# ETSI GR NFV-SEC 005 V1.1.1 (2019-01)


# Network Functions Virtualisation (NFV);
# Trust;
# Report on Certificate Management

***Disclaimer***

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

Reference
DGR/NFV-SEC005

Keywords
certificate, NFV, security

***ETSI***

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

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

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

# Contents

Intellectual Property Rights ..... 5
Foreword ..... 5
Modal verbs terminology ..... 5
1 Scope ..... 6
2 References ..... 6
2.1 Normative references ..... 6
2.2 Informative references ..... 6
3 Definition of terms, symbols and abbreviations ..... 7
3.1 Terms ..... 7
3.2 Symbols ..... 7
3.3 Abbreviations ..... 7
4 Rationale and approach for the use of public key certificates ..... 7
4.1 Scope ..... 7
4.2 PKI Participants ..... 8
4.2.0 Introduction ..... 8
4.2.1 Certificate Authorities ..... 8
4.2.2 Registration Authorities ..... 9
4.2.3 Subscribers ..... 9
4.2.4 Relying Parties ..... 9
4.2.5 Auditors ..... 9
4.3 Mapping of secure relationships to NFV reference points ..... 9
4.4 Use Cases for the use of certificates in NFV ..... 11
4.5 Considerations for PKC validation ..... 12
4.5.1 Certificate path building and chain validation ..... 12
5 Use cases for the use of certificates in NFV ..... 13
5.1 VNF certificate use case ..... 13
5.1.0 Introduction to use cases ..... 13
5.1.1 Use case #1: VNF management connection ..... 13
5.1.2 Use case #2: VNF transport connection ..... 14
5.2 MANO certificate use case ..... 15
5.3 OSS/BSS/EM certificate use case ..... 15
6 Recommendations ..... 15
6.1 General ..... 15
6.2 Deployment recommendations ..... 15
7 Certificate management framework ..... 21
7.1 Certificate hierarchy ..... 21
7.2 Certificate category ..... 22
8 NFV certificate lifecycle management ..... 22
8.1 Certificate generation ..... 22
8.1.1 Initial Credential ..... 22
8.1.1.1 Key pair generation ..... 22
8.1.1.1.1 Option 1: NFVI generates key pair ..... 22
8.1.1.1.2 Option 2: HMEE generates key pair ..... 23
8.1.1.1.3 Option 3: HSM generates key pair ..... 24
8.1.2 VNFCI Certificate ..... 26
8.1.2.0 Introduction to VNFCI certificate issuance ..... 26
8.1.2.1 Option 1: VNFCI generates key pair, constructs and signs certificate request ..... 26
8.1.2.2 Option 2: VNFCI generates key pair, constructs certificate request, and VNFM/NFVO signs certificate request ..... 27
8.2 Certificate update ..... 28
9 NFV Certificate Management ..... 28

9.0 Introduction ..... 28
9.1 MANO and other functional blocks ..... 29
9.2 Tenant domain ..... 29
9.2.1 VNF certificate ..... 29
9.2.1.0 Introduction ..... 29
9.2.1.1 ID and certificate management in VNF ..... 29
9.2.1.2 Certificate lifecycle and VNF lifecycle ..... 33
9.2.1.3 VNF instantiation ..... 33
9.2.1.4 VNF scaling ..... 34
9.2.1.5 VNF migration ..... 34
9.2.1.6 VNF update/Upgrade ..... 34
9.2.1.7 VNF termination ..... 35
9.3 Certificate Provisioning ..... 35
9.4 Trust list management ..... 36

**Annex A: Authors & contributors ..... 37**

History ..... 38

# Intellectual Property Rights

## Essential patents

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

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

## Trademarks

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

# Foreword

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

# Modal verbs terminology

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

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

# 1 Scope

The present document provides guidance to the development community on the use of Public Key Certificates, Attribute Certificates and the supporting infrastructure, including Registration Authorities, and Certificate Authorities. The present document provides this guidance in the context of a number of use cases and references to other publications of ETSI ISG NFV.

