support the capability to notify regarding the performance information on the virtualised resources that are allocated.
Vim.Vrpm.003	The VIM shall support the capability of virtualised resource performance management in response to the request (see note 2).
NOTE 1: Virtualised resource performance information can include the virtualised resource consumption level, such as Central Processing Unit (CPU) utilization, memory usage and bandwidth consumption. NOTE 2: The performance management can include creation, update, query and deletion of PM job or thresholds.	

8.2.5 Functional requirements for virtualised resource fault management

Table 8.2.5-1: Functional requirements for virtualised resource fault management

Numbering	Functional requirements description
Vim.Vrfm.001	The VIM shall support the capability to collect fault information related to virtualised resources (see note 1).
Vim.Vrfm.002	The VIM shall support the capability to notify regarding the fault information on virtualised resources that are allocated (see note 2).
Vim.Vrfm.003	The VIM shall support the capability to notify changes in fault information on virtualised resources (see note 2).
Vim.Vrfm.004	The VIM shall support the capability to perform automated or on-demand corrective operations on virtualised resources failure.
Vim.Vrfm.005	The VIM shall support the capability to provide fault information on virtualised resources that are allocated in response to a query (see note 2).
NOTE 1: The virtualised resources fault can include virtualisation container crashes, virtual network ports errors, virtualisation containers to storage disconnection, etc. NOTE 2: The fault information related to virtualised resources can include the information related to the alarm (e.g. alarm created, alarm cleared, etc.), alarm causes and identification of the virtualised resources causing the alarm, and so on.	

8.2.6 Functional requirements for virtualised resource information management

Table 8.2.6-1: Functional requirements for virtualised resource information management

Numbering	Functional requirements description
Vim.Vrim.001	The VIM shall support the capability of providing information on virtualised resource that can be consumed within its area of responsibility (see note).
Vim.Vrim.002	The VIM shall support the capability to notify the change of information on virtualised resources that can be consumed within its area of responsibility.
NOTE: Virtualised resource Information provided by the VIM can include the description on the characteristic of the virtualised resource that can be consumed, such as virtualised resource configurations (virtual CPU configurations, types of network connectivity (e.g. L2, L3), size of virtual memory, types and size of virtualised storage resource, etc.), and/or templates (e.g. a virtual machine with 2 virtual CPUs and 2 GB of virtual memory), and so on.	

8.2.7 Functional requirements for virtualised resource configuration management

Table 8.2.7-1: Functional requirements for virtualised resource configuration management

Numbering	Functional requirements description
Vim.Vrcm.001	The VIM shall support the capability of configuration management of an individual virtualised resource using specific deployment configuration information received (see note).
Vim.Vrcm.002	The VIM should support the capability of configuration management of a set of related virtualised resources using specific deployment configuration information received (see note).
NOTE: The deployment of specific configuration information can include: Internet Protocol (IP) address types and range, subnet, ports, other guest Operating System (OS) configuration, so on.	

8.2.8 Functional requirements for NFP management

Table 8.2.8-1: Functional requirements for NFP management

Numbering	Functional requirements description
Vim.Nfpm.001	The VIM shall support the capability of management of NFPs, including creating, updating, and deleting a NFP.
Vim.Nfpm.002	The VIM shall support the capability to provide fault notification about the virtualised resources (e.g. CP, virtual network) associated with a specific NFP instance (see note).
Vim.Nfpm.003	The VIM shall validate that the classification and selection rule update does not impact the running classification and selection rules applied to the NFP instance.
NOTE: For example, when a CP instance of a NFP instance is failed, VIM notifies NFVO, and then NFVO disables a NFP or updates the rules applied to the NFP instances.	

8.2.9 Functional requirements for quota management

Table 8.2.9-1: Functional requirements for quota management

Numbering	Functional requirements description
Vim.Qm.001	The VIM shall support the capability to reject virtualised resource allocation requests causing a quota to be exceeded.
Vim.Qm.002	The VIM shall support the capability to create resource quota for the consumer of the virtualised resources (e.g. Tenant).
Vim.Qm.003	The VIM shall support the capability to update the resource quota for the consumer of the virtualised resources (e.g. Tenant) of the virtualised resource.
Vim.Qm.004	The VIM shall support the capability to delete the resource quota for the consumer of the virtualised resources (e.g. Tenant).
Vim.Qm.005	The VIM shall support the capability to provide information on the resource quota for the consumer of the virtualised resources.
Vim.Qm.006	The VIM shall support the capability to notify the changes of the information on the resource quota for the consumer of the virtualised resources.

8.3 Functional requirements for infrastructure resource management

8.3.1 Functional requirements for infrastructure resource performance management

Table 8.3.1-1: Functional requirements for infrastructure resource performance management

Numbering	Functional requirements description
Vim.Irpm.001	The VIM shall support the capability of collection of performance information related to software and hardware resources within the NFVI (see notes 1 and 2).
NOTE 1: Hardware resources within the NFVI refer to physical compute, storage, and networking resources. Software resources refer to software components within the NFVI (e.g. a hypervisor) but do not refer to the VNF's software.	
NOTE 2: Performance information related to software and hardware resource within the NFVI can include software and hardware resource consumption level, such as physical memory consumption, CPU power consumption, Peripheral Component Interface express (PCIe) bandwidth consumption.	

