ETSI GS NFV-SEC 026 V5.1.1 (2025-07)

GROUP SPECIFICATION

Network Functions Virtualisation (NFV) Release 5; Security; Isolation and trust domain 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
DGS/NFV-SEC026

Keywords
access control, cybersecurity, NFV, NFVI,
security, trust

ETSI

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

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

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

Important notice

The present document can be downloaded from the
ETSI Search & Browse Standards application.

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

Users should be aware that the present document may be revised or have its status changed,
this information is available in the Milestones listing.

If you find errors in the present document, please send your comments to
the relevant service listed under Committee Support Staff.

If you find a security vulnerability in the present document, please report it through our
Coordinated Vulnerability Disclosure (CVD) program.

Notice of disclaimer & limitation of liability

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

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

Copyright Notification

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

© ETSI 2025.
All rights reserved.

Contents

Intellectual Property Rights

Essential patents

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

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

Trademarks

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

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

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

Modal verbs terminology

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

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

1 Scope

The present document defines the requirements and solutions for the NFV System to enhance network functions and services isolation between tenants.

For this aim, the present document includes in particular:

- Analysis of the threat models.
- Trust domain separation (multi-tenant NFVI, traffic and resource separation, tenant-dependant resource management and access control).
- Memory protection and access control (protection against memory introspection, confidentiality of sensitive data and credentials).
- Hypervisor trust partitioning.
- The Virtualization Container (e.g. Virtual Machine and OS container) Escape protection (e.g. protection against VNF compromising its local host OS, taking control of the hypervisor and then gaining access to private and sensitive data of co-resident Virtualization Containers).
- Associated key management system for all above items.

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 in the ETSI docbox.

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

[2] IETF RFC 9334: "Remote ATtestation procedureS (RATS) Architecture".

[3] OASIS KMIP: "Key Management Interoperability Protocol Specification Version 2.1".

[4] ETSI GS NFV-SEC 021: "Network Functions Virtualisation (NFV); Security; VNF Package Security 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 may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

[i.1] ETSI GR NFV-EVE 018: "Network Functions Virtualisation (NFV); Evolution and Ecosystem; Report on Multi-tenancy in NFV".

[i.2] ETSI GS NFV-IFA 011: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; VNF Descriptor and Packaging Specification".

[i.3] ETSI GS NFV-IFA 014: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Network Service Templates Specification".

[i.4] ETSI GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GS NFV 003 [i.4] apply.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GS NFV 003 [i.4] and the following apply:

ABAC	Attribute Based Access Control
AI	Artificial Intelligence
BYOE	Bring Your Own Encryption
BYOK	Bring Your Own Key
CC	Common Criteria
CCA	Confidential Compute Architecture
CSP	Communications Service Provider
DEK	Data Encryption Key
EAL4	Evaluation Assurance Level 4
ECC	Elliptic Curve Cryptography
FIPS	Federal Information Processing Standards
GPU	Graphics Processing Unit
GSM	Global System for Mobile communications
HMEE	Hardware Mediated Execution Enclave
HSM	Hardware Security Module
HW	Hardware
HYOK	Hold Your Own Key
IBAC	Identity-Based Access Control
ID	Identifier
KDF	Key Derivation Function
KDK	Key Derivation Key
KEK	Key Encryption Key
KMIP	Key Management Interoperability Protocol
KMS	Key Management System
MANO-P	MANO for the Provider
MANO-T	MANO for the Tenant
MFA	Multi Factor Authentication
MLA	Management Level Agreement
NFVO-P	NFVO for the Provider

NFVO-T	NFVO for the Tenant
NMT	NFV-MANO Tenants
NS-SLA	Network Service - Service Level Agreement
RAN	Radio Access Network
RATS	Remote ATtestation procedureS
RBAC	Role Based Access Control
REQ	Requirement
REQ-ID	Requirement Identifier
RoT	Root of Trust
SEV	Secure Encrypted Virtualization
SGX	Software Guard eXtensions
SNP	Secure Nested Paging
SW	Software
TDX	Trust Domain Extension
TEE	Trusted Execution Environment
TLS	Transport Layer Security
vHSM	virtual Hardware Security Module
VIM-P	VIM for the Provider
VIM-T	VIM for the Tenant
VNFM-T	VNFM for the Tenant
vTPM	virtual Trusted Platform Module

4 Analysis of threat models in a multi-tenant NFV infrastructure

4.1 Introduction

A service provider may provide a network service for which some sensitive VNFs require a strict isolation with the other VNFs. This is a first use case that this present document analyses:

- Use Case #0: Provide Isolation of two VNFs of a single NS

The multi-tenancy is another case where protection and isolation is required.

Multi-tenancy in NFV is analysed in the report ETSI GR NFV-EVE 018 [i.1]. ETSI GR NFV-EVE 018 [i.1] focuses on use-cases where multiple users consume services from the same NFV-MANO or virtual resources from a same NFVI. ETSI GR NFV-EVE 018 [i.1] analyses the need of protection and isolation for the multi-tenancy for several use cases. It identifies as well the solutions or the holes in the specifications of the different reference points to ensure the identification of such protection and isolation needs for the network services or network resources, and the solutions provided by these specifications to implement protection and isolation.

The use cases analysed in ETSI GR NFV-EVE 018 [i.1] are:

- Use Case #1: Two users with own NFVO on shared NFVI
- Use Case #2: Two users share the same NFV environment
- Use Case #3: Network slicing by a single user
- Use Case #4: Nested network services
- Use Case #5: Tenants of a service provider
- Use Case #6: Two users with own MANO stack managed by provider MANO
- Use Case #7: Provide Isolation on different levels
- Use Case #8: Isolation of containerized VNF instances
- Use Case #9: Multiple NMTs use the same entity

Protection and isolation could be of different types. There could be a management isolation, or resources isolation.

The resources isolation could be of different types: Resource usage, resource information, resource usage monitoring, traffic access, resource availability.

The threat analysis of this clause will analyse in details each use cases and identifies the specific threats for each, and the requirements to mitigate these threats.

4.2 Threat analysis

4.2.1 Use Case #0: Provide Isolation of two VNFs of a single NS

4.2.1.1 Description

Security isolation may be required for some critical VNFs within a single NS or for some components of a VNF to make sure that the resources of these VNFs/VNFCs are not affected by the other VNFs/VNFCs.

This use case analyses the virtual resources isolation at different level in the NFVI for two VNFs/VNFCs that require a security isolation:

a) Virtual resources of the two VNFs/VNFCs can share the same physical resource. In this case, isolation needs to be guaranteed by the virtualization layer, i.e. the hypervisor, or by the physical hardware resources, e.g. use of HMEE in case of compute resources, or encryption of storage resources, or traffic encryption for network resources.

b) Virtual resources of the two VNFs/VNFCs can use different physical resource. In this case, isolation is guaranteed by separated hardware. However, physical servers may still share some network equipment.

c) The hardware in NFVI-PoPs may be organized in multiple zones. The entities within a zone may share physical environment, air-conditioning, power or networking (e.g. switches).

d) Virtual resources may be deployed in different NFVI-PoPs that adds additional security isolation over NFVI-PoPs. This could be required for geo-redundancy in case of disaster recovery, security compromise of a data centre or very high level of isolation for critical components.

The anti-affinity groups defined in ETSI NFV-IFA specifications provide a mechanism to tell the VIM necessary isolation needs. Affinity and anti-affinity rules can be defined for the constituents of VNFs as described in ETSI GS NFV-IFA 011 [i.2] and for the constituents of NSs and NS instances by the service provider when designing the network services, see ETSI GS NFV-IFA 014 [i.3]. Clause 5.7.2.2 of ETSI GR NFV-EVE 018 [i.1] explains how the affinity and anti-affinity groups can be used for virtual resource isolation.

The mechanism to tell the VIM necessary need of security isolation at the physical resource level using for example HMEE, data and traffic encryption is not described at this stage in the ETSI NFV-IFA specifications.

4.2.1.2 Threat analysis

Table 4.2.1.2-1: Threats for the use-case #0

Threat-Id	Description of threat	Assets concerned	Mitigation Requirements: REQ-ID
Use Case #0.1	A malicious software in the NFV-MANO, modifies the affinity and anti-affinity rules for the constituents of VNFs/NS in the catalogue or during an instantiation operation requested to the VIM, modifying the virtual resource isolation needs for these VNFCs/VNFs and enabling further attacks.	VNF package data, NS Descriptors, VNF Descriptors,VNFs resources	REQ-4; REQ-5; REQ-7; REQ-8; REQ-16; REQ-17; REQ-18; REQ-22; REQ-24; REQ-29
Use Case #0.2	A malicious software in the NFV-MANO, modifies the information enabling the isolation at physical resource level for the constituents of VNFs in the catalogue or during LCM operation requested to the VIM, modifying the physical resource isolation needs for these VNFs and enabling further attacks.	VNF package data, NS Descriptors, VNF Descriptors,VNFs resources	REQ-4; REQ-5; REQ-7; REQ-8; REQ-16; REQ-17; REQ-18; REQ-22; REQ-24; REQ-29
Use Case #0.3	An attacker is able to exploit vulnerabilities on hypervisor and mount through malicious VM or vulnerable VM (which becomes compromised) a VM escape attack.	VNFs resources, VNF application data	REQ-9; REQ-10; REQ-11; REQ-12; REQ-13; REQ-30; REQ-33; REQ-34
Use Case #0.4	An attacker is able to exploit vulnerabilities of the container environment and mount through malicious container or vulnerable container (which becomes compromised) a container escape attack.	VNFs resources, VNF application data	REQ-9; REQ-10; REQ-11; REQ-12; REQ-13; REQ-31; REQ-33; REQ-34
Use Case #0.5	An attacker is able to exploit a vulnerability of the virtualized system and mount a privilege escalation attack enabling to perform host privilege operations, and gain access to unauthorized hardware resources.	VNFs resources, VNF application data	REQ-9; REQ-10; REQ-11; REQ-12; REQ-13; REQ-32; REQ-33; REQ-34
Use Case #0.6	A malicious VNF overwhelms the virtual memory of vSwitch by sending fake requests , provoking a Denial of Service in the other VNFs, and for the NS.	VNFs resources; Legal agreements (SLA)	REQ-3

4.2.2 Use Case #1: Two users with own NFVO on shared NFVI

4.2.2.1 Description

The clause 5.1 of ETSI GR NFV-EVE 018 [i.1] gives details on this use case. A high-level description is given hereafter for convenience.

In this use case, two users uses their own NFVO and VNFM but allocates resources on the same NFVI-PoP(s) to build their NSs. It is assumed for this use case that a NFVI-PoP is managed by a single VIM and that the NSs of the different users are not sharing physical resources.

For this use case, the on-boarding of NSs is done on a tenant-specific NFVO. The relevant descriptors NSD and VNFD are then isolated from the descriptors of other tenants.

The threats on isolation starts when the NFVO or the VNFM request resources for the VNFs instantiation to the VIM.

4.2.2.2 Threat analysis

Table 4.2.2.2-1: Threats for the use-case #1

Threat-Id	Description of threat	Assets concerned	Mitigation Requirements: REQ-ID
Use Case #1.1	A malicious NFVO or VNFM request instantiation of a VNF on behalf of a legitimate tenant, using the Or-Vi or Vnfm-Vi interface respectively. This malicious NFVO or VNFM uses for the instantiation of the VNF the resourceGroupId of the legitimate tenant. This VNF sharing the domain of other VNFs of the legitimate tenant may access to the resources of the legitimate VNFs.	VNF application data, VNF sensitive parameters, VNF Lawful Interception data	REQ-1; REQ-2
Use Case #1.2	A malicious NFVO or VNFM request reservation of resources (compute, storage, network) using Or-Vi or Vnfm-Vi interface respectively, to jeopardize the NFVI resources available. This malicious NFVO or VNFM uses its own resourceGroupId or the resourceGroupId of the legitimate tenant.	NS SLA, VNF Instance virtual resources	REQ-1; REQ-2; REQ-3
Use Case #1.3	A malicious NFVO uses the Or-Vi software image interface on behalf of a legitimate tenant authorized to access the VIM, to add, delete, update or to query information of Software images from the VIM.	VNF Software images.	REQ-4; REQ-5; REQ-6
Use Case #1.4	A malicious VNFM uses the Vnfm-Vi software image interface on behalf of a legitimate tenant authorized to access the VIM, to query information of Software images from the VIM.	VNF Software images	REQ-5; REQ-6; REQ-7
Use Case #1.5	VNF instantiated in the NFVI accesses to the resources reserved for another tenant and not sharable.	VNF application data	REQ8; REQ-9; REQ-10; REQ-11
Use Case #1.6	A malicious tenant accesses in the WIM to data transiting between another tenant's two VNF instances instantiated in two different NFVI-PoPs.	VNF application data	REQ-13

4.2.3 Use Case #2: Two users share the same NFV environment

4.2.3.1 Description

The clause 5.2 of ETSI GR NFV-EVE 018 [i.1] gives details on this use case. A high-level description is given hereafter for convenience.

In this use case, two service providers use the same NFV environment (NFVO and other FBs) to create their own NSs built with resources of the same NFVI-PoP(s).

For this use case, the service providers use the same NFV-MANO service to deploy their NS instances. It is assumed for this use case that a NFVI-PoP is managed by a single VIM and that the NSs of the different users are not sharing physical resources.

4.2.3.2 Threat analysis

The threats of the use-case #1 applies also for this use-case. The additional threats linked to the fact that the service providers uses the same NFVO and VNFM are listed in table 4.2.2.2-1.

Table 4.2.3.2-1: Threats for the use-case #2

Threat-Id	Description of threat	Assets concerned	Mitigation Requirements: REQ-ID
Use Case #2.1	A legitimate tenant 1 uses the Os-ma-nfvo interface to read the NS information of another tenant 2 sharing the NFVO. The tenant 1 may get sensitive information on the NS topology for a NS of a competitor (tenant 2).	Network topology, VNF package data	REQ-14, REQ-15, REQ-16, REQ-17, REQ-18
Use Case #2.2	A malicious tenant on-boards unused NS/VNF just to consume on-boarding resources (e.g. fill the NS and VNF registries or software image repository) to limit the space available for other tenant.	NS-SLA, NS availability	REQ-19
Use Case #2.3	A malicious tenant uses the Os-ma-nfvo interface to manage the NSs of another tenant. For example, this malicious tenant may scale down the NS of a competitor to get more resources for his own NS or scale up to increase resource cost of another tenant.	NS-SLA, NS availability, VNFs resources	REQ-20, REQ-21, REQ-22, REQ-23, REQ-24

4.2.4 Use Case #3: Network slicing by a single user

4.2.4.1 Description

The clause 5.3 of ETSI GR NFV-EVE 018 [i.1] gives details on this use case. A high-level description is given hereafter for convenience.