# 2 References

## 2.1 Normative references

Normative references are not applicable in the present document.

## 2.2 Informative references

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

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

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

[i.1] ETSI GS NFV 001: "Network Functions Virtualisation (NFV); Use Cases".

[i.2] ETSI GS NFV 002: "Network Functions Virtualisation (NFV); Architectural Framework".

[i.3] ETSI GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[i.4] ETSI GS NFV 004: "Network Functions Virtualisation (NFV); Virtualisation Requirements".

[i.5] ETSI GS NFV-MAN 001: "Network Functions Virtualisation (NFV); Management and Orchestration".

[i.6] ETSI GS NFV-SWA 001: "Network Functions Virtualisation (NFV); Virtual Network Functions Architecture".

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

[i.8] ETSI GS NFV-SEC 001: "Network Functions Virtualisation (NFV); NFV Security; Problem Statement".

[i.9] ETSI GS NFV-SEC 003: "Network Functions Virtualisation (NFV); NFV Security; Security and Trust Guidance".

[i.10] ETSI GS NFV-SEC 009: "Network Functions Virtualisation (NFV); NFV Security; Report on use cases and technical approaches for multi-layer host administration".

[i.11] ETSI GS NFV-SEC 012: "Network Functions Virtualisation (NFV) Release 3; Security; System architecture specification for execution of sensitive NFV components".

[i.12] IETF RFC 4809: "Requirements for an IPsec Certificate Management Profile".

[i.13] Recommendation ITU-T X.509: "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks".

[i.14] IETF RFC 3647: "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework".

[i.15] IETF RFC 8446: "The Transport Layer Security (TLS) Protocol Version 1.3".

[i.16] IETF RFC 7030: "Enrollment over Secure Transport".

[i.17] IETF RFC 5280: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

[i.18] IETF RFC 4210: "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)".

[i.19] ETSI TS 133 310: "Universal Mobile Telecommunications System (UMTS); LTE; Network Domain Security (NDS); Authentication Framework (AF) (3GPP TS 33.310)".

# 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.3] and the following apply:

**attribute certificate:** data structure, digitally signed by an Attribute Authority, that binds some attribute values with identification information about its holder

## 3.2 Symbols

Void.

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GS NFV 003 [i.3] and the following apply:

| | |
|---|---|
| CA | Certificate Authority |
| CSR | Certificate Signing Request |
| PKC | Public Key Certificate |
| PKI | Public Key Infrastructure |
| RA | Registration Authority |
| RCA | Root CA |

# 4 Rationale and approach for the use of public key certificates

## 4.1 Scope

The present document provides a guide to the use of Public Key Infrastructures (PKI) for the purpose of distributing Public Key Certificates (PKC) as applicable to the ETSI ISG NFV for the support of Public Key Cryptography in authenticating, authorizing and encrypting links between objects in NFV.

Each operator should develop Certificate Policy in accordance with their regional and national requirements. The present document assumes that the reader is generally familiar with Digital Signatures, PKIs, and core ETSI NFV specifications. The present document is consistent with the Internet X.509 Certificate Policy and Certification Practices Framework as defined in IETF RFC 3647 [i.14]. The certificate policy defines the structure of PKI.

NOTE: The PKIs described in the present document are privately managed, thus non-private (non-permissioned) PKIs are out of scope of the present document.

## 4.2 PKI Participants

### 4.2.0 Introduction

An NFV PKI can be implemented as a multi-tier hierarchy with a Root Certification Authority (RCA) at tier 1. There may be many certificate chains anchored by the RCA. Identified chains can be organized functionally and might include NFVI, VNF, MANO, and Support (such as OSS/BSS). A representative certificate hierarchy is shown in figure 4.2.0-1. The fewer tiers there are in the hierarchy, the smaller attack surface is, at the cost of limiting the number of trust domains.