8.3.2 Functional requirements for infrastructure resource fault management

Table 8.3.2-1: Functional requirements for infrastructure resource fault management

Numbering	Functional requirements description
Vim.Irfm.001	The VIM shall support the capability to correlate fault information on virtualised resources with fault information related to underlying used software and hardware resources within the NFVI (see note 1).
Vim.Irfm.002	The VIM shall support the capability of collection of fault information related to software and hardware resources within the NFVI (see note 2).
NOTE 1: Hardware resources within the NFVI refer to physical compute, storage, and networking resources. Software resources refer to software components within the NFVI (e.g. a hypervisor) but do not refer to the VNF's software. NOTE 2: The software and hardware resources fault can include suspension of the underlying OS, physical network disconnection due to a Network Interface Controller (NIC) failure, etc.	

8.4 Functional requirements for security consideration

Table 8.4-1: Functional requirements for security consideration

Numbering	Functional requirements description
Vim.Sc.001	The VIM shall support the capability to validate that the received message is from an authenticated and authorized consumer.
Vim.Sc.002	The VIM shall support the capability to verify the integrity of the received message.
Vim.Sc.003	The VIM shall support the capability to encrypt the sent message or decrypt the received message using negotiated key and algorithm to or from an authenticated and authorized consumer or producer.

8.5 Functional requirements for software image management

NOTE: The software image(s) is/are at virtualisation container level, e.g. VM images.

Table 8.5-1: Functional requirements for software image management

Numbering	Functional requirements description
Vim.Sim.001	The VIM shall support the capability of management of software images as requested.
Vim.Sim.002	The VIM shall support the capability to verify the integrity of the software images.
Vim.Sim.003	The VIM should support the capability to manage multiple versions of software images.
Vim.Sim.004	The VIM shall support the capability to provide the information on the software images which it manages.

8.6 Functional requirements for NFV acceleration management

Table 8.6-1: Functional requirements for NFV acceleration management

Numbering	Functional requirements description
Vim.NfvAm.001	The VIM shall support the management of the NFV acceleration resources (see note 1).
Vim.NfvAm.002	The VIM shall support the capability to retrieve feature related information provided by the NFV acceleration resources.
Vim.NfvAm.003	The VIM shall support the capability to provide acceleration capability information to NFVO (see note 2).
Vim.NfvAm.004	The VIM shall support the capability to translate the acceleration capability requirement (e.g. bandwidth value) into acceleration resource context (e.g. number of Field Programmable Gate Array (FPGA) blocks).
NOTE 1: Acceleration resource management in VIM includes discovery, allocation, release, reprogram, etc. of acceleration resources in NFVI. NOTE 2: The information can include type, capacity, NUMA support, etc.	

8.7 Functional requirements for multi-tenancy

Table 8.7-1: Functional requirements for multi-tenancy

Numbering	Functional requirements description
Vim.Mtm.001	The VIM shall support the capability of management of infrastructure tenants (see note 1).
Vim.Mtm.002	The VIM shall support the capability to identify software images assigned to an infrastructure tenant and software images shared among infrastructure tenants.
Vim.Mtm.003	The VIM shall support the capability to allow an infrastructure tenant to instantiate virtual resources using its own private software images or shared software images.
Vim.Mtm.004	The VIM shall support the capability to limit the scope of operations only to the infrastructure resource groups assigned to the requesting infrastructure tenant.
NOTE 1: The management of tenants include: • create, read, update, delete tenants; • associate a tenant to one or more "infrastructure resource groups" managed by a VIM. NOTE 2: A software image which is assigned to a single tenant is commonly referred to as a private software image of this tenant. A software image which is assigned to all tenants is commonly referred to as a public software image. A software image which is assigned to more than one tenant is commonly referred to as a shared software image.	

9 Architectural level Requirements

9.1 General guidelines for NFV management and orchestration interface design

This clause defines general interface guidelines applicable to all NFV-MANO interfaces.

These guidelines are applicable for interface specifications.

Table 9.1-1: General guidelines for NFV management and orchestration interface design

Numbering	Guideline description
Inf.NfvMoidG.001	The interface should be self-contained enabling easy implementation and maintenance (see note).
Inf.NfvMoidG.002	The interfaces should be based on standardized specification, which does not allow room for interpretation.
NOTE: Self-contained implies that the specification should not refer or depend on the specifications of another one.	

9.2 General requirements to NFV management and orchestration interface design

This clause defines general interface requirements applicable to all NFV MANO interfaces.

NOTE: The requirements for individual interfaces will not be covered in this clause.

These requirements are applicable for interface specifications.

Table 9.2-1: General requirements to NFV management and orchestration interface design

Numbering	Requirements description
Inf.NfvMoid.001	The interface shall provide an extension mechanism.
Inf.NfvMoid.002	The interface extension mechanism should support the addition of private extensions.
Inf.NfvMoid.003	The interface specification shall identify for each information element and attribute whether is mandatory or optional in the context where it is used (see note 4).
Inf.NfvMoid.004	The interface specification shall contain the complete specification of all mandatory information elements necessary for interoperability at the interface.
Inf.NfvMoid.005	Entity names (see note 5) shall be unique across all entity types and all reference points in a given naming domain (see note 1).
Inf.NfvMoid.006	Entity names (see note 5) shall not embed any information beyond the name itself (see note 2).
Inf.NfvMoid.007	An entity (see note 5) shall have the same name across all reference points that it appears.
Inf.NfvMoid.008	A common filtering description shall be used across all NFV interface operations having a filter input parameter (see note 3).
NOTE 1: The extent of a naming domain is a deployment decision which can potentially cover multiple instances of the various NFV reference architecture FBs. NOTE 2: For example, it is not recommended to embed location or containment hierarchy in an entity names (such information should be kept in separate attributes). NOTE 3: Only a subset of the filtering capability might be needed for a given operation. Typical filtering might be entity list or type matching, template matching or attribute value matching. NOTE 4: A context is either a set of input/output information elements for an operation or a set of attributes within a structured information element. NOTE 5: Depending on the actual communication solution, an entity may take different forms (e.g. a parameter in a message, a field in a URI, etc.). Consequently its name may take different forms as well (e.g. a field or parameter tag).	