In this use case, a service provider uses network slicing and creates two slice subnets by creating network service instances on the same NFV environment (i.e. the same NFVO and other FBs) and thus being built with resources of the same NFVI-PoP(s). In difference to use-case #2, the same service provider manages both NS instances and expects isolation of the slice subnets.

In this use case it is assumed that the NS instances of the different slice subnets can be deployed on different resources within the NFVI. For sharing of NFVI resources see other use cases.

The different slice subnets may be created using different NS or using different instances of the same NS, Each slice subnet may also contain different NS instances. In this latter case, the isolation applies for the slice subnets (i.e. the NS instances belonging to different slice subnets) but not for the NS instances constituents of a single slice subnet.

4.2.4.2 Threat analysis

For this use case, the NFV-MANO is not multi-tenant, and manages and orchestrates network services of a single service provider. In this case isolation shall be effective on the resources used for the different slice subnets but not on the management of the slice subnets.

The use-case #1.5 threat of the use-case #1 applies also for this use-case except that there are not different tenant in this case but different isolated slice subnets of a same tenant. Inside the NFVI, this could be managed in the same way using the resource Group Id as identifier of the trust domain.

Table 4.2.4.2-1: Threats for the use-case #3

Threat-Id	Description of threat	Assets concerned	Mitigation Requirements: REQ-ID
Use Case #3.1	A compromised (intentionally or simply misconfigured) VNF instantiated in one slice subnet is able to access resources of another slice subnet.	VNF application data	REQ8; REQ-9; REQ-10; REQ-12

4.2.5 Use Case #4: Nested network services

4.2.5.1 Description

The clause 5.4 of ETSI GR NFV-EVE 018 [i.1] gives details on this use case. A high-level description is given hereafter for convenience.

In this use case, service provider 1 uses a network service NS1 with a nested network service NS3. Another service provider 2 uses a network service NS2 with a nested network service NS4. NS3 and NS4 are managed by the same NFVO, NFVO3. NS1 and NS2 are managed by two other NFVOs, NFVO1 and NFVO2 respectively. NS1 and NS3 may share resources. NS2 and NS4 may share resources. Isolation is needed between the even and odd numbered NSs and their constituents and resources. NS3 and NS4, managed by the same NFVO, need isolation both for their management access and for their resources. NS3 and NS4 may share a VNF using the same VNF package in NFVO3, but isolation is needed for their respective VNF instances.

In this use case, NS3 is instantiated through the Or-Or reference point between NFVO1 and NFVO3. In the same way, NS4 is instantiated through the Or-Or reference point between NFVO2 and NFVO3. The isolation of NS3 and NS4 is similar to that of use-case #2, however the difference in this case is due to the reference point that manages the NS. In use-case #2, the NS is managed solely through the Os-Ma-Nfvo reference point, but in this current use case, the management of the NS is initiated bn the Os-Ma-Nfvo but processed through an Or-Or reference point.

It is assumed that for this use case the VNFs of different network services may be either instantiated in the same or different NFVI-PoPs, but are managed by the same VIM.

The detailed flows for instantiation of the NS and their nested NS are described in clause 5.4 of ETSI GR NFV-EVE 018 [i.1].

The VNF package on-boarded in the NFVO is signed and may be encrypted with the service provider credentials as defined in ETSI GS NFV-SEC 021 [4] and ETSI GS NFV-SOL 004 [1]. When a Nested NS is on-boarded in an external NFVO, the protection of the sensitive artifact is done as described in clause 5.5 of ETSI GS NFV-SOL 004 [1].

In case of asymmetric encryption, the key used to encrypt the sensitive artifact is the public key of the external NFVO. The external NFVO uses its own private key to decrypt the encrypted artifact.

In case of symmetric encryption, the public key of the external NFVO is used to encrypt a key generated by the VNF provider. This latter key is used to encrypt the sensitive artifact. The encrypted form of this key is included in the package to be shared with the external NFVO. The external NFVO decrypts the shared key with its own private key and then uses the obtained shared key to decrypt the artifact.

4.2.5.2 Threat analysis

For this use case the NFV-MANOs, including the NFVO3, are multi-tenant and shall ensure the isolation between these tenants. The VIM is also multi-tenant and shall ensure the isolation of resources for the VNF instances of different service providers and shall be aware of this multi-tenancy.

The threats of the use-case #2 apply also for this use-case except that the interface used is the Or-Or reference point. The resource Group Id may be used as identifier of the trust domain for managing the isolation in NFVO3, VNFM3, VIM, and NFVI.

Table 4.2.5.2-1: Threats for the use-case #4

Threat-Id	Description of threat	Assets concerned	Mitigation Requirements: REQ-ID
Use Case #4.1	A legitimate tenant 1, that uses an NFVO for a nested NS constituent of an NS, uses the Os-ma-nfvo interface to read the nested NS information of another tenant 2 sharing the NFVO through the Or-Or interface. The tenant 1 may get sensitive information on the NS topology for an NS of a competitor (tenant 2).	Network topology, VNF package data	REQ-16, REQ-17, REQ-18, REQ-26
Use Case #4.2	A malicious tenant on-boards an unused NS/VNF in an NFVO, through another NFVO (using the Or-Or interface), to consume on-boarding resources (e.g. fill the NS and VNF registries or the software image repository) to limit the space available for another tenant(s).	NS-SLA, NS availability	REQ-19
Use Case #4.3	A malicious tenant uses the Os-ma-nfvo interface to manage the NSs of another tenant through the Or-Or interface. For example, this malicious tenant may scale down the NS of a competitor to get more resources for their own NS or, conversely, scale up to increase resource cost of another tenant.	NS-SLA, NS availability, VNFs resources	REQ-20, REQ-21, REQ-22, REQ-23
Use Case #4.4	A malicious tenant uses a VNF belonging to another tenant sharing the NFVO for their own nested NS. The instantiation of the nested NS is done through the Or-Or interface.	VNF Package,	REQ-20
Use Case #4.5	A legitimate tenant sharing a VNF package with another tenant for their nested NS deletes the VNF package through the Or-Or Interface, stopping the Network Service of the other tenant.	NS availability	REQ-27
Use Case #4.6	A malicious tenant uses their legitimate Os-Ma-Nfvo interface to manage an NS and its constituents on behalf of another tenant through the Or-Or interface.	NS availability, VNFs resources, VNF Package	REQ-25
Use Case #4.7	A malicious NFVO intercepts an on-boarding or management request, from the Os-Ma-Nfvo interface to an NS or NS constituent of a nested NS in an external NFVO, and register a false owner of the NS/NS constituent with an altered resourceGroupId, thus changing the isolation needs and the access control of the NS/NS constituent.	VNF Package, NS Descriptors, VNF resources	REQ-24
Use Case #4.8	A malicious NFVO intercepts an on-boarding or management request, from the Os-Ma-Nfvo to an NS/NS constituent of a nested NS in an external NFVO, and sends the request to a malicious NFVO.	VNF Package, NS descriptors, NS availability, VNF resources	REQ-28

4.2.6 Use Case #6: Two users with own MANO stack managed by provider MANO

4.2.6.1 Description

The clause 5.7 of ETSI GR NFV-EVE 018 [i.1] gives details on this use case. A high-level description is given hereafter for convenience.

In this use-case, two NFV-MANO Tenants (NMTs) are able to manage their own services and resources with autonomy using their own MANO stack (MANO-T) managed and monitored by a MANO provider (MANO-P). The NMT can request to the MANO provider a full MANO stack with all the Functional Blocks (FBs) of a MANO (i.e. NFVO-T, VNFM-T and VIM-T) or a partial MANO stack. The NMT and the MANO provider negotiate the type and bounds on which lifecycle management operations the MANO-T system is allowed to execute within its reserved resource quota/pool/zone without involving the MANO-P system; for example, the MANO-T may be allowed to instantiate NS instances within the resource quota of the NMT, scale-out the VNF but within a certain limit or migrate virtualized resources or instantiate VNFs beyond a certain number. This management agreement negotiated between NMT and MANO Provider is the Management Level Agreement (MLA).

ETSI GR NFV-EVE 018 [i.1] gives:

- The flow for enabling the Provider of the NFV-MANO for deploying the MANO-T system for a NMT, reserving the virtualized resource quota by the VIM-P, and after verification that the MLA is acceptable for the MANO-P, the NFVO-P instantiates and configures the different FBs of the MANO-T according to the MLA, in the same way an NS is instantiated.
- The flow of MANO-T performing the NS LCM operation within the limits specified in the MLA, and the scale-out of VNF instances which exceed the scale-out limit specified in the MLA.
- And the flow of LCM operation exceeding the MLA bounds.

4.2.6.2 Threat analysis

The instantiation of a MANO-T system is similar to that of a NS instance, with MANO-T FBs being the VNFs. The same threats apply than those of use-case #2, where two service providers use the same NFV environment (NFVO and other FBs) to create their own NSs built with resources of the same NFVI-PoP(s).

Additional threats applying to the resources quota and MLA template are listed in table 4.2.6.2-1 below.

Table 4.2.6.2-1: Threats for the use-case #6

Threat-Id	Description of threat	Assets concerned	Mitigation Requirements: REQ-ID
Use Case #6.1	A malicious software in MANO-P modifies in the VIM-P the virtualized resource quota under the control of a MANO-T, either to give more resources to the NMT to the detriment of other tenant or to decrease the resources and obtain a denial of service on the NS managed by the MANO-T of this NMT.		REQ-42
Use Case #6.2	A malicious software in MANO-P modifies the MLA agreed between the NMT and the MANO provider, either to decrease the autonomy of the MANO-T and reduce e.g. the capability of scaling for the NS of NMT, or increase the autonomy of the MANO-T.		REQ-43, REQ-44
Use Case #6.3	A malicious software in NFVO-T modifies the limits configured according to MLA agreed between the MNT and the MANO provider, and either e.g. scales the VNF beyond the limit without seeking permission to NFVO-P to execute the scale operation, or overwhelmes the NFVO-P with requests seeking permission to execute scaling.		REQ-45, REQ-46

4.2.7 Use Case #7: Provide Isolation on different levels

4.2.7.1 Description

The clause 5.7 of ETSI GR NFV-EVE 018 [i.1] gives details on this use case. A high-level description is given hereafter for convenience, in the context of security isolation.

Usually security isolation is required to make sure that resources of one NS cannot affect another NS.

This use case analyses the virtual resources isolation at different level in the NFVI:

a) Virtual resources of components of two NS can share the same physical resource. In this case, isolation needs to be guaranteed by the virtualization layer, i.e. the hypervisor, or by the physical hardware resources, e.g. use of HMEE in case of compute resources, or encryption of storage resources, or traffic encryption for network resources.

b) Virtual resources of components of two NS can use different physical resource. In this case, isolation is guaranteed by separated hardware. However, physical servers may still share some network equipment.

c) The hardware in NFVI-PoPs may be organized in multiple zones. The entities within a zone may share physical environment, air-conditioning, power or networking (e.g. switches).

d) Virtual resources may be deployed in different NFVI-PoPs that adds additional security isolation over NFVI-PoPs. This could be required for geo-redundancy in case of disaster recovery, security compromise of a data centre or very high level of isolation for critical components.

NOTE: Levels A, B, C, D may be applied in parallel to different VNFs within a single NS (see use-case #0).

The anti-affinity groups defined in ETSI NFV-IFA specifications provide a mechanism to tell the VIM necessary isolation needs. Affinity and anti-affinity rules can be defined for the constituents of VNFs as described in ETSI GS NFV-IFA 011 [i.2] and for the constituents of NSs and NS instances by the service provider when designing the network services, see ETSI GS NFV-IFA 014 [i.3]. Clause 5.7.2.2 of ETSI GR NFV-EVE 018 [i.1] explains how the affinity and anti-affinity groups can be used for virtual resource isolation.

The mechanism to tell the VIM necessary need of security isolation at the physical resource level using for example HMEE, data and traffic encryption is not described at this stage in the ETSI NFV-IFA specifications.

To guarantee isolation of VMs in a hypervisor for level A, there shall be spatial, temporal and fault isolation (e.g. a partitioning hypervisor).

The spatial isolation prevents data and code of a VM from being altered or eavesdropped by another VM.

The temporal isolation ensures that the execution timing of one VM does not interfere with the others.

The fault isolation ensures that a fault in one VM does not affect the others.

4.2.7.2 Threat analysis

Table 4.2.7.2-1: Threats for the use-case #7

Threat-Id	Description of threat	Assets concerned	Mitigation Requirements: REQ-ID
Use Case #7.1	A malicious software in the NFV-MANO, modifies the affinity and anti-affinity rules for the constituents of VNFs/NSs in the catalogue or during an instantiation operation requested to the VIM, modifying the virtual resource isolation needs for these VNFs/NSs and enabling further attacks.	VNF package data, NS Descriptors,VNFs resources	REQ-4; REQ-5; REQ-7; REQ-8; REQ-16; REQ-17; REQ-18; REQ-22; REQ-24; REQ-29
Use Case #7.2	A malicious software in the NFV-MANO, modifies the information enabling the isolation at physical resource level for the constituents of VNFs in the catalogue or during LCM operation requested to the VIM, modifying the physical resource isolation needs for these VNFs and enabling further attacks.	VNF package data, NS Descriptors,VNFs resources	REQ-4; REQ-5; REQ-7; REQ-8; REQ-16; REQ-17; REQ-18; REQ-22; REQ-24; REQ-29
Use Case #7.3	An attacker is able to exploit vulnerabilities on hypervisor and mount through malicious VM or vulnerable VM (which becomes compromised) a VM escape attack.	VNFs resources, VNF application data	REQ-9; REQ-10; REQ-11; REQ-12; REQ-13; REQ-30; REQ-33; REQ-34
Use Case #7.4	An attacker is able to exploit vulnerabilities of the container environment and mount through malicious container or vulnerable container (which becomes compromised) a container escape attack.	VNFs resources, VNF application data	REQ-9; REQ-10; REQ-11; REQ-12; REQ-13; REQ-31; REQ-33; REQ-34
Use Case #7.5	An attacker is able to exploit a vulnerability of the virtualized system and mount a privilege escalation attack enabling to perform host privilege operations, and gain access to unauthorized hardware resources.	VNFs resources, VNF application data	REQ-9; REQ-10; REQ-11; REQ-12; REQ-13; REQ-32; REQ-33; REQ-34