The end-entity certificate, its private key, and all sub-CA certificates for a given CA chain should be installed on the device (hardware resource or software element, as appropriate). During authentication messaging exchange (using TLS or similar protocol) the end-entity and all sub-CA chain certificates should be sent to the other end point.

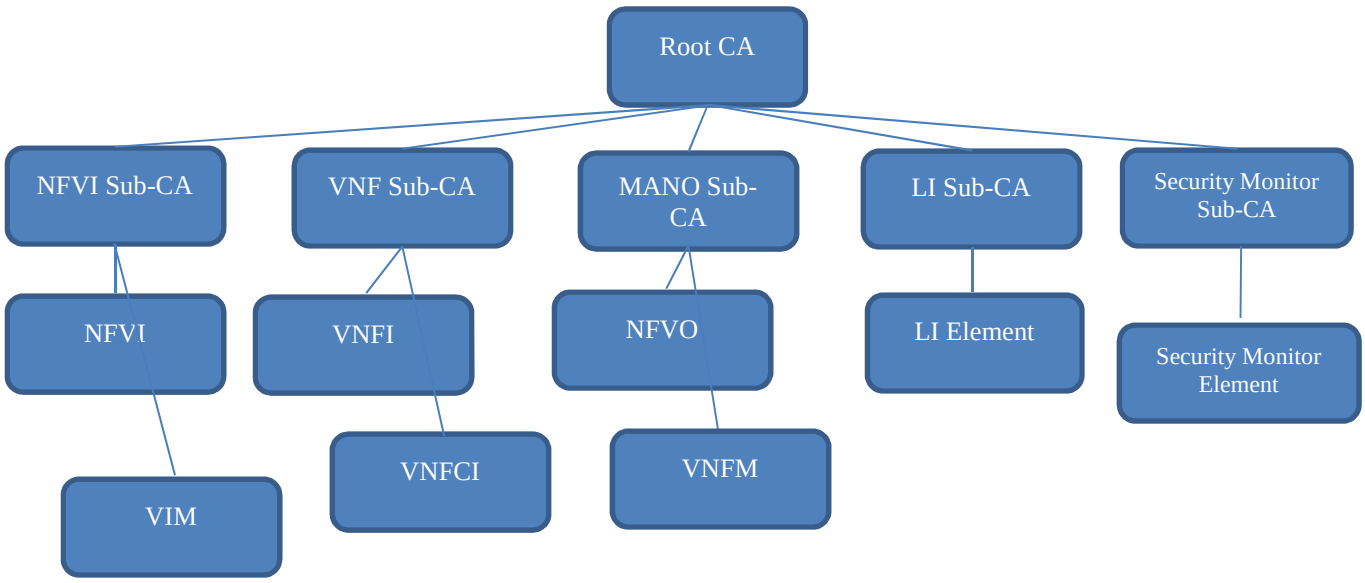


**Figure 4.2.0-1: ETSI NFV PKI Certificate Hierarchy**

Support of multiple roots is possible and when used it is expected to be specified by the operator. To anchor trust, certificates issued in ecosystems comprised of multiple roots have to be verifiable (chainable) to the corresponding root. This may be accomplished by cross signing certificates or allowing subscribers to honour multiple roots. This may provide ecosystem supply chain benefits at the risk of a substantially increased PKI attack surface. Furthermore, PKI operations of either deploying multiple valid chains or executing cross signing while achieving security over time has proven difficult.

PKI participants can include registration authorities, subscribers, relying parties, and auditors. PKI participants are described below.

### 4.2.1 Certificate Authorities

The entities called Certificate Authorities (CAs) are the heart of the ETSI NFV PKI. The CA is an aggregate term encompassing the collection of hardware, software, and operating personnel that create, sign, and issue public key certificates to Subscribers or other CAs. The CAs are responsible for:

- Implementing and maintaining a Certificate Policy (CP).
- Issuing compliant certificates.
- Delivery of certificates to Subscribers in accordance with the CP and other documents such as a Subscriber Agreement.