9.3 General requirements for NFV management and orchestration services

Table 9.3-1: General requirements for NFV management and orchestration services

Numbering	Guideline description
Mano.NfvMos.001	The NFV-MANO shall enable the discovery and retrieval of information regarding management and orchestration related interfaces, including all information necessary for their usage (e.g. interface endpoint address).

9.4 General requirements for multi-tenancy

Table 9.4-1: General requirements for multi-tenancy

Numbering	Functional requirements description
Nfv.Mtm.001	A consumer of an interface which supports multi-tenancy shall provide the identification of an appropriate tenant (infrastructure tenant, VNF tenant or NS tenant) when performing an operation.

Annex A (informative): Resource management additional information

A.1 Quota based resource management

A.1.1 Overview

To ensure appropriate allocation of NFVI resources, resource quotas can be used in the VIM. These quotas can be used to constrain the NFVI resources which a consumer of these resources can obtain. A consumer identifier will be included in all resource requests to the VIM where quota based resource management is supported. The entities which the consumer identifier maps to are up to service provider configuration. A request for resources beyond a quota limit will be rejected by the VIM.

To ensure that the NFVO has visibility of actual resource utilization in the NFVI, resource consumption and availability information can be exchanged between the VIM and NFVO via processes of event notification, periodic update and query.

A.1.2 Summary of key aspects

Key aspects of the quota based resource management approach are:

- A consumer quota is associated with a consumer identifier.
- Service providers determine the appropriate level of resource quotas associated with consumer identifiers, and the mapping of consumer identifiers to entities.
- A consumer quota for NFVI resources is set in the VIM via interaction with the NFVO or via an alternative configuration mechanism.
- The VNFM may be informed of the resource quotas at the VIM which is imposed on it or the VNFs which it manages.
- The VNFM takes direction from the NFVO before taking any action relating to the instantiation and scaling of VNFs.
- A VIM that supports quota based resource management will validate that requests for resources are within the quota of the consumer identifier provided in the request prior to allocation.
- If a quota associated with a consumer identifier is exceeded the VIM will reject the request.

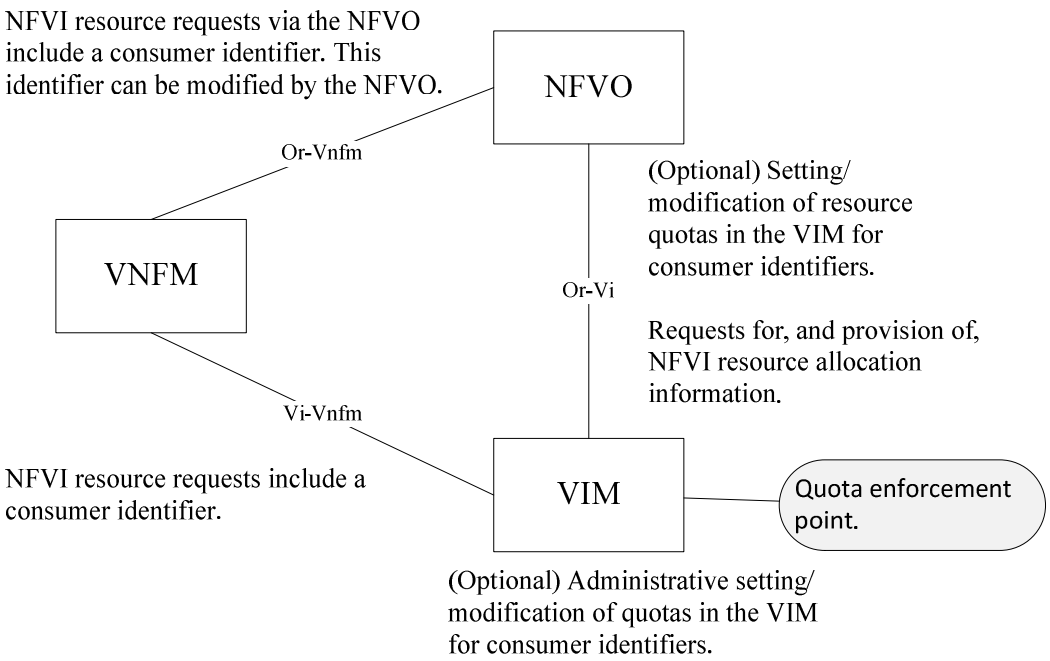

Figure A.1.2-1: Architectural outline of resource quotas

A.1.3 Allocation of consumer identifiers

Consumer identifiers will be assigned via local configuration or via instruction from the NFVO. The entities which the consumer identifier is associated with are determined by the service provider.

A.1.4 Setting of quotas

To avoid unexpected or inappropriate use of NFVI resources, defined quotas (limits) for consumers can be set in the VIM regarding the type and quantity of resources which can be requested from the VIM. The quota information which associates consumer identifiers with specific quotas is communicated to the VIM over the Or-Vi reference point or by some other configuration process. Quota can be modified after being set.

A.1.5 NFVO awareness of NFVI resource consumption

To enable the NFVO to intelligently manage resources, the NFVO can obtain information from the VIM regarding NFVI resource allocations and outstanding resource reservations. It can do this via notification of NFVI resource consumption change events, resource information change notifications from the VIM or a periodic resource information query to the VIM.