Threat-Id	Description of threat	Assets concerned	Mitigation Requirements: REQ-ID
Use Case #7.6	A malicious NS overwhelms the virtual memory of vSwitch by sending fake requests provoking a Denial of Service in the other NS.	VNFs resources, Legal agreements (SLA)	REQ-3
Use Case #7.7	An attacker overwhelms the virtual memory of vSwitch by sending fake requests in order to impact one NS and as a consequence of the attack, the attacker makes all NSs unavailable for other tenants, provoking a Denial of Service.	VNFs resources, Legal agreements (SLA)	REQ-3
Use Case #7.8	An attacker uses a one or more VNFs within a malicious NS to attack another NS.	VNFs resources, VNF application data, Legal agreements (SLA)	REQ-1; REQ-2; REQ-3; REQ-4; REQ-5; REQ-6; REQ-7; REQ-8; REQ-9; REQ-10; REQ-11; REQ-12; REQ-13; REQ-14; REQ-15; REQ-16; REQ-17; REQ-18; REQ-22; REQ-24; REQ-29; REQ-30; REQ-31; REQ-33; REQ-34
Use Case #7.9	An attacker has compromised an NS (by compromising one or more VNFs within the NS) using a previous described threat, move the NS closer to another NS that is the final target of the attack to compromise this last NS.	VNFs resources, VNF application data, Legal agreements (SLA)	REQ-1; REQ-2; REQ-3; REQ-4; REQ-5; REQ-6; REQ-7; REQ-8; REQ-9; REQ-10; REQ-11; REQ-12; REQ-13; REQ-14; REQ-15; REQ-16; REQ-17; REQ-18; REQ-22; REQ-24; REQ-29; REQ-30; REQ-31; REQ-33; REQ-34
Use-case #7.10	An attacker is able to exploit a vulnerability of the virtualized system and using a malicious VM alter the data or code of another VM.	VNF application data	REQ-8; REQ-9; REQ-10; REQ-11; REQ-12 ; REQ-30 ; REQ-31 ; REQ-48; REQ-49
Use-case #7.11	An attacker is able to exploit a vulnerability of the virtualized system and using a malicious VM eavesdrops the data or code of another VM.	VNF application data	REQ-9; REQ-10; REQ-11; REQ-12; REQ-48 ; REQ-49
Use-case #7.12	An attacker is able to exploit a vulnerability of the virtualized system and using a malicious VM overwhelms the virtual resources (memory and CPU) provoking the decay of performance, starvation and throughput reduction of the other VM.	VNFs resources, Legal agreements (SLA)	REQ-47; REQ-50
Use-case #7.13	An attacker is able to exploit a vulnerability of the virtualized system and using a malicious VM provokes fault in memory or CPU usage that impact the other VMs provoking a Denial of Service.	VNFs resources, Legal agreements (SLA)	REQ-51

4.2.8 Use Case #8: Isolation of containerized VNF instances

4.2.8.1 Description

The clause 5.9 of ETSI GR NFV-EVE 018 [i.1] gives details on this use case.

This use case describes the case of containerized workloads that can share the same virtual machine. This is an additional level to the level A-D described in the use-case #7 (see clause 4.2.7). In this use case, the isolation depends on the capabilities of the container technology.

4.2.9 Use Case #9: Multiple NMTs use the same entity

4.2.9.1 Description

The clause 5.10 of ETSI GR NFV-EVE 018 [i.1] gives details on this use case.

Two different cases are described:

a) NMTs can share VNF packages and descriptors, but each NMT creates own instances.

b) NMTs share instances, Compute, storage or network resources are commonly used by both (or multiple) tenants.

For the case A, sharing the VNF package between tenants is challenging:

- The VNF package is on-boarded, updated, signed and encrypted by one tenant. The other tenant shall be aware of any modification in the VNF package, and shall be able to verify signature of the VNF package and potentially decrypt it.
- Some modification in the VNF package may have impact on the service of the other tenant that uses the same VNF package.

For the use-case B, sharing VNF instance between tenants requires an isolation of resources managed directly by the VNF design and cannot be guaranteed by the NFVI. Isolation with the others constituents of the NS of each tenant requires the use of a specific trust domain resource ID for the shared VNF instance and external end-point for the connection to the other NS constituents of each tenant.

The LCM management of the shared VNF instance (scaling, healing, etc.) is also a challenge and the control by several tenants shall be discouraged. This control should be done through one and only one VNFM, the VNFM of the owner of the VNF instance, i.e. the tenant that launched the instantiation of the VNF.

4.2.9.2 Threat analysis

Table 4.2.9.2-1: Threats for the use-case #9

Threat-Id	Description of threat	Assets concerned	Mitigation Requirements: REQ-ID
Use Case #9.1	A tenant 1 on-boards through the Os-ma-nvfo interface a VNF package that is used by another tenant 2. The tenant 2 further updates the VNF package with a new version through Os-ma-nfvo interface without notifying the tenant 1. The new version of the VNF package has impact on the tenant1 NS (e.g. shut down, degradation of performance).	NS-SLA, NS availability	REQ-35, REQ-36, REQ-27
Use Case #9.2	A legitimate tenant 1 owner of a VNF package (see note) uses the Os-ma-nfvo interface to update a VNF package. The VNF package is used by another tenant 2 to build its own NS. The new version of the VNF package has impact on the tenant2 NS (e.g. shut down, degradation of performance).	NS-SLA, NS availability	REQ-37, REQ-38
Use Case #9.3	A NS provider 1 that shares a NS constituent with another NS provider 2 uses the shared NS constituent to mount an attack on the other NS constituent of NS provider 2.	NS-SLA, NS availability, VNF application data	REQ-39, REQ-40
Use Case #9.4	The LCM of a shared VNF instance between two tenants receives conflicting order from the two tenants.	NS-SLA	REQ-41
NOTE: The owner of a VNF package is the tenant that first on-boards the VNF package.			

4.3 Requirements

Table 4.3-1: Security Requirements

REQ-Id	Description of mitigation requirement	Associated Threat-Id
REQ-1	The resourceGroupId used for isolation identification shall be linked to the legitimate tenant authorized to use it.	Use Case #1.1; Use Case #1.2; Use Case #7.8; Use Case #7.9
REQ-2	The API access control to the Or-Vi and Vnfm-Vi interface shall restrict access to the operation that uses the resourceGroupID for the legitimate tenant only.	Use Case #1.1; Use Case #1.2; Use Case #7.8; Use Case #7.9
REQ-3	A legitimate tenant shall be able to reserve a guaranteed minimum and maximum amount of physical resources in the VIM for its own network services.	Use Case #0.6; Use Case #1.2; Use Case #7.6; Use Case #7.7; Use Case #7.8; Use Case #7.9; Use Case #7.12
REQ-4	The consumer of the Or-Vi interface shall be authenticated.	Use Case #0.1; Use Case #0.2; Use Case #1.3; Use Case #7.1; Use Case #7.2; Use Case #7.8; Use Case #7.9
REQ-5	The access to the Software images registry shall be controlled and authorized only for the legitimate tenant or specific administrator authorized by the legitimate tenant.	Use Case #0.1; Use Case #0.2; Use Case #1.3; Use Case #1.4; Use Case #7.1; Use Case #7.2; Use Case #7.8; Use Case #7.9
REQ-6	For sensitive VNF software images, confidentiality and integrity protection of the registry using the tenant own key shall be supported.	Use Case #1.3; Use Case #1.4; Use Case #7.8; Use Case #7.9
REQ-7	The consumer of the Vnfm-Vi interface shall be authenticated.	Use Case #0.1; Use Case #0.2; Use Case #1.4; Use Case #7.1; Use Case #7.2; Use Case #7.8; Use Case #7.9
REQ-8	Resources associated to the VNF during instantiation in the NFVI shall take into account the isolation amongst tenant (or isolation amongst slice subnets) need of these resources identified by the resourceGroupId.and affinity anti-affinity group.	Use Case #0.1; Use Case #0.2; Use Case #1.5; Use Case #3.1; Use Case #7.1; Use Case #7.2; Use Case #7.8; Use Case #7.9; Use Case #7.10
REQ-9	Access control on the resources in NFVI shall be implemented.	Use Case #0.3; Use Case #0.4; Use Case #0.5; Use Case #1.5; Use Case #3.1; Use Case #7.3; Use Case #7.4; Use Case #7.5; Use Case #7.8; Use Case #7.9; Use Case #7.10; Use Case #7.11
REQ-10	Confidentiality protection of storage resources in NFVI shall be supported.	Use Case #0.3; Use Case #0.4; Use Case #0.5; Use Case #1.5; Use Case #3.1; Use Case #7.3; Use Case #7.4; Use Case #7.5; Use Case #7.8; Use Case #7.9; Use Case #7.10; Use Case #7.11

REQ-Id	Description of mitigation requirement	Associated Threat-Id
REQ-11	Confidentiality protection of storage resources using a KMS with a possibility for a tenant to bring in the system its own key shall be supported.	Use Case #0.3; Use Case #0.4; Use Case #0.5; Use Case #1.5; Use Case #7.3; Use Case #7.4; Use Case #7.5; Use Case #7.8; Use Case #7.9; Use Case #7.10; Use Case #7.11
REQ-12	Confidentiality protection of storage resources using a KMS shall be able to derive keys for different slice subnets from the tenant KMS master key.	Use Case #0.3; Use Case #0.4; Use Case #0.5; Use Case #3.1; Use Case #7.3; Use Case #7.4; Use Case #7.5; Use Case #7.8; Use Case #7.9; Use Case #7.10; Use Case #7.11
REQ-13	Each tenant's cross NFVI-PoP traffic shall be isolatable from all other tenants (e.g. based on a resource group ID or other tenant identifier).	Use Case #0.3; Use Case #0.4; Use Case #0.5; Use Case #1.6; Use Case #7.3; Use Case #7.4; Use Case #7.5; Use Case #7.8; Use Case #7.9
REQ-14	The consumer of the Os-ma-nfvo interface shall be authenticated.	Use Case #2.1; Use Case #7.8; Use Case #7.9
REQ-15	Authorization of Os-ma-nfvo API shall restrict access to NS/VNF descriptors and information (e.g. nsInfo, vnfInfo, Fault and Performance management) to the legitimate tenant. The parameters and filters used to query NS/VNF information or subscribe to notifications shall be used for access control in addition to basic access right to the API when allowing access for an authenticated consumer.	Use Case #2.1; Use Case #7.8; Use Case #7.9
REQ-16	Integrity protection of the NS/VNF packages in the registry using tenant specific credentials shall be supported.	Use Case #0.1; Use Case #0.2; Use Case #2.1; Use Case #4.1; Use Case #7.1; Use Case #7.2; Use Case #7.8; Use Case #7.9
REQ-17	Confidentiality protection of the NS/VNF packages in the registry using tenant specific credentials shall be supported.	Use Case #0.1; Use Case #0.2; Use Case #2.1; Use Case #4.1; Use Case #7.1; Use Case #7.2; Use Case #7.8; Use Case #7.9
REQ-18	The registry shall support being partitioned for each tenant with the support of an access control restricting access to legitimate tenant requests.	Use Case #0.1; Use Case #0.2; Use Case #2.1; Use Case #4.1; Use Case #7.1; Use Case #7.2; Use Case #7.8; Use Case #7.9
REQ-19	The on,boarding resources (e.g. packages registry/software image repository) shall be able to be partitioned for each tenant with a minimum and maximum amount of storage space.	Use Case #2.2; Use Case #4.2
REQ-20	The tenant of the NS shall be registered during the on-boarding of the NS and the NS constituents, and for each request related to this NS or NS constituents, the tenant shall be authenticated and verified before processing the operation.	Use Case #2.3; Use Case #4.3; Use Case #4.4
REQ-21	The binding between the tenant and NS or NS constituents shall be integrity protected.	Use Case #2.3; Use Case #4.3
REQ-22	The control for isolation for the NS and NS constituents shall be integrity protected.	Use Case #0.1; Use Case #0.2; Use Case #2.3; Use Case #4.3; Use Case #7.1; Use Case #7.2; Use Case #7.8; Use Case #7.9

REQ-Id	Description of mitigation requirement	Associated Threat-Id
REQ-23	The parameters and filters applied for NS/VNF management operation requests shall be used for the access control in addition to basic access right to the API when allowing access for an authenticated consumer.	Use Case #2.3; Use Case #4.3
REQ-24	Resource isolation requirements as requested by a tenant for an NS or NS component shall not be altered as the request is progressed from the NFVO to VNFM, VIM and NFVI or from NFVO to another NFVO. This isolation requirement shall be identified by a trust domain resource ID or affinity/anti-affinity group. The resourceGroupId specified in IFA may be used to populate the trust domain resource ID.	Use Case #0.1; Use Case #0.2; Use Case #2.3; Use Case #4.7; Use Case #7.1; Use Case #7.2; Use Case #7.8; Use Case #7.9
REQ-25	The external NFVO, where a nested NS in on-boarded using the Or-Or interface, shall be able to authenticate the initial requester of an NS/VNF management operation through the Os-Ma-Nfvo or shall be able to obtain proof that the requester has been authenticated.	Use-Case #4.1; Use Case #4.6
REQ-26	Authorization of Or-Or API shall restrict access to NS/VNF descriptors and information (e.g. nsInfo, vnfInfo, and Fault and Performance management) to only the legitimate tenant. The parameters and filters used to query NS/VNF information or subscribe to notifications shall also be used for access control in addition to basic access rights to the API when allowing access for an authenticated consumer.	Use Case #4.1;
REQ-27	The authorized users of the NS or NS constituents shall be registered during the on-boarding of the NS and the NS constituents. For each operational request related to this NS or NS constituents, the requester shall be verified for authentication and authorization of the operation before execution.	Use Case #4.5; Use Case #9.1
REQ-28	The legitimate tenant of an NS/NS constituent shall be able to authenticate the NFVO for where the NS/NS constituent is on-boarded.	Use Case #4.8
REQ-29	The NFV-MANO entities interfaces shall provide as a minimum the following protection: Transport layer protection for data transferred over exposed interfaces Authentication of transmitting party with verification of identity and location. Authorization mechanism for access control to resource by authorization server using proven protocols .message and session integrity checks on the interfaces	Use Case #0.1; Use Case #0.2; Use Case #7.1; Use Case #7.2; Use Case #7.8; Use Case #7.9
REQ-30	NFVI shall implement protection against hypervisor VM escape attacks.	Use Case #0.3; Use Case #7.3; Use Case #7.8; Use Case #7.9; Use Case #7.10
REQ-31	NFVI shall implement protection against container environment escape attacks.	Use Case #0.4; Use Case #7.4; Use Case #7.8; Use Case #7.9; Use Case #7.10
REQ-32	The access privilege in the virtualized environment shall be restrictive for the VNF to the lower level. The host privilege operation shall not be possible through the VNF, except if explicitly authorised by NFV-MANO or network security management systems on a per VNF image or VNF instance basis. Such access requirements are provided by the VNF vendor and shall be detailed in the VNFD or VNF package. NFV-MANO shall ensure that VNFs with privileged access to the NFVI are appropriately isolated from other VNFs (see note 1).	Use Case #0.5; Use Case #7.5
REQ-33	The NFVI shall implement mechanisms for ensuring Hardware Based Root of Trust and attestation at each layer.	Use Case #0.5; Use Case #7.3; Use Case #7.4; Use Case #7.5; Use Case #7.8; Use Case #7.9
REQ-34	The NFVI shall implement a secure patch management process of the NFVI, restricted to privileged user.	Use Case #0.5; Use Case #7.3; Use Case #7.4; Use Case #7.5; Use Case #7.8; Use Case #7.9
REQ-35	The management of a NS package or NS constituent package (upload, update, deletion) shall be restricted to the administrator (i.e. tenant owner) of the NS/NS constituent package, using an access control.	Use Case #9.1
REQ-36	The administrator of the NS/NS constituent shall be registered in the NFVO package registry during the on-boarding of the NS/NS constituent.	Use Case #9.1
REQ-37	A legitimate user of an NS/NS constituent shall be notified for any change in the NS/NS constituent package done by the legitimate administrator of the NS/NS constituent.	Use Case #9.2