- Revocation of certificates.
- Generation of key pairs, protection, operation, and destruction of CA private keys.
- CA certificate lifecycle management ensure that all aspects of the CA services, operations, and infrastructure related to certificates issued under the CP are in fact compliant to the CP.
- Facilitating as a trusted party the confirmation of the binding between a public key and the identify, and/or other attributes, of the "Subject" of the certificate.

Sub-CAs are operated by designated sub-CA service providers and issue end-entity device certificates to subscribers.

### 4.2.2 Registration Authorities

Registration authorities (RAs) are entities that enter into an agreement with a CA to collect and verify each Subscriber's identity and information to be entered into the Subscriber's certificates. The RA performs its function in accordance with the CP and will perform front-end functions of confirming the identity of the certificate applicant, approving or denying certificate applications (manual) or requests (dynamic), requesting revocation of certificates, and managing account renewals.

### 4.2.3 Subscribers

The Subscriber is an organization or process acting on behalf of an organization identified in a Digital Certificate Subscriber Agreement (DCSA). The Subscriber is responsible for completing the certificate application or request. The CA relies on the RA to confirm the identity of the Certificate Applicant and either approves or denies the application or request. If approved, the RA communicates to the CA, and the Subscriber can then request certificates.

Subscribers are expected to comply to both CP requirements and any additional certificate management practices that govern the Subscribers' request for certificates and for handling the corresponding private keys. The Subscriber agrees to be bound by its obligations through execution of the DCSA between the Subscriber and the RA, and any other applicable agreements.

Technically, CAs are also Subscribers of certificates within a PKI, either as a Root CA issuing a self-signed certificate to itself, or as a sub-CA. However, in the present document, Subscriber apply only to the organization requesting device certificates, including those Subscribers who may have arranged to have a sub-CA operated onsite at their facility.

### 4.2.4 Relying Parties

Relying Parties validate the binding of a public key to a Subscriber's name in a device certificate. The RP is responsible for deciding whether or how to check the validity of a certificate by checking the appropriate certificate status information. The RP can use the certificate to verify the integrity of a digitally signed message, to identify the initiator of a communication, to attest the validity of a device setting or software component, or establish confidential communications with the holder of the certificate. For instance, an NFVi resource can use the device certificate presented by a Support server providing a firmware update and validate the signature of the signed firmware.

### 4.2.5 Auditors

PKI participants compliance to the CP may be verified by a third party authority.

## 4.3 Mapping of secure relationships to NFV reference points

The NFV reference architectural framework as defined in ETSI GS NFV 002 [i.2] identifies a number of named reference points and the set of allowed entities that communicate via them. Any of these entities or components may benefit by having a certificate and associated and protected private key to execute cryptographic security functions with other terminating entities.