A.1.6 NFVI resource acquisition

A VNFM with granted permission for the instantiation or scaling of a VNF can send a resource request to the VIM containing a consumer identifier. If the resources are available in the NFVI, and the quota associated with consumer identifier is not exceeded, then the requested resources will be allocated. If allocation of the requested resources would breach the quota for the consumer identifier, then the request will be rejected. Additionally, a notification can be sent to the NFVO informing it of the action taken by the VIM.

The NFVO can use the notification of this event to determine a subsequent action to: free up NFVI resources; seek access to alternative NFVI resources; or take whatever action was felt to be appropriate.

A.1.7 Resource contention mitigation

The NFVO is expected to have the ability to monitor resource allocation in the NFVI via the VIM. Hence it is anticipated that any decision it takes which would require consumption of additional NFVI resources would take into account its understanding of resource availability in the NFVI. If the NFVO was aware of resource limitations in the NFVI, and hence that there was a probability of insufficient resources to complete a VNF lifecycle management task, then the NFVO might not grant this task and take alternative action instead.

A.1.8 Data centre resource utilization efficiency

Resource management without reservation maximizes the availability of NFVI resources by ensuring that resources are only removed from the pool of available resources when in active use.

A.1.9 Resource management evolution and interoperability

The resource quota enforcement approach could be commercially deployed in phases. For example, an initial deployment can involve very simple consumer resource limitations quotas administratively configured in the VIM. The deployed solution could then be enhanced over time as each entity became more capable. Further enhancement could be provided via a mechanism to enable reservation of NFVI resources from the NFVO. This capability might be used to assure resource availability for critical VNFs or where it was felt necessary in a data centre environment shared by different commercial entities.

A.1.10 Co-existence of resource quota enforcement and resource management with reservation

It is anticipated that the reservation of NFVI resources from the NFVO to the VIM would render the requested resources unavailable until they were released. Hence a resource request without a reservation and using the quota based resource management would have resources allocated to it from a pool of free resources not under active reservation. Additionally, local rules will determine the behaviour in the VIM if a reservation is received which is in excess of an applicable consumer quota.

A.2 Management of resource reservations

A.2.1 Introduction

Reservation enables securing resources to guarantee their availability without allocating them, i.e. resources are committed to a particular consumer or consumer type, but not necessarily all of them are allocated/instantiated yet.

Various use cases for reservation are introduced and the key aspects of reservation presented.

A.2.2 Use cases

A.2.2.1 Use case for securing resources for several tenants

The NFV-MANO framework enable tenants to request and make use of virtualised resources provided by the platform. VIM manages the NFVI and offers to consumers (tenants) operations for managing virtualised resources. In NFV deployments, several tenants can coexist, and in this scenario resource management race conditions can happen, ending in resource service denegation. In carrier telco environments, with stringent SLAs, reliability and performance requirements, resource service denegation can become an issue.

The NFVO plays a key role in the NFV-MANO, as central point for orchestrating the resource consumption by VNFs and NSs and granting the lifecycle operations. The NFVO cannot guarantee resource availability during the granting of a VNF lifecycle request if the resources needed to accommodate such lifecycle operation have not been secured (i.e. reserved) by the VIM, entity responsible for the NFVI resources management.

A.2.2.2 Use case for securing resources with detailed capabilities

The VIM, as end point for managing and controlling the NFVI resource holds more detailed information about the managed resources and their availability. At the NFVO, visibility of specific resources is not the same as the VIM. The NFVO holds information about the availability, reserved and allocated NFVI resources as abstracted by the VIM.

Examples of more detailed information are specific acceleration capabilities, CPU-pinning, etc. This information is visible at the VIM level in order to execute the right allocation of virtualised resources according to the resource capability requirements. If such capabilities are needed, and the NFVO has no visibility on the particular resources accommodating such capabilities, granting the VNF lifecycle operations can lead to undesired resource service denegation, in particular those that follow with subsequent virtualised resource management requests for detailed capabilities.

A.2.2.3 Use case for securing resources during NS instantiation

A NS can be composed of a number of VNFs, VLs to interconnect them, etc. In order to realize a NS, it is possible that a great quantity of NFVI resources will be needed. Thus, the instantiation of an NS will be possible as long as all the resources can be secured to be available at the time of the instantiation of the NS.

The instantiation of an NS can involve several transactions, with possibly a number of different VIMs managing the required NFVI resources, and VNFMs managing the lifecycle of the VNFs to instantiate. During the instantiation process, if resources cannot be secured to be available by the VIM(s) for the NS, the overall instantiation can fail. This can lead to inefficient processing and arrangement of NS instantiation.

A.2.2.4 Use case for securing resources during NS scaling

A NS can be composed of a number of VNFs, VLs to interconnect them, etc. In order to realize an NS, as well as for scaling purposes, it is possible that a great quantity of NFVI resources will be needed. Moreover, under certain scenarios, such as sport events or natural disasters, operators require that NSs can scale to accommodate the extra traffic to handle. Such NS scaling requires adding extra resources to be used by the VNFs part of the NS, or new ones to be instantiated. By reserving resources in advance against the VIM managing the resources, it is ensured that NS can scale properly.

A.2.2.5 Use case for securing resources related to a scheduled event

NSs or certain capacity may only be needed for a specified duration. For instance, the duration of a scheduled sport event is usually known in advance, i.e. with an expectation to be ended at some point in time.

To support the event, the operator may need to add extra NS capacity or instantiate a new NS. In this scenario, the service provider wishes to secure the instantiation of new VNF instances, or the expansion of existing instances for the NS by reserving underlying NFVI resources.

The present use case exemplifies the need for the NFVO and VIM to handle reservation time information.