REQ-Id	Description of mitigation requirement	Associated Threat-Id
REQ-38	Legitimate user of a NS/NS constituent shall be registered in the NFVO package registry by the administrator of the NS/NS constituent.	Use Case #9.2
REQ-39	A VNF instance shared between two tenants shall be instantiated within an independent trust domain than the other NS constituents of the two tenants.	Use Case #9.3
REQ-40	A VNF instance shared between two tenants shall be designed in a way that the connections with the other constituents of the NS are external end point (crossing the trust domain).	Use Case #9.3
REQ-41	The LCM of a shared VNF instance should be controlled by one tenant, owner of the VNF instance (see note 2).	Use Case #9.4
REQ-42	The virtualized resource quota configured in the VIM-P for a NMT MANO-T stack is a sensitive asset that shall be protected in integrity, during the process of configuration and in the storage.	Use Case #6.1
REQ-43	The MLA agreed between the NMT and the MANO provider is a sensitive asset that shall be protected in confidentiality and integrity during the process of request by the NMT for deploying the MANO-T system.	Use Case #6.2
REQ-44	The MLA agreed between the NMT and the MANO provider is a sensitive asset that shall be protected in confidentiality and integrity in the storage in NFVO-P.	Use Case #6.2
REQ-45	The limits configured in MANO-T according to MLA agreed between the MNT and the MANO provider is a sensitive asset that shall be protected in integrity and confidentiality during the process of MANO-T's configuration by the MANO-P.	Use Case #6.3
REQ-46	The limits configured in MANO-T according to MLA agreed between the MNT and the MANO provider is a sensitive asset that shall be protected in integrity and confidentiality in the storage of NFVO-T.	Use Case #6.3
REQ-47	The hypervisor shall take into account physical resource quotas for the VNF and allocate memory, CPU and I/O quotas for it accordingly.	Use Case #7.12
REQ-48	The hypervisor shall support spatial partitioning (memory, CPU, cache, I/O, Hardware resources).	Use Case #7.10; Case #7.11
REQ-49	The hypervisor shall support memory partition encryption.	Use Case #7.10; Case #7.11
REQ-50	The hypervisor shall support temporal partitioning.	Use Case #7.12
REQ-51	The hypervisor shall support fault partitioning.	Use Case #7.13
NOTE 1: Authorisation is defined by local operator policy. NOTE 2: The owner of the VNF instance is the tenant that instantiated the VNF instance.		

5 Trust domain separation at resource level

5.1 Introduction

The threat analysis in clause 4 of the present document shows that multitenancy in NFV, where the physical infrastructure is shared by multiple mobile network operators is a significant security challenge. This analysis highlights the need to harden and use technologies that isolate the workloads for each of those tenants.

This clause introduces the technologies and best practices to harden the infrastructure and strive for separating the trust domain at resource level.

5.2 Resource separation

The requirements of the clause 4 that relate to the resource separation are the following:

Table 5.2-1: Resource separation requirements

REQ-Id	Description of mitigation requirement	Associated Threat-Id
REQ-3	A legitimate tenant shall be able to reserve a guaranteed minimum and maximum amount of physical resources in the VIM for its own network services.	Use Case #0.6; Use Case #1.2; Use Case #7.6; Use Case #7.7; Use Case #7.8; Use Case #7.9
REQ-8	Resources associated to the VNF during instantiation in the NFVI shall take into account the isolation amongst tenant (or isolation amongst slice subnets) need of these resources identified by the resourceGroupId.and affinity anti-affinity group.	Use Case #0.1; Use Case #0.2; Use Case #1.5; Use Case #3.1; Use Case #7.1; Use Case #7.2; Use Case #7.8; Use Case #7.9

In Kubernetes®, a Namespace provides a mechanism for isolating groups of API resources within a single cluster. Associating a namespace or several namespaces to a tenant is a way for managing isolation between tenants.

A Pod may consume all the resources available (CPU and memory) and could cause other Pods to be evicted from the node. To avoid resources contention and the Denial-of-Service attack, a resource quota shall be allocated to a namespace.

In Kubernetes®, the resource quota is set using the ResourceQuota for a specific namespace. Mapping tenants to namespaces allows to set resource quota for a specific tenant.

With ResourceQuota, it is possible to limit the total sum of compute resources (CPU, Memory) that can be requested in a given namespace. Other extended resources as GPU resource for example could be limited as well. It is possible as well to set a quota for the storage resources (persistent volume, ephemeral storage) and to set a quota for other types of resources such as configmaps, persistentvolumeclaims, pods, services, secrets, etc.

It is possible to control the allocation of resources with more granularity than the namespace. In this case the LimitRange set constraints and enforce minimum and maximum compute resources usage per Pod or container in a namespace.

Same concept exists in the hypervisors and virtual machines are provisioned with the amount of resources it requires to do the job. It is also possible to use limits to control the resource usage.

Resource quotas and limits are sensitive information of the VM or container settings that the CISM and hypervisor shall protect securely.

Quotas cannot protect against all kinds of resource sharing, such as network traffic.

5.3 Data protection

5.3.1 Introduction

Data protection in a multi-tenancy environment is of high importance. New compliance and regulatory mandates around protection of sensitive information continue to be introduced, and existing regulations become more stringent. Isolation of data to prevent unauthorized access to sensitive information and to prevent alteration of the data shall be ensured.

The data shall be protected during the whole lifecycle of the data:

- Protection of data in transit.
- Protection of data at rest.
- Protection of data in use.

5.3.2 Protection of data in transit

The requirement of clause 4 that relates to the data protection in transit in a multi-tenant environment is the following:

Table 5.3.2-1: Protection of data in transit requirements

REQ-Id	Description of mitigation requirement	Associated Threat-Id
REQ-13	Each tenant's cross NFVI-PoP traffic shall be isolatable from all other tenants (e.g. based on a resource group ID or other tenant identifier).	Use Case #0.3; Use Case #0.4; Use Case #0.5; Use Case #1.6; Use Case #7.3; Use Case #7.4; Use Case #7.5; Use Case #7.8; Use Case #7.9

By default the pod-to-pod communication is allowed in a Kubernetes® cluster, with traffic that is unencrypted. It is possible to control pod-to-pod communication using network policies and restrict the communication to specific namespace labels.

But greater isolation may be provided by service meshes based on the workload identity, in addition to namespaces. Service meshes also offer encryption using mutual TLS protecting the data in motion.

To ensure isolation of traffic between virtual functions, encryption of the traffic shall be support and used.

When there are multiple facilities used for the provisioning of a service or for the purpose of backup, the communications should be cryptographically protected in transit between facilities.

With higher speeds and expanded network capacity, there will be more data in motion than ever, from the RAN to the edge data center and back to the core of the network where mass amounts of data will be aggregated and moved in large scale. On the other hand, increasing the latency wreaks havoc on real time applications and causes poor user experience or renders some applications such as remote surgery impossible.

For long distance data motion, the use of high speed encryption accelerators to encrypt the traffic, using strong cryptographic algorithms with strong integrity protection and with efficient bandwidth utilization is a best practice.

The separation of duties in the multitenancy case shall be realized using a cryptographic key management system that generates the encryption/decryption keys for each tenant with a high level of entropy, stores the master keys (used to derive the traffic keys) in a tamper resistant storage and controls the access to keys for each tenant's traffic. The master key shall be securely stored in hardware e.g. HSM, HMEE.

Each tenant should be able to manage their own hardware-based Root-of-Trust for transmission of data. In this case each tenant controls their keys and may Bring Their Own Keys (BYOK) and may Hold Their Own Keys (HYOK).

The key hierarchy is described in figure 5.3.2-1.

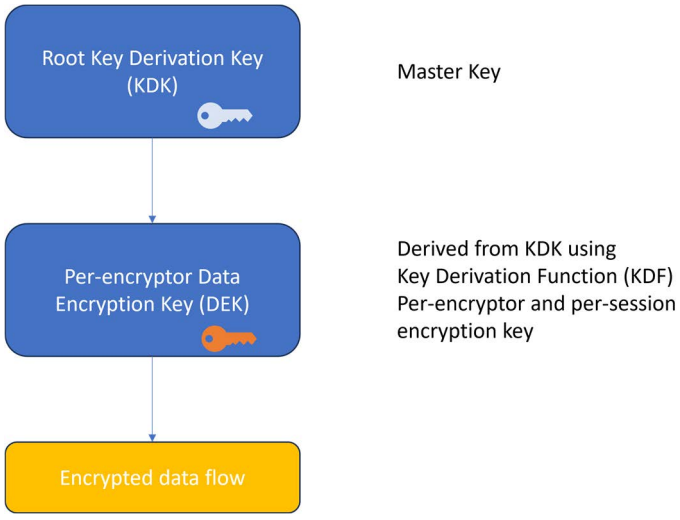

Figure 5.3.2-1: Key hierarchy for protection of data in transit

1) Master key root Key Derivation Key (KDK):

   a) This key is the foundation of the key hierarchy, and a master cryptographic anchor.

   b) This key is established and stored securely within the encryptors.

   c) This key is used to generate the per-session encryption keys deterministically.

2) Per-encryptor Data Encryption Keys (DEK):

   a) These keys are derived from the KDK using a Key Derivation Function (KDF). This ensures forward and backward secrecy. Even if a DEK is compromised past and future keys remain secure.

   b) These keys are unique per encryptor, ensuring data compartmentalization.

   c) These keys encrypt the data traffic.

   d) To enhance the security the keys are ephemeral encryption keys and rotate based on time or traffic-based triggers.

The master key (KDK) remains static in the encryptor and is used only for key derivation. Each Data Encryption Key (DEK) is unique to each traffic encryptor and is deterministically derived from the KDK. Each traffic encryptor generates a unique changing sequence of DEKs for the key rotation. This solution allows a group encryption and tunnel-less design. This enables any-to-any encryption topologies across thousands of encryptors without the operational burden of managing individual tunnels and enables scalability.

5.3.3 Protection of data at rest using transparent encryption

Data at rest are data that are provided by the VNF. Example of these data are:

- Persistent data and ephemeral data affecting the VNF processing.
- Persistent application data such as subscriber data used for network access.
- Confidential internal data such as authentication data, cryptographic keys but also logs and errors messages.

These data may contain Personally Identifiable Information and sensitive information of the subscriber and shall be protected at rest.

These data could be stored in primary storage, but also in mirrors or backup storage. Data protection shall address any type of storage.

Transparent encryption that encrypts data from a container or a VM to the persistent storage exists for VMs and containers and shall be used in multi-tenant infrastructure to avoid critical information to be exposed to other third parties, but also to privileged users. This transparent encryption may provide encryption capabilities at the granularity of the container or VM and protects data inside the containers (or VM) and in external storage accessible from the containers (or VM). It encrypts entire VM and all its dependency files (e.g. config files, snapshots, etc.).

The encryption shall use strong cryptographic algorithms with a strong integrity protection as recommended by relevant standard bodies such as NIST.

A rotation of keys shall be provided to add security in the transparent encryption.

To increase the security of the system the data encryption key shall never be left in clear in the hypervisor or container management system and shall be encrypted. A centralized key management system with the key encryption key stored in hardware support, e.g. HSM, HMEE is added to manage these key encryption keys with an access control and a secure storage. Hardware-based root of trust shall be used to protect data-at-rest. With multiple hardware-based roots-of-trust, e.g. HSM, HMEE, one for each tenant, each tenant controls their keys and may Bring Their Own Keys (BYOK) and may Hold Their Own Keys (HYOK).

Figure 5.3.3-1 shows the architecture of the transparent encryption using an External Key Broker service, independent to the Cloud Service Provider, to manage and control the keys and enabling the service provider, owner of the workloads, to Bring Its Own Keys (BYOK) or Hold Its Own Keys (HYOK). Figure 5.3.3-1 shows the architecture in the case of workloads in containers, but similar architecture may be used for the VM case.

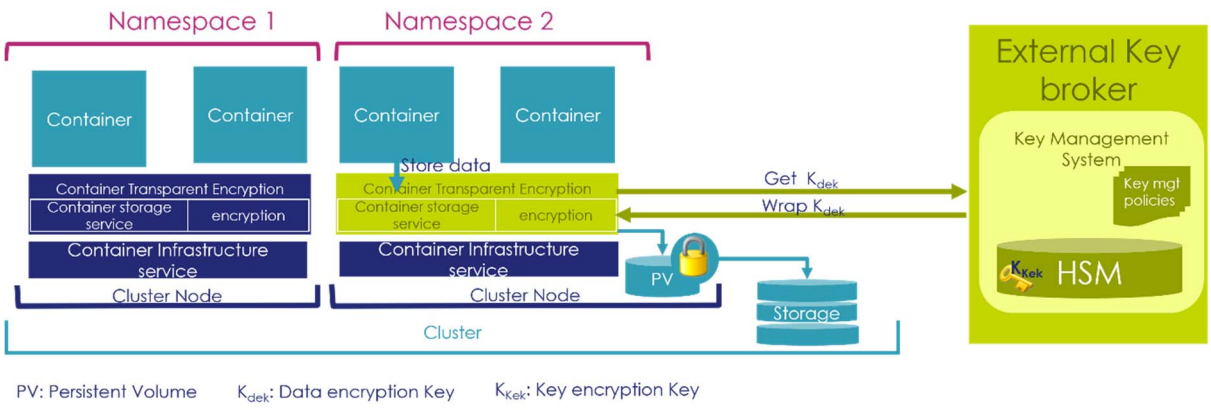

Figure 5.3.3-1: Transparent Encryption Architecture

Encryption of data in the Container Transparent Encryption may be done by the Cloud Service Provider as shown in figure 5.3.3-1 for namespace 1, or by the data owner either directly or through a trusted third party (i.e. the provider of the External Key Broker service) as shown in figure 5.3.3-1 for namespace 2. Separation of duties is ensured only if the Container Transparent Encryption is provided by the trusted party such as the external key broker provider (as shown for namespace2 in figure 5.3.3-1, and shall be the preferred option.