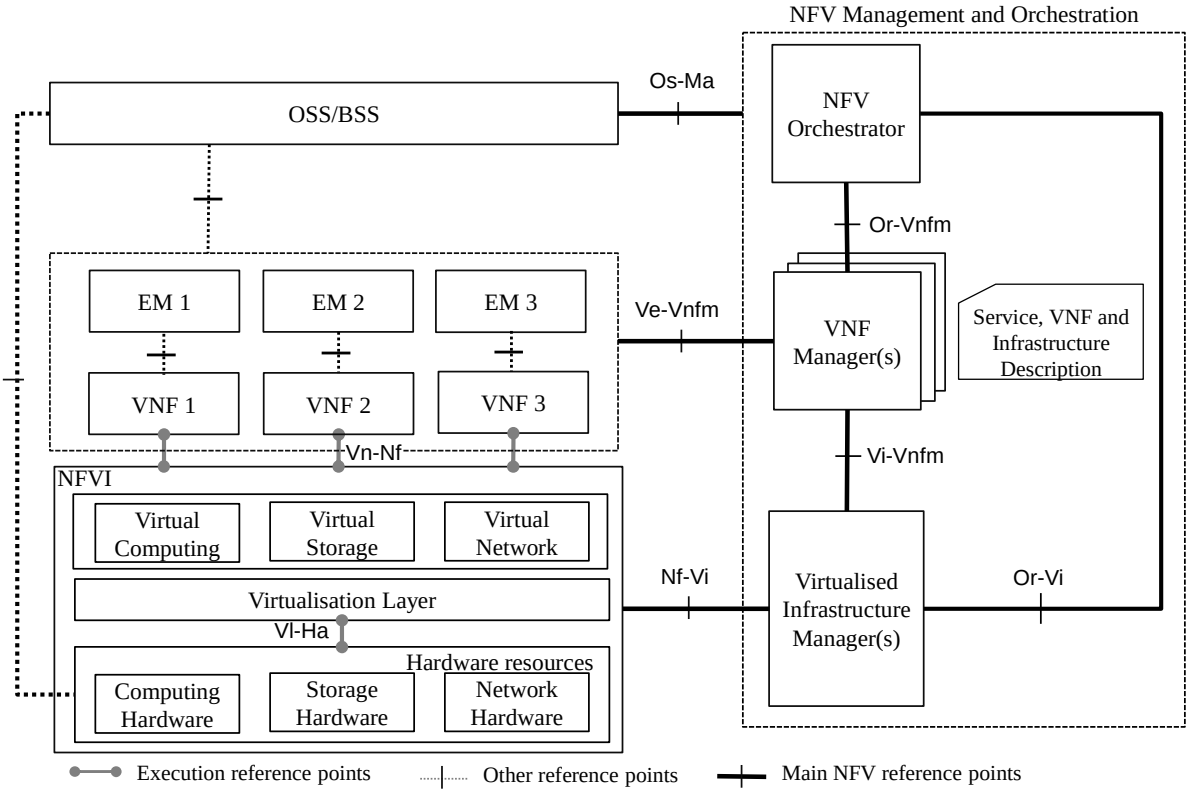


**Figure 4.3-1: NFV reference architectural framework**

The means by which participants are connected in the PKI is expected to be specified either using online or off-line processes. Furthermore, the Sub-CAs and RAs may exist within the MANO functionality or OSS/BSS environment. Online CA connectivity should be proxied probably via the OSS/BSS. This may facilitate on-line enrolment for certificate issuance in accordance with Enrolment of Secure Transport (EST, IETF RFC 7030 [i.17]).

**Table 4.3-1: Reference points and Functional Entities they link**

| Reference point classification | Reference point | PKI applicability | Terminating entities | |
|---|---|---|---|---|
| **Main NFV reference points** | **Os-Ma** | **Yes** | **MANO** | **OSS/BSS** |
| | Ve-Vnfm | Yes | VMF-Manager | EM or VNF |
| | Nf-Vi | Yes | VIM | NFVI |
| | Or-Vi | Yes | VIM | NFV Orchestrator |
| | Vi-Vnfm | Yes | VIM | VNF-Manager |
| | Or-Vnfm | Yes | VNF-Manager | NFV Orchestrator |
| Execution refence points | Vi-Ha | NA | Hardware resources | Virtualisation layer |
| | Vn-Nf | Yes | VNF | Network Function |
| Other reference points | Not specified | Yes | EM | VNF |
| | Not specified | Yes | OSS/BSS | EM/VNF |
| | Not specified | Yes | OSS/BSS | HW resources |
| NOTE: Vi-Ha is shown here as not applicable simply because it does not appear there is a technical solution (instruction set or other implementation) to allow a VNFI/VNFCI to cryptographically challenge the hardware on which it is being installed. This is a gap as this capability would be useful. | | | | |

## 4.4 Use Cases for the use of certificates in NFV

The benefit to using PKI is the ability to establish security associations between any entity within the domain of the PKI. Security associations are application of security principals to each of the reference points implemented in NFV. The security principals addressable by PKI including authentication, encryption, and signing. Transport Layer Security (TLS) as specified by IETF RFC 8446 [i.15] provides support for authentication, encryption, and message authentication (signing). File or image signing can also be supported by PKI and may be useful in NFV for distribution of images, packages, and configuration files.