As part of the NS instantiation/expansion, the NFVO requests to the appropriate VIM(s) the reservation of virtualised resources needed by the VNF instances. In addition, the NFVO provides information about the expected timespan where the virtualised resources will be used, i.e. it provides start and end time information. The time information may either be the same or have certain deviation from the scheduled event timing to allow for certain backup time. This information about start and end time helps the VIM to determine the best scheduling of resources and their availability in the NFVI-PoP(s). This is particularly applicable when scheduling resources for multiple future events, i.e. the VIM will know about reservations that have been scheduled but whose reserved resources are not being used yet or reservations that have been scheduled, but whose reserved resources will be freed prior to another reservation.

A.2.3 Summary of key aspects

Key aspects of the resource reservation are:

- NFVO decides if and when a resource reservation is needed.

- Resource reservation can be done:
  - before a VNF LCM operation as part of a NS LCM operation;
  - as part of granting procedure for a VNF LCM operation; and
  - during configuration/reconfiguration of resources in the NFVI-PoP(s).
- NFVO requests the reservation of the needed resources to the VIM.
- Reservations are identifiable. A reservation identifier establishes the identity of the arrangement for securing the future usage of resources by a consumer.
- When resource reservation is performed as indicated by policies, the reservation identifier is directly used by NFVO as part of managing the resource reservation. The identifier is provided to the VNFM, either as part of a VNF LCM operation request or in response to a granting request:
  - VNFM uses the reservation identifiers for requests related to the resources that are needed for the instantiation and lifecycle of VNF.

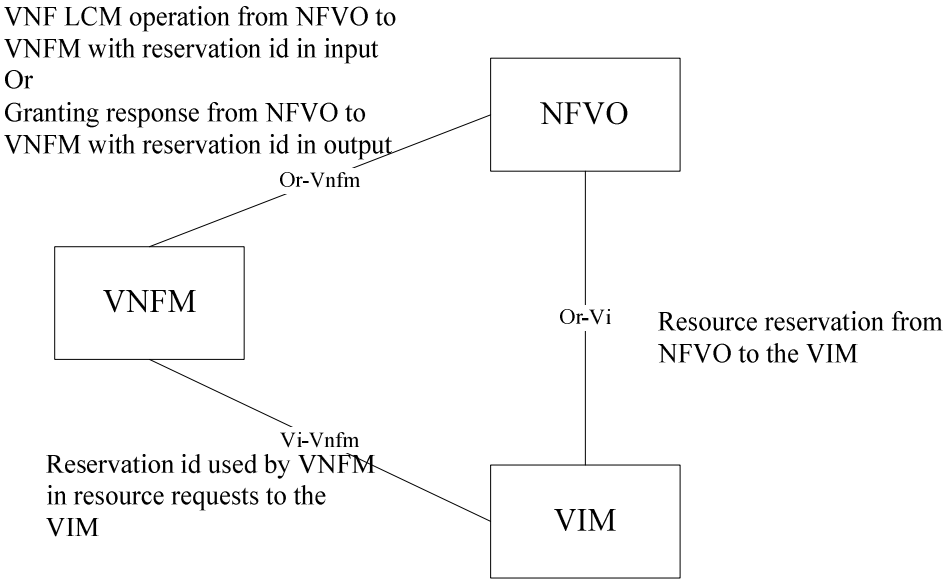

Figure A.2.3-1: Architectural outline of reservation

A.2.4 Resource reservation management by NFVO

Resource reservations are triggered by NFVO by calling the corresponding VIM to reserve the resources. It is anticipated that the reservation of NFVI resources from the NFVO to the VIM would render the requested resources unavailable until they were released.

In case of NS LCM operation where reservation is needed, NFVO will reserve the resources needed for each VNF LCM operation for all the impacted VNFs in the NS. Once the reservations are successfully secured, the NFVO will issue corresponding reservation identifier(s) to the VNFM.

In case of failure of one of the LCM operations, the NFVO will cancel any pending reservations associated with the LCM request.

In case of VNF LCM operation, not coming from a NS LCM operation, if reservation is needed, the NFVO will reserve the needed resources as part of the granting request. The corresponding reservation identifier(s) will be returned as part of the grant response.

A.2.5 Resource reservation handling by the VNFM

A VNFM can receive, either in part of the input parameters of a VNF LCM operation or in the response of a grant request, one or more than one reservation identifier.

A reservation identifier indicates that a reservation has been performed for this VNF. The VNFM makes use of this reservation identifier(s) in the subsequent resource requests for this VNF made to the VIM.

A.2.6 Resource reservation contention mitigation

The VIM handles the resource reservation contention mitigation as the VIM is responsible for the control of whether virtualised resources can be reserved or not based on the detailed internal capacity information that it maintains.

The VIM is expected to have the ability to monitor the availability of resources in the NFVI and how virtualised resources can be accommodated in the NFVI. To mitigate reservation contention, it is also expected the VIM will ensure that NFVI resources are reserved efficiently. For instance, performing by the VIM a uniform reservation in the physical NFVI resources may lead to a situation where certain virtualised resources demanding large amount of resources cannot be allocated when needed.

EXAMPLE: Consider 2 physical NFVI resource nodes (Node-1 and Node-2) with 4 capacity units that can be reserved. A first reservation requests for 2 affine capacity units (i.e. on the same node) is processed by the VIM, and these 2 capacity units are reserved from Node-1. A second reservation request for 2 affine capacity units is also processed by the VIM, and using a uniform reservation policy these 2 capacity units are reserved from Node-2. A third reservation request for 3 affine capacity units cannot be successfully processed as there are not enough free capacity units neither from Node-1 nor from Node-2.