The Key Management System (KMS) of the External Key Broker service is responsible for managing the lifecycle of the keys. This key management is governed by the key management and usage policies that are sensitive assets. These policies define also the access control to the keys.

The key storage shall be done in a trusted environment such as an HSM. The KMS interacts with its dedicated HSM to generate, store, retrieve, encrypt and share keys with authorized targets.

The main characteristics of the KMS is that it can periodically and automatically rotate the cryptographic keys to increase the security of the system. Longer the key is used, the more likely it is to be compromised.

The Container Transparent Encryption service in the cluster node interacts with the External Key Broker service as described below in the flows of figure 5.3.3-2.

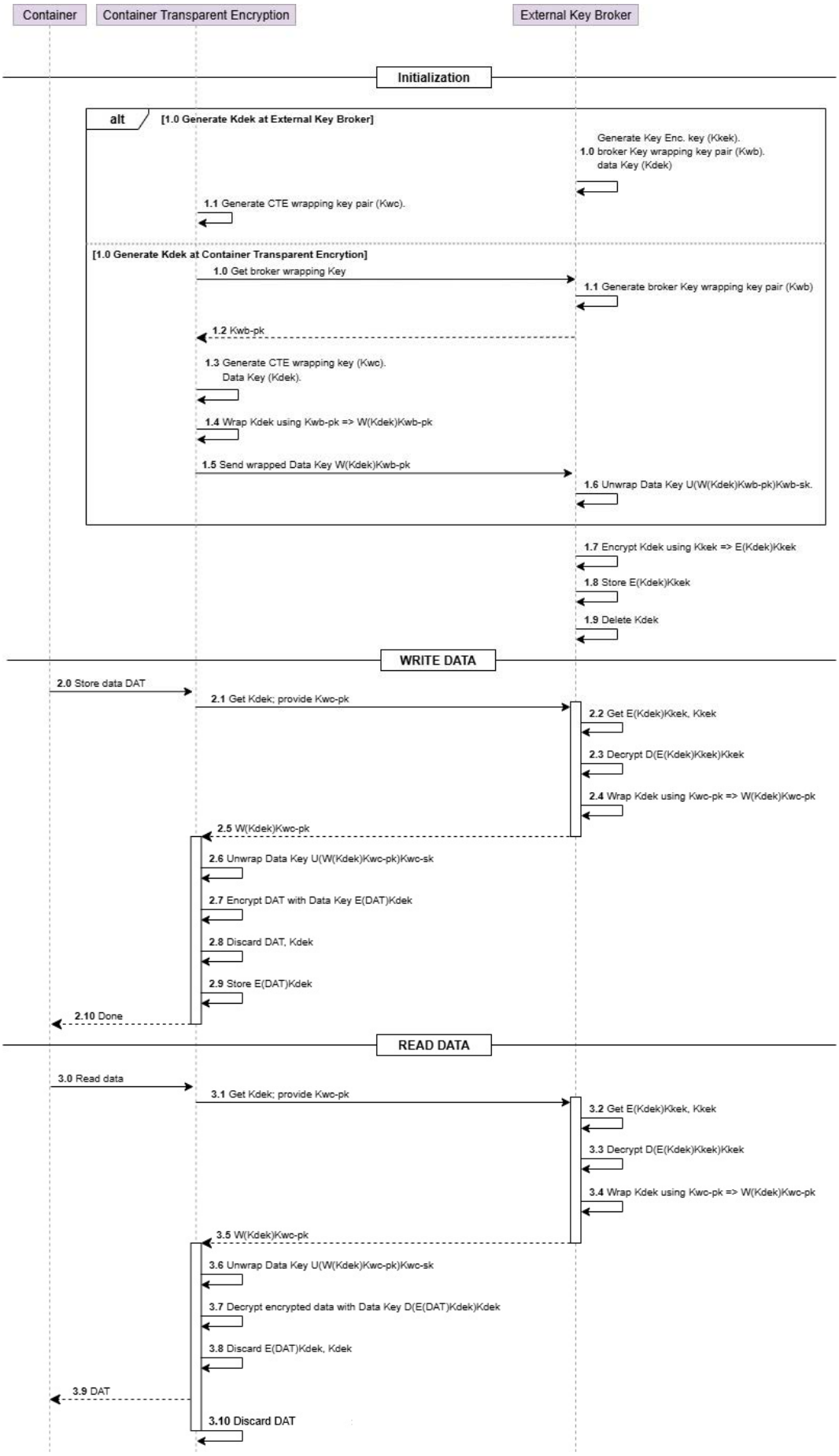

Figure 5.3.3-2: Transparent encryption flows

The discarding of DAT and Kdek in steps 2.8, 3.8 and 3.10 are important steps to ensure that the key and data are not left in clear in the Container Transparent Encryption. This implies an implementation of the Container Transparent Encryption by a trusted party (e.g. the External Key broker provider). This solution shall be the preferred option.

5.3.4 Protection of data in use

When data are processed by the Central Processing Unit (CPU) outside an enclave, they are held as plain text in memory. In multi-tenancy environments, the Containers and VM from the different tenants could be running on the same machine. To protect sensitive workloads and the data-in-use from the underlying privileged system or other physical access from third parties, Hardware-Mediated Execution Enclave (HMEE) shall be used. With HMEE Computations on data are performed in a cryptographically isolated hardware-based Trusted Execution Environment (TEE), removing, or reducing the ability for a rogue operator at a cloud provider to access code and data while being executed.

In the current HMEE, the cryptographic material that verifies the confidential computing environment and controls the workload protection is managed by the cloud provider without separation of duties. The separation of duties is a best practice to comply with data sovereignty requirements and keep control of keys.

The principle of separation of duties is used as a best practice for data protection. The benefits are the followings:

- Service provider's control of keys: to provide the service providers additional controls of their own data protection with keys managed by the service provider or a trusted partner designated by the customer; but in any case, the keys are controlled and managed by the service provider and outside the control of the cloud infrastructure provider.
- Separation of the holding of data and keys: to reduce or minimize the vulnerabilities associated with attacks to a single entity holding the data and keys.
- Trusted attestation service independent from the CSP: to verify by a CSP-independent trust party that the HMEE is not impersonated by a malicious actor. If a local attestation is only done by the Cloud Service Provider, there's not an independent verification of the authenticity of the SW/HW stack, with the risk of the potential service provider's sensitive workload being observed, and hacked, by an impersonated malicious actor.

A Hardware Security Module may be securely connected to a HMEE and provide the keys used in the HMEE, after a successful attestation process that ensures that the HMEE is genuine.

This is a way for the tenant to control the keys used and benefit from a Bring Your Own Key (BYOK) and Hold Your Own Key (HYOK) services.

This solution may be used for the protection of sensitive images during their migration to the cloud, from the on-boarding in the MANO, the storage in images registries up to the instantiation in the HMEE as described in clause 5.3.5.

Figure 5.3.4-1 shows the architecture, using a key broker service in the HMEE, that communicates with a Key Management Service for the key provisioning after a successful attestation.

This architecture allows a separation of duties, where the service provider controls the keys through a trusted key broker service, without any control of the infrastructure provider. The establishment of trust between the service provider and the key broker service is outside the scope of the present document.

NOTE 1: The colours used in Figure 5.3.4-1 do not represent trust domains but the different functionalities.

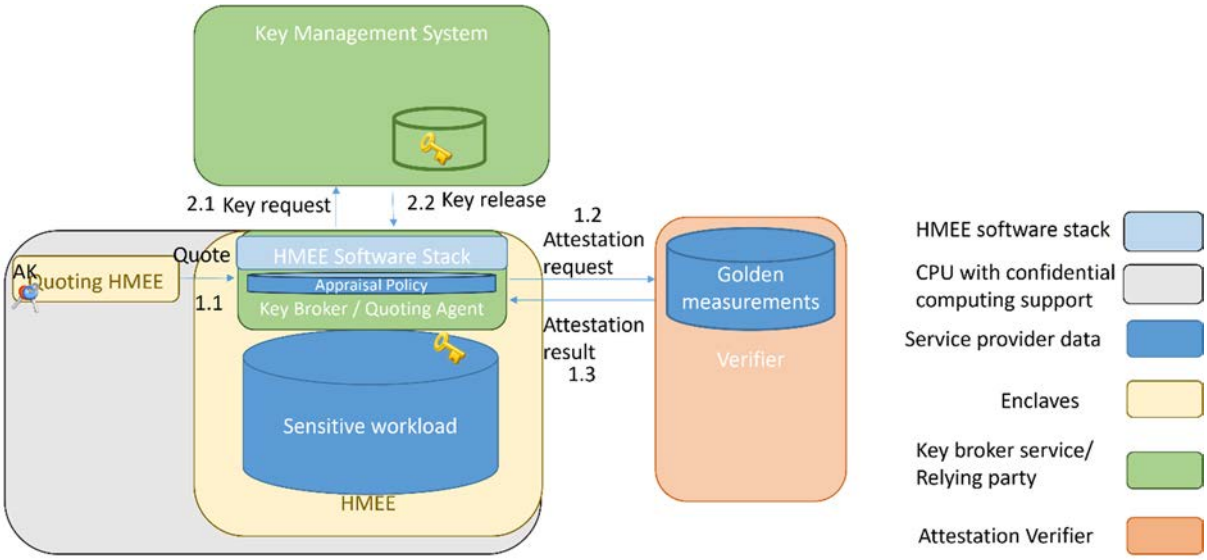

Figure 5.3.4-1: Architecture for key provisioning after attestation using a key broker service in HMEE

Description of the functional entities:

- Service Provider data:
  - The sensitive workload, containing the application that shall be instantiated in a HMEE.
  - The measurement of this sensitive workload that constitutes the golden measurement used by the Attestation Server during the attestation verification.
  - The appraisal policy used by the relying party (i.e. key broker service agent) to appraise the attestation result as described in IETF RFC 9334 [2].
- The attestation verifier that verifies against the golden measurements the attestation quote provided by the HMEE as defined by IETF RFC 9334 [2].

NOTE 2: The attestation verifier is the entity on which all the trust relies on and use the golden measurements of service provider. The service provider needs to trust the attestation verifier. There will be an implicit trust if the attestation verifier is in the trust domain of the service provider as shown in table 5.3.4-1 below. How the attestation verifier verifies the quote of a system in a multi-tenant environment needs further work.

- The key broker service:
  - A Key Management System including a tamper resistant key storage (e.g. an HSM).
  - An agent including the HMEE Software Stack, with the functionality of quoting agent requesting the verification of the quote generated by a Quoting HMEE, and the functionality of key broker allowing the provisioning by the Key Management System of secrets (e.g. key) if the attestation is successful. This key broker is a relying party as described by IETF RFC 9334 [2].

NOTE 3: These entities composing the Key broker service should be trusted by the Service Provider. There will be an implicit trust if the key broker service is in the trust domain of the service provider as shown in table 5.3.4-1 below.

- The HMEE software stack:
  - The HMEE software stack contains the virtual firmware/BIOS and Libraries integrated in the HMEE to run the sensitive applications and providing the APIs that abstract the communication with the CPU HW layer to request the quote and to abstract the communication layer with the quoting HMEE that provides the signed quote.

- The CPU:
  - The CPU used in this architecture supports the confidential computing functionality either using VM-Based HMEE (e.g. AMD SEV-SNP, Intel TDX, ARM CCA) or using Process-based HMEE (e.g. Intel SGX).
  - A quoting HMEE (e.g. Trust Domain Quoting Enclave for Intel TDX, vTPM). This quoting HMEE signs the quote with the private key part of the Attestation Key, that enables the attestation server to verify that the quote has been provided by a genuine confidential computing CPU.

The workflow depicts by this architecture figure 5.3.4-1 is divided in 2 phases:

- Phase 1: The attestation process.
- Phase 2: Provisioning of key for sensitive workload in the HMEE.

Phase 1:

- Step 1.0: Initialisation step: The quoting agent in the HMEE generates a key pair and exchanges the public key with the key management system.
- Step 1.1: The quoting agent in the HMEE request a nonce (nonce-ats) to the attestation server and request a nonce (nonce-kms) to the key management system. These nonces will be used for anti-replay protection. The quoting agent in the HMEE request the quote to the HW CPU and receives the quote report signed by the Quoting HMEE. The quoting agent in the request for the quote to the HW CPU may include the two nonces, nonce-ats and nonce-kms, as report data to ensure the freshness of the quote.
- Step 1.2: the quoting agent in the HMEE sends the attestation request including the signed quote.
- Step 1.3: the attestation server verifies the authenticity of the signed quote using the attestation public key, the freshness of the signed quote with the nonce-ats and verifies the measurements against the golden measurements. The attestation server sends back the attestation result that includes the nonces to the HMEE quoting agent.

Phase 2:

- Step 2.1: after successful verification of the attestation result and conformance to the appraisal policies, the Key broker agent requests the key for the sensitive workload using OASIS KMIP [3]. the Key broker agent includes in the request the attestation result including the nonces.
- Step 2.2: The KMS verifies with the nonce-kms the freshness of the attestation result and provides to the key broker service in the HMEE, the key wrapped with the public key of the key broker service in the HMEE and received during the initialization step 1.0. The key broker service in the HMEE de-wraps the key using its private key.
- Step 2.3: the workload is configured with the key and the application is launched.

For this architecture a trust relationship exists between the different actors.

Table 5.3.4-1 below shows the trust domain of the entities described in figure 5.3.4-1.

Table 5.3.4-1: Trust domain of the entities described in figure 5.3.4-1

Entity in the trust domain of	Infrastructure Provider	Service / KMS provider	Comments
CPU	X		
HMEE software stack		X	The key broker agent processing sensitive data (e.g. key) includes the OS and libs to control the security of the service.
Key broker service		X	The Key broker service processing service provider's data is owned by the service provider and in the service provider trust domain to ensure an implicit trust.
Attestation verifier		X	The attestation verifier using sensitive service provider's data is owned by the service provider and in the service provider trust domain to ensure an implicit trust.
Service provider data		X	These data are owned by the service provider.

Figure 5.3.4-2 shows the architecture, using an external key broker service, that communicates with a Key Management Service for the key provisioning after a successful attestation.

NOTE 4: The colours used in figure 5.3.4-2 below do not represent trust domains but the different functionalities.