It is also possible for the NFVO to perform actions to mitigate resource reservation contention by monitoring the capacity usage of resources from the NFVI-PoP(s), as reported by the corresponding VIM(s). For instance, requesting resource reservation on a highly loaded NFVI-PoP can increase the chances of rejection of the resource reservation.

A.2.7 Co-existence of reservation with quota

The quota mechanism is used to constrain the NFVI resources that a consumer of these resources can obtain. If applicable, the VIM will also apply the quota to the reservation being made. Local rules will determine the behaviour in the VIM if a reservation is received which is in excess of an applicable consumer quota.

A.2.8 Resource reservation types

Resource reservation can be performed at different levels, namely:

1) For virtualised containers, virtual networks, network ports and/or storage volumes; or/and

2) For virtualised resource capacity (on compute, storage, and network resource types).

The first case considers the reservation of virtualised containers (e.g. VMs) based on defined container configurations, e.g. it supports the reservation based on certain VM flavours that determine the number and disposition of vCPUs, virtual memory, virtual storage and number of virtual network interfaces. Reservation for defined virtual networks, network ports and storage volumes is also part of this category.

The second case considers the reservation of resource capacity without a specific virtualised container disposition. For example, a resource reservation in this format may indicate the total required capacity in terms of number of vCPUs and virtual memory. Reservation of total capacity for virtual storage, or number of public IP addresses is also part of this category.

A.3 Management of permitted allowance

A.3.1 Introduction

To ensure consumption of resources stays within the limits defined by service providers, permitted allowance can be used at NFVO level to control resource consumption by VNFMs in relation to some granularity associated with the permitted allowance. The granularity might vary (VNFM, VNF, group of VNFs, NS, etc.). Permitted allowance is maintained by the NFVO.

All VNF LCM request that imply potential resource changes, i.e. instantiation, scaling in/out, update, terminate, upgrade and healing of VNF instances are using the grant operation and as part of the processing of the grant operation, the permitted allowance is checked and the current level maintained.

A.3.2 Summary of key aspects

Key aspects of the management of permitted allowance are:

- Service providers determine the appropriate level of resource for the permitted allowance and the corresponding granularity.
- Permitted allowances are provided to the NFVO.
- NFVO supports the permitted allowance by checking the matching one during the processing of the grant request.
- NFVO maintains the current level of the permitted allowance based on the granted requests.

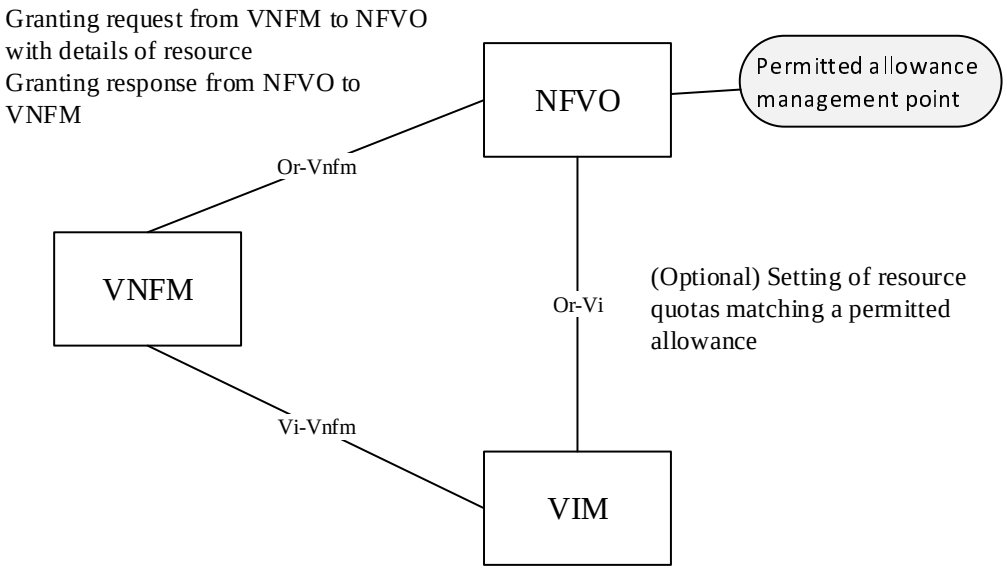

Figure A.3.2-1: Architectural outline of permitted allowance

A.3.3 Setting of permitted allowance

To ensure consumption of resources stays within the limits defined by service providers, the operator or the OSS can define permitted allowance regarding the type and quantity of resource associated with a given granularity. This permitted allowance might be applicable across multiple VIMs.

This permitted allowance information can be communicated to the NFVO over the Os-Ma-nfvo reference point or configured by some other process.

A.3.4 Permitted allowance management by NFVO

The permitted allowance are managed by NFVO, as a maximum and current level of resources. The maximum level corresponds to the definition of the permitted allowance and the current level is what is being marked as consumed as a result of the grant requests.

When receiving a grant request from a VNFM, as part of the processing of the grant, the NFVO matches the request to the permitted allowance with corresponding granularity.

If the request is asking for resources, i.e. instantiate, scale out, etc., NFVO checks if adding the desired resources provided as part of the grant request to the current level of resources still maintains the current level below the maximum level. If so, the request stays within the permitted allowance.

If the request is freeing resources, i.e. terminate, scale in, the NFVO subtracts the provided resources from the current level, making it lower.

In case the VNF LCM operation fails at VNFM, resources might be marked as used (not used) in the permitted allowance while not used (used) in reality. NFVO would need to check that the resources are affectively used (not used), for instance by checking for correct lifecycle instantiation /scale/termination events of a VNF to avoid this problem.

A.3.5 Permitted allowance awareness by the VNFM

A VNFM when processing a VNF LCM request that imply potential resource changes, i.e. instantiation, scaling in/out, update, terminate, upgrade and healing of VNF instances issues a grant request to the NFVO with the details of the operation, the VNF and the resource change (resource needed or resource released).

One of the actions of the processing of the grant request is to validate the request against matching permitted allowance. The VNFM is not aware of the details of the permitted allowance used by the NFVO for the grant operation.

If the response from the grant is successful, the VNFM can issue resource requests.

A.3.6 Permitted allowance contention mitigation

The NFVO is managing permitted allowance and when a permitted allowance reaches its limit, NFVO should issue a notification and should reject granting requests asking for more resources and matched to this permitted allowance.

The OSS or the operator are expected to have the ability to monitor these notifications and might react by extending the permitted allowance that reached its limit.

A.3.7 Co-existence of permitted allowance and resource quota enforcement

If the definition of a permitted allowance is compatible with the definition of quota, i.e. applicable to a single VIM, using the resource granularity supported by quotas, the NFVO might choose to enforce a permitted allowance by defining in the VIM a quota that correspond to a given allowance using a specific tenant.

In this case, the tenant associated with the quota would be communicated to the VNFM in the grant response and the VNFM will use it for all resource allocation requests associated to the granted VNF LCM request.

A.3.8 Co-existence of permitted allowance and resource management with reservation

The permitted allowance is managed at NFVO level while the reservation is made at VIM level. So they can both co-exist without impact.

As well as actual resource consumption, resources reserved can count towards permitted allowance. The handling of permitted allowance for reserved resources is similar to normal resources as described in clause A.3.4.

Annex B (informative): Virtualised resources capacity management

B.1 Introduction

Virtualised resources capacity management encompasses functionalities to gather information about virtualised resource capacity usage. Both the VIM and NFVO perform functionality related to virtualised resources capacity.

B.2 Virtualised resources capacity information management by the VIM

B.2.1 Functionality

The VIM executes the following functionality as baseline to support virtualised resources capacity information management:

- It manages inventory related information of NFVI hardware resources (compute, storage, network) and software resources (e.g. hypervisors), including the discovery of capabilities of such resources.
- It keeps information about reservation and usage of virtualised resources identifying the association of the virtualised resources to the physical compute, storage and network resources.

NOTE: The particular allocation, update, migration, scaling, operation and termination of virtualised resources are virtualised resource management functions.

The VIM executes the following functionality to actually perform virtualised resources capacity information management:

- It manages information about virtualised resources capacity per NFVI-PoP and resource zone, detailing total, available, allocated and reserved virtualised resource capacity per resource type.
- It provides information about virtualised resources capacity and notifies changes about the virtualised resources capacity.
- It provides information about NFVI-PoP(s) it administers, such as network connectivity endpoints and geographical location.

B.3 Virtualised resources capacity management by the NFVO

B.3.1 Functionality

The NFVO performs the following functionality related to virtualised resources capacity information management:

- It retrieves and processes notifications from VIM instances with information about NFVI-PoP virtualised resources capacity usage at different granularities and levels as provided by the VIM, including total per NFVI-PoP and per resource zone.
- It retrieves information from VNFM instances about virtualised resources usage and mapping with instantiated VNFs.

- It retrieves information about the connectivity to and in-between NFVI-PoPs and Network Point of Presences (N-PoPs) and builds network topology map information.
- It keeps information about retrieved virtualised resources capacity and synchronizes such information on-demand or periodically with VIMs, WAN Infrastructure Managers (WIMs) in order to keep the information updated.
- It keeps information about retrieved VNF's resource usage and synchronizes such information on-demand or periodically with VNFMs in order to keep the information updated.
- It aggregates the capacity information received from VIMs and WIM, and correlates such information with VNF's resource usage from VNFMs to quantify and determine the virtualised resource capacity usage mapped to VNF and NS instances throughout time.

The NFVO makes use of the virtualised resources capacity information to:

- Support analytics for virtualised capacity planning to determine best usage of NFVI resources across NFVI-PoPs.
- Generate virtualised resources capacity reports and notify about resource shortage.
- Validate NS resource usage and distribution of resource usage across operator's Infrastructure Domains.
- Validate and grant VNF lifecycle operations requested from VNFM, as those may impact the way requested resources are allocated within one NFVI-PoP or across multiple NFVI-PoPs.
- Placement optimization for the instantiation and LCM of VNFs and NSs, including:
  - Identifying and selecting the target VIM and WIM to which virtualised resources will be reserved and/or consumed for VNFs and NS.
  - Selecting the target resource zones in NFVI-PoPs to accommodate VNF instantiation according to input resource, performance and resiliency requirements.

Annex C (informative): VNF management

C.1 Introduction

This annex reports on concepts related to VNF management.

Clause C.2 introduces use cases related to VNF management.

C.2 Use cases

C.2.1 Use case for stopping a VNF instance

C.2.1.1 Introduction

The goal of the use case is to enable stopping a running VNF instance without releasing the virtualised resources that have been instantiated to such VNF instance. As part of this process, the guest operating system (OS) of the VNF instance may be shutdown. The VNFM is responsible for executing the procedure.

Stopping a VNF instance allows fast re-activation of a VNF without having to re-instantiate the virtualised resources. Together with starting a VNF instance, it provides a means to reboot a VNF instance, e.g. to be used to reactivate a VNF whose application was faulty and there were no other means to recover from the fault.

Both EM and NFVO need to be able to request stopping a VNF instance. For instance, the EM as manager of the application from OSS/BSS perspective is involved in the procedures related to commissioning and decommissioning of the VNF into service and failure correction. The NFVO needs to also be able to trigger the operation, e.g. as part of NS lifecycle and fault management procedures.