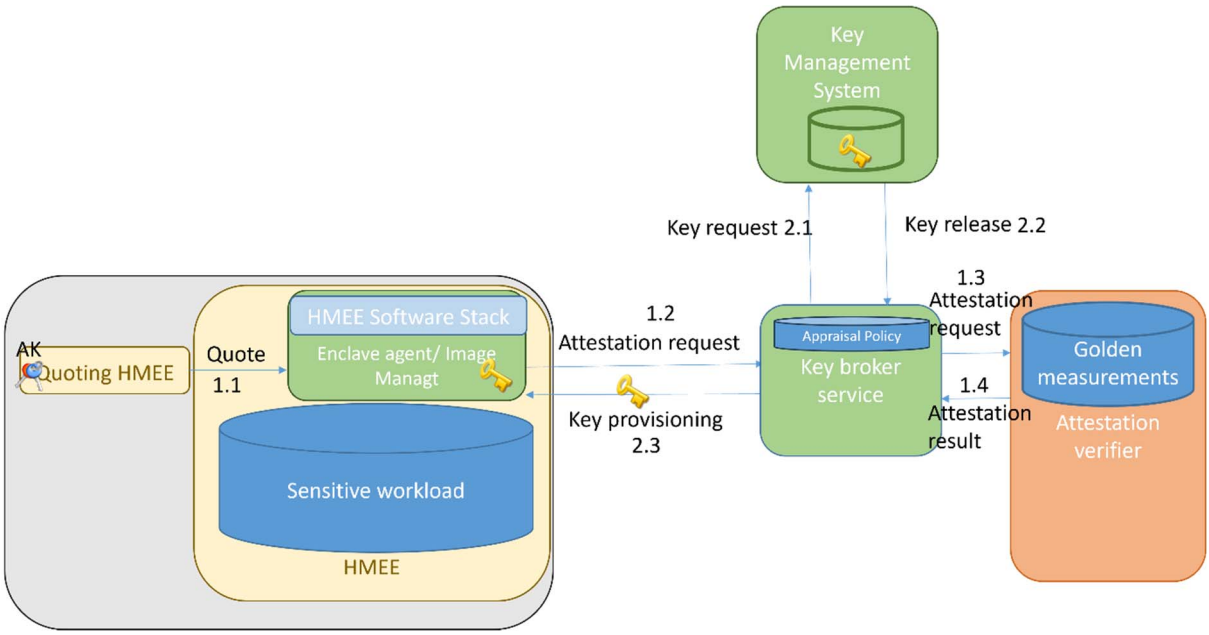

Figure 5.3.4-2: Architecture for key provisioning after attestation using an external key broker service

5.3.5 Protection of Software images

Software images shall be protected in integrity and authenticity and the requirements specified in ETSI GS NFV-SEC 021 [4], clause 5.2 shall apply.

As described by the REQ-6 in clause 4 of the present document, sensitive workload images shall be protected in confidentiality during their whole lifecycle (prior to on-boarding up to instantiation in an HMEE) and migration to the cloud (i.e. in all registries). For this protection, the sensitive workload images are encrypted before they are on-boarded into the NFV-MANO and before their signing for the integrity and authenticity protection. They remain encrypted in the different stores (images registries, images mirrors, etc.) and when they are instantiated in the HMEE. A key management system, controlled by the tenant, provides the encryption key for the encryption of the workload image, and after a successful attestation of the content of HMEE, provides the decryption key (wrapped with a public key of the HMEE) to decrypt the software image and to allow its execution.

Figure 5.3.5-1 below gives an architectural view of the encryption/decryption of sensitive workloads image using a key management system provided by the trusted provider.

NOTE 1: The colours used in figure 5.3.5-1 do not represent trust domains but the different functionalities.

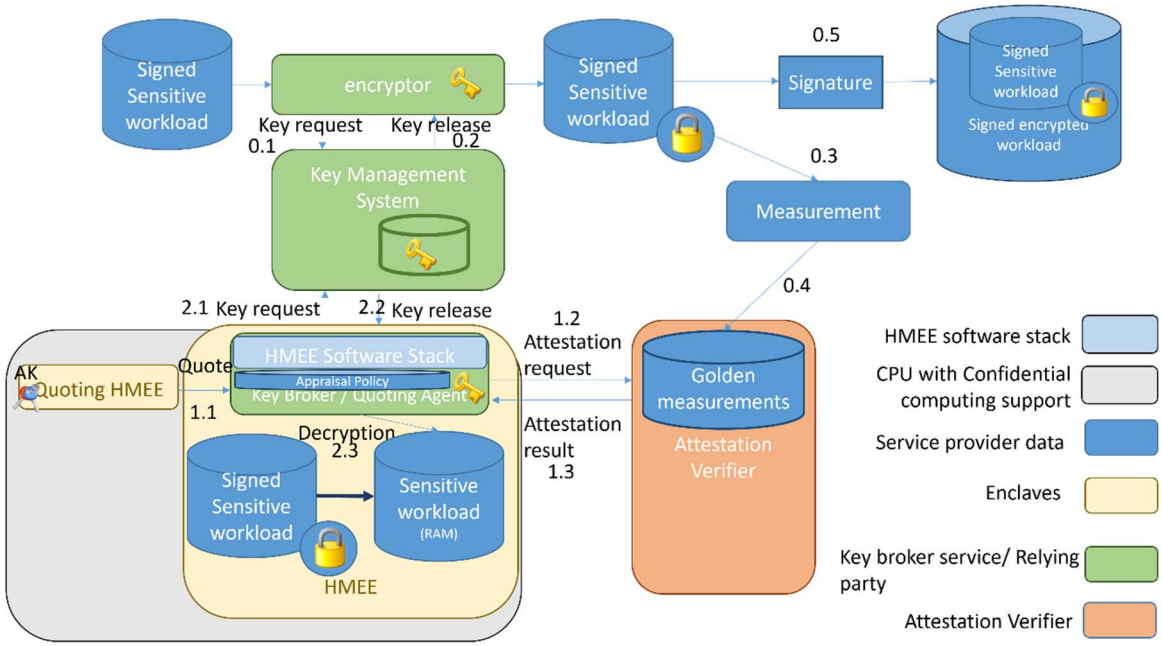

Figure 5.3.5-1: Architectural view for encryption/decryption of sensitive workloads

NOTE 2: The validation of the signature of the encrypted workload, that is done during the instantiation is not represented here.

Description of the functional entities:

- Service Provider data:
  - The sensitive workload, containing the application that shall be protected using an encryption and that shall be instantiated in a HMEE. The sensitive workload has been signed by the VNF provider.
  - The encrypted sensitive workload, protected in integrity and authenticity with a signature.
  - The measurement of this encrypted sensitive workload. The output of this measurement constitutes the golden measurement used by the Attestation Server as appraisal policy for evidence as defined in IETF RFC 9334 [2] during the attestation verification.
  - The appraisal policy used by the relying party (i.e. key broker service agent) to appraise the attestation result as described in IETF RFC 9334 [2].
- The attestation verifier that verifies against the golden measurements the attestation quote provided by the HMEE as defined by IETF RFC 9334 [2].

NOTE 3: The attestation verifier is the entity on which all the trust relies on and use the golden measurements of service provider. The service provider needs to trust the attestation verifier. There will be an implicit trust if the attestation verifier is in the trust domain of the service provider as shown in table 5.3.5-1 below. How the attestation verifier verifies the quote of a system in a multi-tenant environment needs further work.

- The key broker service:
  - An encryptor that encrypts the sensitive workload using a key provided by a Key Management System.
  - A Key Management System including a tamper resistant key storage (e.g. an HSM).

- An agent including the HMEE Software Stack, with the functionality of quoting agent requesting the verification of the quote generated by a Quoting HMEE, and the functionality of key broker allowing the decryption of the sensitive workload with the decryption key provided by the Key Management System if the attestation is successful and conforms to the appraisal policies. This key broker agent is a relying party as described by IETF RFC 9334 [2].

NOTE 4: These entities composing the key broker service should be trusted by the Service Provider. There will be an implicit trust if the key broker service is in the trust domain of the service provider as shown in table 5.3.5-1 below.

- The HMEE software stack:
  - The HMEE software stack contains the virtual firmware/BIOS and Libraries integrated in the HMEE to run the sensitive applications and providing the APIs that abstract the communication with the CPU HW layer to request the quote and to abstracts the communication layer with the quoting HMEE that provides the signed quote.
- The CPU:
  - The CPU used in this architecture supports the confidential computing functionality either using VM-Based HMEE (e.g. AMD SEV-SNP, Intel TDX, ARM CCA) or using Process-based HMEE (e.g. Intel SGX).
  - A quoting HMEE (e.g. Trust Domain Quoting Enclave for Intel TDX, vTPM). This quoting HMEE signs the quote with the private key part of the Attestation Key, that enables the attestation server to verify that the quote has been provided by a genuine confidential computing CPU.

The workflow depicts by this architecture figure 5.3.5-1 is divided in 3 phases:

- Phase 0: The preparation of the sensitive workload before the start of its journey in the cloud. In this phase the sensitive workload is encrypted and measured, and signed.
- Phase 1: The attestation process.
- Phase 2: the decryption of the sensitive workload in the HMEE.

Phase 0:

- Pre-condition: the sensitive workload has been signed by the VNF provider.
- Step 0.1: the encryptor requests the encryption key to the Key Management System (KMS) using the standard protocol OASIS KMIP [3].
- Step 0.2: The KMS provides an encryption key for the signed sensitive workload.
- Step 0.3: The encrypted workload and the stack software including the agent are measured (hash).
- Step 0.4: the measurements are included in the Attestation server policies as golden measurement.
- Step 0.5: the encrypted sensitive workload is signed by the service provider for integrity and authenticity protection.

Phase 1:

- Pre-condition: The validation of the signature of the encrypted workload is processed during the instantiation of the encrypted sensitive workload.
- Step 1.0: Initialisation step: The quoting agent in the HMEE generates a key pair and exchanges the public key with the key management system.
- Step 1.1: The quoting agent in the HMEE request a nonce (nonce-ats) to the attestation server and request a nonce (nonce-kms) to the key management system. These nonces will be used for anti-replay protection. The quoting agent in the HMEE request the quote to the HW CPU and receives the quote report signed by the Quoting HMEE. The quoting agent in the request for the quote to the HW CPU may include the two nonces, nonce-ats and nonce-kms, as report data to ensure the freshness of the quote.

- Step 1.2: the quoting agent in the HMEE sends the attestation request including the signed quote.
- Step 1.3: the attestation server verifies the authenticity of the signed quote using the attestation public key, the freshness of the signed quote with the nonce-ats, and verifies the measurements against the golden measurements. The attestation server sends back the attestation result that includes the nonces to the HMEE quoting agent.

Phase 2:

- Step 2.1: after successful verification of the attestation result and conformance to appraisal policies, the Key broker agent requests the decryption key for the encrypted sensitive workload using OASIS KMIP [3]. the Key broker agent includes in the request the attestation result including the nonces.
- Step 2.2: The KMS verifies with the nonce-kms the freshness of the attestation result and provides to the key broker service in the HMEE, the decryption key wrapped with the public key of the key broker service in the HMEE and received during the initialization step 1.0. The key broker service in the HMEE de-wraps the key decryption key using its private key.
- Step 2.3: the encrypted workload is decrypted using this decryption key, the signature of the sensitive workload is validated, and if successful the application is launched.

For this architecture a trust relationship exists between the different actors.

Table 5.3.5-1 below shows the trust domain of the entities described in figure 5.3.5-1.

Table 5.3.5-1: Trust domain of the entities described in figure 5.3.5-1

Entity in the trust domain of	Infrastructure Provider	Service / KMS provider	Comments
CPU	X		
HMEE software stack		X	The key broker agent processing sensitive data (e.g. key) includes the OS and libs to control the security of the service.
Key broker service		X	The Key broker service processing service provider's data is owned by the service provider and in the service provider trust domain to ensure an implicit trust.
Attestation verifier		X	The attestation verifier using sensitive service provider's data is owned by the service provider and in the service provider trust domain to ensure an implicit trust.
Service provider data		X	These data are owned by the service provider.

Figure 5.3.5-2 below gives the trust relationship between the different actors. It is to be noted that there is no need of trust relationship between the Infrastructure Provider (i.e. Cloud service provider) and the Attestation Verifier or the Key Provider, showing the separation of duties.

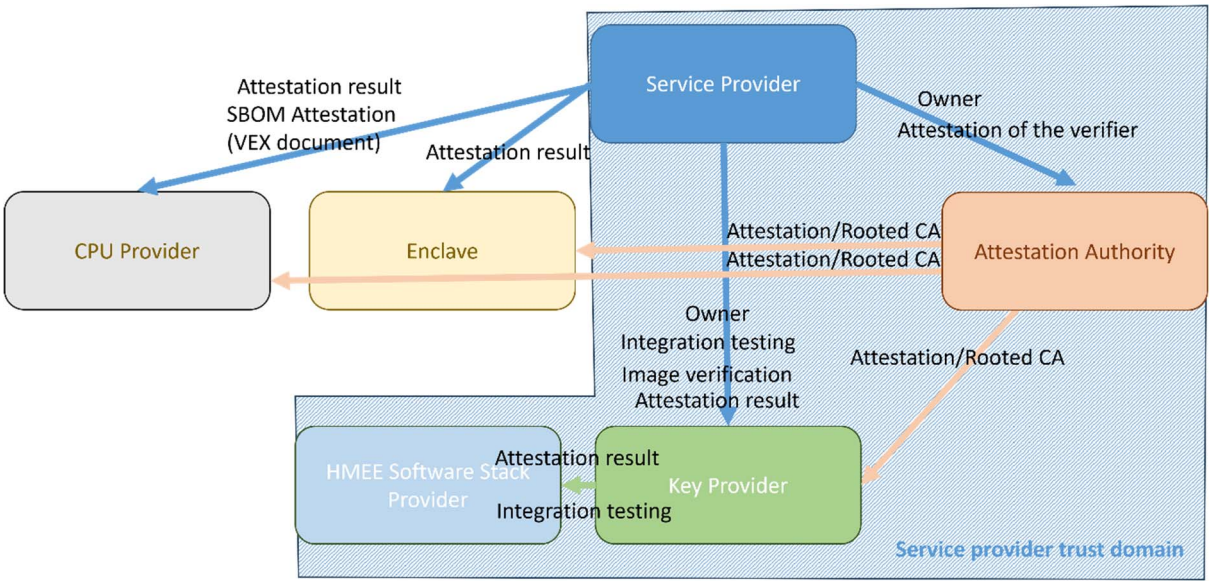

Figure 5.3.5-2: Trust relationship between actors

The Key Broker Service of the architecture presented in figure 5.3.5-1 and integrated in the HMEE could be also a Key Broker Service included in an external server connected to the KMS, acting as a relying party as defined in IETF RFC 9334 [2], as presented in figure 5.3.5-3 below.