C.2.1.2 Steps

Actors:

- NFV-MANO (VIM, NFVO and VNFM).
- VNF instance.

Pre-Conditions:

- The VNF instance is instantiated and running.
- NFV-MANO (VIM, NFVO and VNFM) is running.

Steps:

1) The VNFM receives a request from the NFVO or the EM to stop the VNF instance.

2) The VNFM sends VNF lifecycle change notification to consumers (NFVO and/or EM) about the start of the stopping procedure.

3) The VNFM knows the shutdown order between VNFC instances of the VNF (e.g. in accordance with workflow(s) in VNFD) and sends command to VIM to shut down the associated virtualised containers (e.g. VMs).

NOTE: If the workflow requires a graceful stop, as part of this process the VNFM will interact with VNF/EM to gracefully stop the application.

4) VIM processes the request and signals to the hypervisor in the NFVI to shut down the virtualised container(s).

5) VIM returns confirmation of shutting down the virtualised container(s) to the VNFM.

6) VNFM sends notification with the result of the operation to consumers (NFVO and/or EM).

Post-Conditions:

- The VNF instance is stopped.

C.2.2 Use case for starting a VNF instance

C.2.2.1 Introduction

The goal of the use case is to enable starting a VNF instance that was previously in the state "stopped" without having to modify the virtualised resources that were previously instantiated. As part of this process, the guest OS of the VNF instance may be booted if it has been shut down. The VNFM is responsible for executing the procedure.

Starting a VNF instance allows fast re-activation of a VNF without having to re-instantiate the virtualised resources. Together with stopping a VNF instance, it provides a means to reboot a VNF instance, e.g. to be used to reactivate a VNF whose application was faulty and there were no other means to recover from the fault.

Both EM and NFVO need to be able to request starting a VNF instance. For instance, the EM as manager of the application from OSS/BSS perspective is involved in the procedures related to commissioning and decommissioning of the VNF into service and failure correction. The NFVO needs to also be able to trigger the operation, e.g. as part of NS lifecycle and fault management procedures.

C.2.2.2 Steps

Actors:

- NFV-MANO (VIM, NFVO and VNFM).
- VNF instance.

Pre-Conditions:

- The VNF instance is instantiated and stopped.
- NFV-MANO (VIM, NFVO and VNFM) is running.

Steps:

1) The VNFM receives a request from the NFVO or EM to start the VNF instance.

2) The VNFM sends VNF lifecycle change notification to consumers about the start of the starting procedure.

3) The VNFM knows the boot-up order between VNFC instances of the VNF (e.g. in accordance with workflow(s) in VNFD) and sends command to VIM to boot up the associated virtualised containers (e.g. VMs).

4) VIM processes the request and signals to the hypervisor in the NFVI to boot up the virtualised container(s).

5) VIM returns confirmation of booting the virtualised container(s) to the VNFM.

6) VNFM sends notification with the result of the operation to consumers (NFVO and/or EM).

Post-Conditions:

- The VNF instance is started.

Annex D (informative): Authors & contributors

The following people have contributed to the present document:

Rapporteur:
Amanda Xiang, Huawei Technologies Inc.

Other contributors:
Aijuan Feng, Huawei
Andy Bennett, Cisco
Anatoly Andrianov, Nokia Networks
Anh Le, NetCracker
Anton Korchak, NetCracker
Arturo Martin de Nicolas, Ericsson
Ashiq Khan, DOCOMO Communications Lab
Astrid Mann, Huawei
Bertrand Souville, DOCOMO Communications Lab
Bruno Chatras, ORANGE
Deepanshu Gautam, Huawei
Dmytro Gassanov, NetCracker
Elena Demaria, TELECOM ITALIA S.p.A.
Fang Yu, Huawei
Gerald Kunzmann, DOCOMO Communications Lab
Giuseppe Monteleone, Italtel SpA
Gongying Gao, China Unicom
Gyula Bodog, Nokia Networks
Gergely Csatari, Nokia Networks
Hai Liu, Huawei
Itzik Kitroser, Amdocs
Janusz Pieczerak, ORANGE
Jeremy Fuller, GENBAND
Jianning Liu, Huawei
Jie Miao, China Unicom
Joan Triay, DOCOMO Communications Lab
Jong-Hwa Yi, ETRI
Jun Peng, Huawei
Junsheng Chu, ZTE
Kai Zhang, Huawei
Kazuaki Obana, DOCOMO Communications Lab
Linghui Zeng, Huawei
Marc Flauw, Hewlett-Packard Enterprise
Michael Brenner, Alcatel-lucent
Michael Klotz, Deutsche Telekom AG
Myung-Ki Shin, ETRI
Nicola Santinelli, TELECOM ITALIA S.p.A.
Olivier Le Grand, ORANGE
Peng Zhao, China Mobile
Peter Wörndle, Ericsson
Serge Manning, SPRINT
Seung-lk Lee, ETRI
Shumin Cheng, Huawei
Stefan Arntzen, Huawei
Stephen Fratini, Ericsson
Susana Sabater, Vodafone
Thinh Nguyenphu, Nokia Networks
Tommy Lindgren, Ericsson
Uwe Rauschenbach, Nokia Networks
Xiaojuan Hu, China Telecommunications
Xiaoxiang He, Huawei

Xu Yang, Huawei
Yan Zhou, Huawei
Yiqiang Hua, China Unicom
Yunpeng Xie, China Telecommunications
Zarrar Yousaf, NEC EUROPE LTD
Zhipeng Huang, Huawei

History

Document history		
V2.1.1	April 2016	Publication
V2.2.1	September 2016	Publication
V2.3.1	August 2017	Publication