NOTE 5: The colours used in the figure 5.3.5-3 do not represent trust domains but the different functionalities.

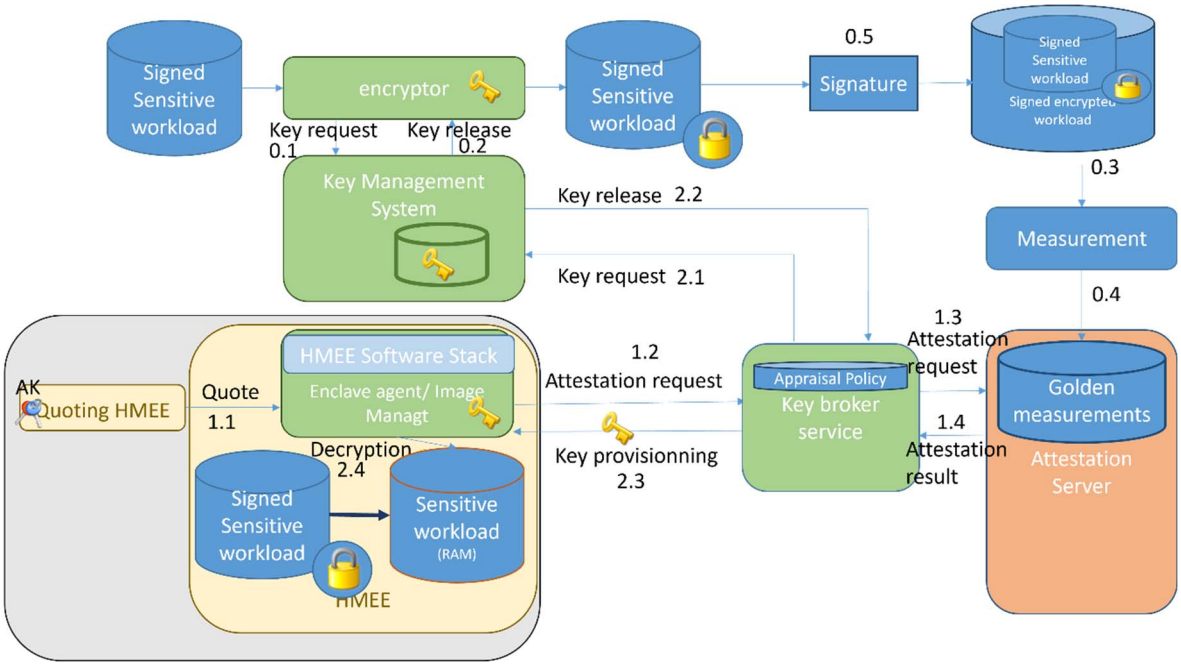

Figure 5.3.5-3: Architectural view with external Key Broker Service

The trust relationship for this architecture is similar to the trust relationship of the architecture depicted in figure 5.3.5-1.

6 Hypervisor partitioning

6.1 Introduction

Hypervisor partitioning refers to the logical division of a physical server into multiple isolated environments, known as partitions. Each partition operates as if it were a separate physical machine, running its own operating system and applications.

Isolation shall be at the same time a spatial, temporal and fault isolation:

- Spatial isolation or memory isolation is the ability to isolate code and data of a partition from the others. This prevents data and code alteration or eavesdropping.
- Temporal isolation is the ability to isolate the impact of the usage of a resource (e.g. CPU or memory) of a certain partition on the decay of the performance of another partition and avoid starvation or throughput reduction.
- Fault isolation prevents any failure in one partition from causing failures in another partition or impacting the overall system throughput.

The partitioning hypervisors, based on the separation kernel, are the virtualization solutions that best fit the concept of partitioning and are specifically designed to ensure isolation.

6.2 Solution description

6.2.0 Introduction

This clause describes the partitioning mechanisms that shall be used by partitioning hypervisors to ensure the different isolations: spatial, temporal and fault isolation.

6.2.1 Spatial partitioning

6.2.1.0 Introduction

Partitioning hypervisors shall use several spatial isolation mechanisms to ensure that different partitions (or virtual machines) do not interfere with each other.

These mechanisms help maintain strong isolation properties, which are crucial for security and safety in systems that run multiple applications with different criticality levels on the same hardware platform.

6.2.1.1 Memory Partitioning

This involves dividing the physical memory into separate regions, each allocated to a different partition.

An access control to user-space memory, by restricting accesses, ensures that one partition cannot access the memory of another. With the control of writes, memory integrity and with the controls of reads, the memory confidentiality is ensured between two or more different partitions.

In the same way, an access control to the kernel-space memory, containing for example the management data, ensures the separation of the kernel space memory partitions.

6.2.1.2 Memory partition encryption

To ensure confidentiality of the data at rest in the memory partitions, the mechanism of transparent encryption of data shall be used. As described by figure 6.2.1.2-1, each partition (or VM) is encrypted using a short-term Data Encryption Key (DEK) in the hypervisor. A long-term Key Encryption Key (KEK) protects the DEKs stored within the hypervisor. The KEK shall be stored in a secure storage such as HSM and access to these KEKs controlled by a key management system. The encryption and access control may apply on a per-VM basis.

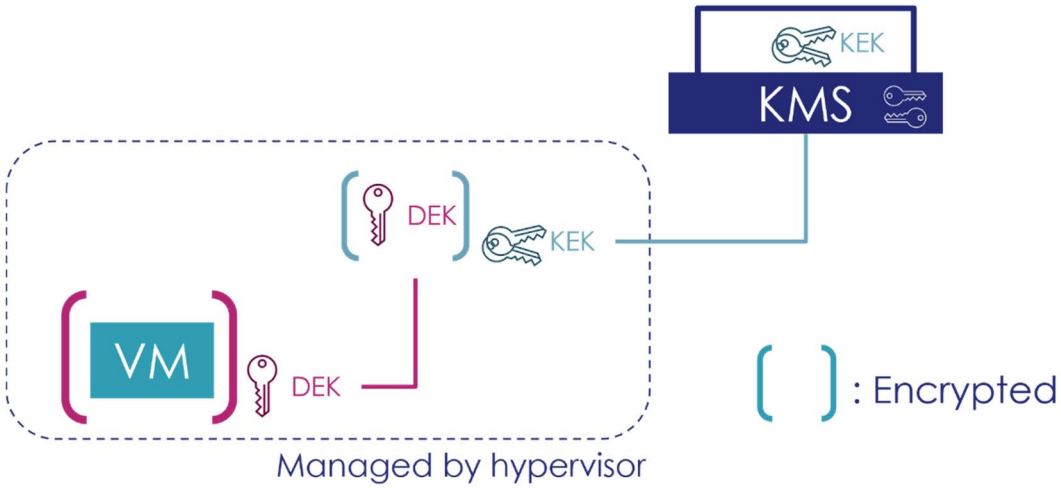

Figure 6.2.1.2-1: VM transparent encryption

6.2.1.3 Cache Partitioning

By using techniques like cache coloring, the hypervisor can ensure that each partition uses a distinct portion of the cache, preventing cache-based side-channel attacks.

6.2.1.4 I/O Partitioning

This mechanism isolates I/O devices, ensuring that each partition has exclusive access to certain devices or that access is controlled and mediated by the hypervisor.

6.2.1.5 CPU Partitioning

The hypervisor can allocate specific CPU cores or time slices to different partitions, ensuring that the execution of one partition does not affect the performance of another.

6.2.1.6 Hardware resources Partitioning

The hardware resources may be completely separate among partitions using kernel configurations. If the resources are shared, their use are allocated to each partition by time windows.

6.2.1.7 Key vault Partitioning

A key vault is a service used to manage and store cryptographic keys, secrets, and certificates. It acts as a secure storage mechanism that enables users to control access to these sensitive items. An HSM is a physical device that includes the functionality of a key vault, managing, storing and handling cryptographic keys but with a focus of high security equipped with tamper-resistant features and provide protection against unauthorized access and attacks. Additionally, HSMs are used for secure key generation, and to perform cryptographic operations (such as encryption, decryption, signing, and verification) in a secure environment. HSMs offer a higher level of security by ensuring that keys never leave the module in plaintext. If the key vault is a software-based solution, it is more vulnerable to certain attacks. This can be mitigated to a certain extent by using hardware-backed secure storage (e.g. an HSM).

Each partition may use keys that shall be protected in a secure key vault such as HSM, virtual HSM (vHSM) may be used to provide the HSM functionality for each partition, using a single or several physical HSM. Each key vault is isolated between each other with distinct access control, cryptographic domains, policies, memory and time separation managed by a Key Management System. The isolation shall be done on a per-key vault basis and may be a logical separation or a physical separation where dedicate hardware resource are allocated per key vault.

6.2.1.8 Static Memory allocation

Fixed quotas of memory are allocated to each partition by the memory management, to avoid the depletion of storage space and its unavailability.

6.2.1.9 Interrupt Partitioning

The hypervisor manages interrupts in a way that ensures each partition only receives its relevant interrupts, preventing interference from other partitions.

6.2.2 Temporal partitioning

6.2.2.0 Introduction

Partitioning hypervisors use several temporal isolation mechanisms to ensure that the execution timing of one partition does not interfere with another.

These mechanisms help maintain strong temporal isolation, which is crucial for systems with mixed-criticality applications where timing predictability is essential.

6.2.2.1 Fixed Cyclic Scheduling

This involves assigning fixed time slots to each partition, ensuring that each partition gets a predictable amount of CPU time without interference from others.

6.2.2.2 CPU registers reuse

A temporal partitioning of CPU register between partitions makes residual information unavailable on context switches that enforces the confidentiality.

6.2.2.3 Memory Bandwidth Reservation

By reserving a specific portion of the memory bandwidth for each partition, the hypervisor ensures that partitions do not compete for memory access, which helps in maintaining predictable performance.

6.2.2.4 Priority-based Scheduling

The hypervisor may prioritize partitions based on their criticality, ensuring that higher-priority partitions get access to CPU and other resources before lower-priority ones.

6.2.2.5 Worst-case execution time

The hypervisor may ensure an upper bond of the worst-case execution time for each critical execution path in the system. This enforces readiness and availability, preventing a task from stalling another task.

6.2.2.6 Interrupt Management

Proper management of interrupts ensures that each partition only receives its relevant interrupts, preventing unnecessary delays caused by irrelevant interrupts.

6.2.2.7 Resource Reservation

The hypervisor may reserve specific hardware resources (like CPU cores or I/O devices) for certain partitions, ensuring that these resources are not shared or contended.

6.2.3 Fault partitioning

6.2.3.0 Introduction

Partitioning hypervisors use several fault isolation mechanisms to ensure that faults in one partition do not affect other partitions.

These mechanisms help maintain strong fault isolation, which is crucial for systems with mixed-criticality applications where reliability and safety are paramount.

6.2.3.1 Resource Isolation

By isolating resources such as CPU, memory, and I/O devices, the hypervisor ensures that faults in one partition do not impact the resources of another partition.

6.2.3.2 Memory Protection

The hypervisor enforces memory protection policies to prevent one partition from accessing the memory space of another partition, which helps contain faults within individual partitions.

6.2.3.3 Error Detection and Handling

The hypervisor may implement error detection mechanisms, such as parity checks and ECC (Error-Correcting Code), to identify and handle faults within partitions.

6.2.3.4 Fault Containment

In the event of a fault, the hypervisor may isolate and contain the fault within the affected partition, preventing it from spreading to other partitions.

6.2.3.5 Redundancy and Checkpointing

The hypervisor may use redundancy and checkpointing techniques to recover from faults within partitions, ensuring that the system can continue to operate even if a partition experiences a fault.

7 Containerized VNF escape protection

7.1 Introduction

Container escape is a critical security concern in containerized environments. It occurs when an application or process running inside a container gains unauthorized access to resources outside the container, potentially compromising the host system and other containers.

The following clause gives some mitigations solutions for these container escape attacks.

7.2 Solutions description

To mitigate the risks associated with container escape, several protection strategies can be employed and are listed below. By adopting these measures, organizations can significantly enhance the security of their containerized environments and protect against potential escape attempts.

Vulnerability Management

The management of vulnerabilities within container images is a key component of preventing container escape. Vulnerability management can include first four controls below.

Use trusted sources for base images

The first step for container security is to use trusted image (including the sub-components of this image) coming from a trusted source, i.e. known company or open-source group and hosted on a reputable registry.

Regularly update container images to address known vulnerabilities

An image scanner shall be used to scan all images of the enterprise's registry on a regular cadence. Attribute-Based Access Control to this enterprise's registry shall be in place.

Runtime Image Scanning

Runtime scanning is important to check if image contains newly discovered security vulnerabilities. Vulnerabilities management controlled by policies.

Image Registry - Image controls

It is important to know the version of the image that is deployed at a given time. The containerized VNF/VNFC image shall be signed to have a means to check the integrity of the images during the whole process of deployment until the instantiation.

An attestation shall be done on the image during the instantiation to check the integrity of the image at very last stage of the image deployment.

Namespace isolation

Containers should have isolated namespaces to ensure they cannot access each other's resources or the host system's resources.

Capabilities Limitation

Reducing the capabilities of containers to the minimum required for their operation is an essential principle. This limits the potential damage if a container is compromised.

Reducing the container image to the strict minimum to execute the function for which they have been deployed is a best practice to restrict the possible exploitation of an attacker once it succeeds to get in. this implies to restrict the image to the only required binaries, libraries and configuration files.

A container should not run with elevated privileges capability. Any use of elevated privileges shall be risk assessed and other mitigations put in place.

Resources Limitation

Setting resource limits (CPU, memory, etc.) to prevent a compromised container from exhausting host resources is required.

The use of control groups (cgroups) enables the control and limitation of resources consumed by a container.

SysCall limitation

To reduce the surface of attacks, reducing the possible system calls from the container is required.

Seccomp is a tool that allows administrators to define system call security that shall be blocked during container runtime.

Read-Only Filesystems

To prevent unauthorized modifications to the container's file system, the use of read-only filesystems for the container is a best practice.

Access Control

Other security tools such as AppArmor and SELinux may be used to enforce mandatory access controls based on security policies and further isolate containers.

Encryption of the data at rest issued from containers

An encryption of the data issued by the container and using a dedicated key provided by a key management system is also a mean to protect against unauthorized access to the container data.

This transparent encryption of data is described in more details in clause 5.3.3 of the present document.

Monitoring and Logging

Implementation of a robust monitoring and logging tool is also important to detect and respond to suspicious activities promptly.

These tools may include a control of the communication between containers based on network policies and can limit exposure to potential attacks. Some tools may also use an AI functionality for the detection of communication anomalies.

8 Key Management and access control

8.1 Introduction

Key management refers to the administrative control over cryptographic keys across their life cycle. It encompasses their creation, distribution, storage, inventory, and deletion. This process ensures that the key materials used for cryptographic operations are securely handled and accessible only to authorized entities. Effective key management is fundamental to maintaining the integrity and confidentiality of sensitive information.

Clause 8 of the present document describes best practices for effective key management to ensure securing sensitive data and maintaining trust.

8.2 Strong cryptographic algorithms

Cryptographic keys shall be generated with a high degree of randomness to ensure their unpredictability and avoid the use of keys that are easy to guess. The quality of the random generation process shall be certified FIPS or other equivalent standard and should generate enough entropy to guarantee uniqueness, utilizing specialized hardware where required.

Manual key generation processes, that are prone to errors and inconsistencies shall not be used.

Generating cryptographic keys using weak algorithms is a significant security risk. This can result in keys that are too predictable or easy to crack, making it easier for bad actors to gain unauthorized access to sensitive data.

Another best practice concerning key generation is to use robust well-established cryptographic algorithms, widely recognized for their security and efficiency.

It is also critical to support strong key usage/typing control. The key attributes such as its usage, its export restrictions and its validity shall be protected with strong cryptographic mechanisms.

Quantum computing presents a distinct threat. The general rule is that both mathematics, algorithmic and silicon technological progress require continuous upgrade of key mechanisms and length. A modern key management platform should support crypto agility, or the ability to introduce new or upgraded key mechanisms, and shall have weak mechanisms or length deprecated. This requires flexible and adaptable key management infrastructure that integrates new cryptographic methods without disrupting existing operations.

8.3 Secure key storage

A critical part of key storage is that it shall be under strict access control, with proper auditability, and keys at rest shall always be encrypted with the strongest encryption mechanism.

To protect the keys, Hardware Security Modules (HSMs) or equivalent mechanism compliant with FIPS 140-3 shall be used. Key storage shall:

- Optimally comply with FIPS 140-3 level 3 or CC EAL4+.
- Minimally comply with FIPS 140-3 level 1, with a Root of Trust (RoT) which complies with FIPS 140-3 level 3 or CC EAL4+.

These mechanisms provide a secure environment for generating, storing, and managing cryptographic keys. They offer tamper-resistant protection and compliance with stringent security standards.

Trusted providers may offer managed key management services or cloud HSMs relying on physical hardware HSMs that provide high levels of security and compliance.

When storage of keys is not practical into hardware (e.g. the number of keys is too great, or to meet high availability requirements in a highly distributed setup), keys can be stored outside the HSM as a secure blob, but the use of these keys shall only be allowed in certified hardware.

8.4 Separation encrypted data and encryption keys

The separation of encrypted data and encryption keys is a foundational principle of effective key management. This prevents a malicious actor from exfiltrating encrypted data and the associated keys, and being able to decrypt the data.

8.5 Regular key rotation

The longer a key remains in use and the larger number of times this key is used, the greater the chance it will be exposed or compromised. Over time, the likelihood of a key being discovered or leaked increases. Additionally, cryptographic algorithms and key lengths that were once secure may become vulnerable over time due to advances in computational power and cryptographic attacks. In some cases, a key shall be deprecated because it is no longer secure against the latest threats or following a specific security incident.

Implementing regular key rotation policies minimizes the risk associated with long-term key exposure. Automated key rotation, supported by robust policy frameworks, ensures that keys are periodically updated without disrupting operations. This practice helps mitigate the potential impact of a compromised key. In addition, limiting the number of messages encrypted with the same key helps prevent attacks enabled by cryptanalysis. Implementing a crypto-agile platform that can streamline key configuration is critical.

8.6 Access controls and auditing

Anyone with access to cryptographic keys can intentionally or unintentionally expose them. Using third-party services or cloud providers without ensuring they have adequate security measures can expose keys to bad actors.

Access management enables determining whether a user has permission to access a certain resource and enables the enforcement of the access policy that has been set up for that resource. Access management is implemented based on policies that are defined by administrators. Key access can be very granular. Access policies can restrict the ability to create, delete, revoke, renew keys and be based on various attributes.

Strict access controls ensure that only authorized individuals can access cryptographic keys. Implementing Attribute-Based Access Controls (ABAC), Role-Based Access Controls (RBAC), Identity-Based Access Control (IBAC) and Multi-Factor Authentication (MFA) can significantly reduce the risk of unauthorized access. Regular auditing of key usage and access logs is important for detecting and responding to anomalous activities.

In case of mechanisms that deliver in-container or in-VM capabilities of encryption (transparent encryption), access controls and data access logging shall be used to establish strong safeguard around data. This data access control shall ensure that users (e.g. docker, cluster administrator) operate as non-privileged users without gaining unauthorized access to sensitive data. Encryption, access controls and data access logging apply at per-VM or per-container basis. This enables a single policy to be applied to all containers (or VMs) within a cluster, or distinct policies to be applied to each container (or VM).

8.7 Disaster recovery and high availability

Disaster recovery plans for key management should include procedures for key back-up, secure storage of backup keys and swift key recovery processes to minimize downtime and data exposure.

In case a node is down, keys should be distributed automatically by the key management system to the multiple nodes of a cluster to prevent down time and ensure high availability.

8.8 Key management approaches

8.8.1 Introduction

Several key management approaches may be chosen depending on the needs and constraints of the system. There are several approaches:

- Bring Your Own Key (BYOK).
- Hold Your Own Key (HYOK).
- Bring Your Own Encryption (BYOE).

8.8.2 Bring Your Own Key (BYOK)

BYOK allows companies to generate and manage their own encryption keys, even when using cloud services. This approach provides better control over proprietary data and benefits compliance by ensuring that keys are managed according to specific regulatory requirements.

This key management approach enables an enhanced control for sensitive data.

8.8.3 Hold Your Own Key (HYOK)

HYOK allows users full custody of their encryption keys, ensuring more control over data security.

This key management approach enables more control over encryption keys and data security, with lower vendor lock-in, that reduces the dependency on a single vendor.

8.8.4 Bring Your Own Encryption (BYOE)

BYOE involves encrypting data before it is sent to a cloud provider. Organizations can maintain control over their data security by enabling them to use their own encryption keys and algorithms within cloud services and third-party applications, ensuring that only authorized users can access sensitive information.

This key management approach ensures that data always remains encrypted and retains encryption control even in un-trusted environments.

8.8.5 Key management approaches selection

The key management approach to be used depends on the data sensitivity, the regulatory requirements, the availability of technical expertise to manage complex key management, and the use in cloud services for which there is a need for scalable, flexible key management solutions.

Figure 8.8.5-1 below shows the key management approach to be used depending on data portability, the level of separation of duties and auditability and the level of Service Provider control needed.

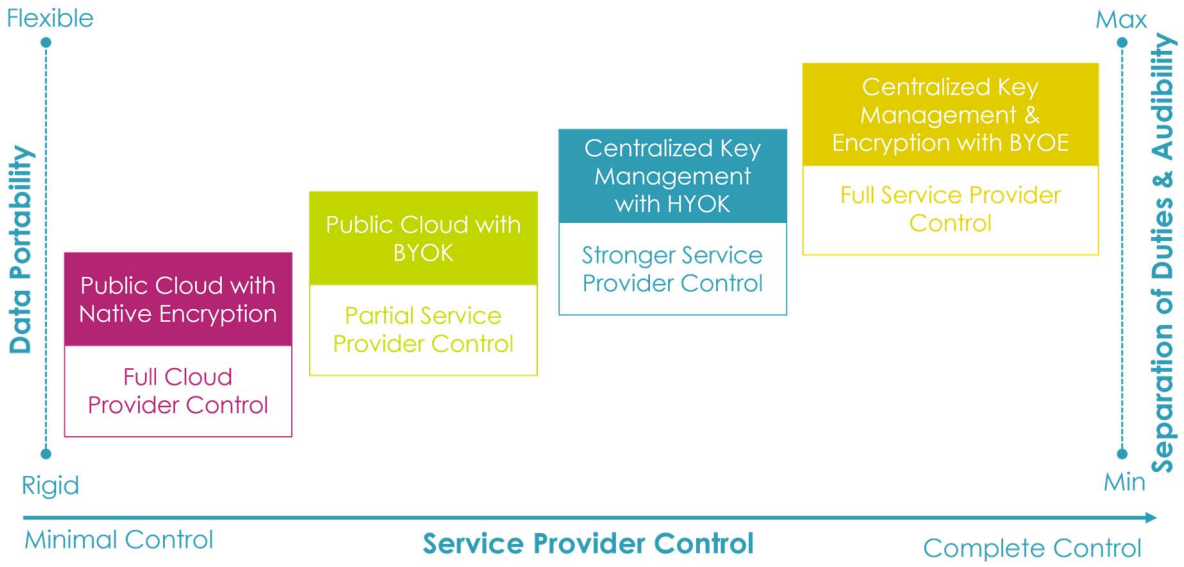

Figure 8.8.5-1: Data portability, separation of duties, auditability and Service Provider control on keys depending on key management approaches

8.9 Centralized Key management

Centralized key management offers significant business benefits by providing consistent policies, secure storage, and comprehensive auditing capabilities:

Regular key inventory and compliance checks: Regular key inventory and compliance checks should be performed to ensure active keys meet the latest security requirements in term of mechanisms and length and if that is not the case to keep track of their rotation with stronger options and their eventual deprecation. This operation is easier in case of centralized key management.

Consistent Key generation and rotation: Centralized key management systems enforce standardized procedures for creating and updating keys, with strong key generation, regular key rotation and enforcing uniform and consistent policies. Policies should include selection of key mechanisms and lengths that are deemed secure at a given time.

Secure storage: Centralized key management systems use advanced storage solutions such as remote Hardware Security Modules (HSM) to protect keys from unauthorized access and theft.

Enforcement of access controls: Implementing centralized access policy enforcement, where the system collects all relevant information in a single place for easy audit and in human-readable form makes demonstrating compliance with internal and external policies a more straightforward task.

Data sovereignty: Centralized key management can ensure compliance with local data residency laws and regulations, by enabling organisations to select the physical or geographical location in which the keys are stored. Approaches such as HYOK or BYOK also allow companies to maintain control over their encryption keys, providing them with the necessary sovereignty over their data.

Regulatory compliance: Centralized key management systems facilitate compliance by:

- Offering comprehensive auditing and reporting capabilities, providing detailed records of key management activities essential for demonstrating compliance during audits and regulatory reviews.
- Using automated tools for compliance checks.
- Enforcing security policies aligned with regulatory requirements and ensuring key management practices meet the necessary standards.

Centralized auditing of key usage: Centralized key management boosts auditing capabilities by:

- Maintaining comprehensive logs of all key-related activities, including key generation, access, rotation, and decommissioning.

- Enabling real-time monitoring of key usage and detecting and responding to suspicious activity promptly.
- Facilitating analysis during and after a security incident.

9 Conclusion

The present document, based on use-cases developed in ETSI GR NFV-EVE 018 [i.1], presents an analysis of the security threats induced by the multi-tenancy in a NFV environment. A list of requirements to mitigate these threats are proposed, and technical solutions to implement these requirements and improve the isolation of the NFV system are described.

The present document shall be used for further NFV-IFA specifications and NFV-SOL specifications where multi-tenancy is involved and when NFV systems are designed with a multi-tenancy support.

Annex A (informative): Change history

Date	Version	Information about changes
04-2020	v001	First draft as baseline
12-2020	V002	Implementation of the following contribution accepted during the SEC#179 meeting: - NFVSEC(20)000114r2_SEC026_Threat_Analysis_UC_1
02-2021	V003	Implementation of the following contribution accepted during the SEC#180 and SEC#181 meetings: - NFVSEC(21)000004r1_SEC026_Additional_threat_and_requirements_UC_1 - NFVSEC(21)000003r2_SEC026_Threat_Analysis_UC_2 - NFVSEC(21)000010r1_SEC026_Threat_Analysis_UC_3
04-2021	V004	Implementation of the following contribution accepted during the SEC#185 meeting - NFVSEC(21)000034r3_SEC026_Threat_Analysis_UC_4
10-2021	V005	Implementation of the following contribution accepted during the SEC#195 meeting - NFVSEC(21)000067r5_SEC026_Threat_Analysis_UC_7
09-2022	V006	Implementation of the following contribution accepted during the SEC#215 meeting - NFVSEC(22)000087_SEC026_Threat_Analysis_UC_8
01-2023	V007	Implementation of the following contribution accepted during the SEC#221 meeting - NFVSEC(22)000088r1_SEC026_Threat_Analysis_UC_9
12-2023	V008	Implementation of the following contributions accepted during the SEC#245 meeting - NFVSEC(23)000246_SEC026_Threat_Analysis_UC_6 - NFVSEC(23)000248r1_SEC026_section_5_1 - NFVSEC(23)000249r1_SEC026_section_5_2 - NFVSEC(23)000250r1_SEC026_section_5_3_1 - NFVSEC(23)000251r2_SEC026_section_5_3_2 - NFVSEC(23)000252r1_SEC026_section_5_3_3 - NFVSEC(23)000254r1_SEC026_section_5_3_5 Change "Kubernetes" to "Kubernetes® "
01-2024	V009	Implementation of the following contribution accepted during the SEC#246 meeting - NFVSEC(23)000253r1_SEC026_section_5_3_4
02-2024	V010	Implementation of the following contribution accepted during the SEC#248 meeting - NFVSEC(24)000018r1_SEC026_section_5_3_4_separation_of_duties
03-2024	V011	Implementation of the following contribution accepted during the SEC#251 meeting – NFVSEC(24)000026r1_SEC026_section_5_3_3_architecture
06-2024	V012	Implementation of the following contribution accepted during the SEC#263 meeting - NFVSEC(24)000020r7_SEC026_section_5_3_5_architecture - NFVSEC(24)000021r7_SEC026_section_5_3_4_architecture
10-2024	V013	Implementation of the following contribution accepted during the SEC#272 meeting - NFVSEC(24)000185r1_SEC026_section_7
12-2024	V014	Implementation of the following contributions accepted during the SEC#277 meeting - NFVSEC(24)000209r2_SEC026_section_6 - NFVSEC(24)000214r1_SEC026_use-case__7_additions - NFVSEC(24)000215r1_SEC026_section_8
02/2025	V015	Implementation of the following contributions accepted during SEC#279 and SEC#281: - NFVSEC(25)000009_SEC026_deletion_editor_s_note - NFVSEC(25)000010_SEC026_section_9 - NFVSEC(25)000014_SEC026_deletion_editor_s_note_5_3_2
03/2025	V016	Implementation of the following contribution accepted during SEC#283 - NFVSEC(25)000021r1_SEC026_Removal_of_references_to_SEC023_and_SEC025
06/2025	V017	Implementation of contribution NFV(25)000076 as well as some editorial changes.

History

Document history		
V5.1.1	July 2025	Publication