Reference points may be applied between both trusted and untrusted entities. This may apply to multi-tenant or multi-operator environments or to high risk functions within a single-tenant and single-operator environment (such as security monitoring or lawful intercept functions). These use cases and how authentication, encryption, and signing are applied become the primary security association use cases in application of PKI. The criticality of benefit of these capabilities are shown as high, medium, and low in the following tables. The present document is informative, but the intent of the criticality is to indicate the priority of actions: to be mandated (high), to be highly recommended (medium), and to be given careful consideration (low) be done. Also, the use case model here does not imply that PKI and use of PKC are the only way to achieve authentication, encryption, and signing.

**Table 4.4-1: PKI trusted use case mapping to NFV reference points**

| Reference point | Authentication | Encryption | Signing/message authentication |
|---|---|---|---|
| Os-Ma | High | Medium | Low |
| Ve-Vnfm | High | Medium | Low |
| Nf-Vi | High | Medium | Low |
| Or-Vi | Medium | Low | Low |
| Vi-Vnfm | Medium | Low | Low |
| Or-Vnfm | Medium | Low | Low |
| Vi-Ha | NA | NA | NA |
| Vn-Nf | Medium | NA | NA |
| EM-VNF (not specified) | High | Medium | Low |
| OSS/BSS-EM/VNF (not specified) | High | Medium | Low |
| OSS/BSS-NFVi (not specified) | High | Medium | Low |

**Table 4.4-2: PKI untrusted use case mapping to NFV reference points**

| Reference point | Authentication | Encryption | Signing/message authentication |
|---|---|---|---|
| Os-Ma | High | High | Medium |
| Ve-Vnfm | High | High | Medium |
| Nf-Vi | High | High | Medium |
| Or-Vi | High | High | High |
| Vi-Vnfm | High | High | High |
| Or-Vnfm | High | High | High |
| Vi-Ha | NA | NA | NA |
| Vn-Nf | High | NA | NA |
| EM-VNF (not specified) | High | Medium | Medium |
| OSS/BSS-EM/VNF (not specified) | High | Medium | Medium |
| OSS/BSS-NFVi (not specified) | High | Medium | Medium |

While the uses above focus on security associations to support reference points explicitly included on the ETSI NFV reference architecture, any interface connecting to an NFV component can similarly implement authentication, encryption, and signing. Moreover, while authorization in context of role-based or attribute-based access controls not explicitly treated here, use of PKI credentials rather than traditional user or process identities may provide for greater confidence policy assertions. Moreover, network wide attestation may be similarly possible.

## 4.5 Considerations for PKC validation

### 4.5.1 Certificate path building and chain validation

When a PKC is received by an application (e.g. during TLS negotiation), in order for the entity proffering the certificate to be trusted and for the relying party to trust any assertions made in the PKC, the relying party is required to validate the PKC by means of verifying the signature of the certificate with the known public key of the PKC issuer, and verifying the validity of the PKC. Details of steps that should be taken to validate the certificate are outlined in clause 4.5.1.

Any decision to act on the content of a valid PKC is independent of the validity of the PKC, however an invalid PKC should in most circumstances be discarded and any information asserted by the key in the PKC should not be acted on.

A certificate chain (or certificate path) is validated from the end-entity certificate (i.e. the certificate of the entity that the relying party is authenticating or connecting to), through to the root acting as the Trust Anchor of all PKCs in the PKI. The relying party may accept the validity of the end-entity certificate at any stage in the tree, e.g. if any intermediate certificate is considered as "fresh" validation of the chain may be chosen to stop when validation checks reach the first fresh and valid point in the certificate chain.

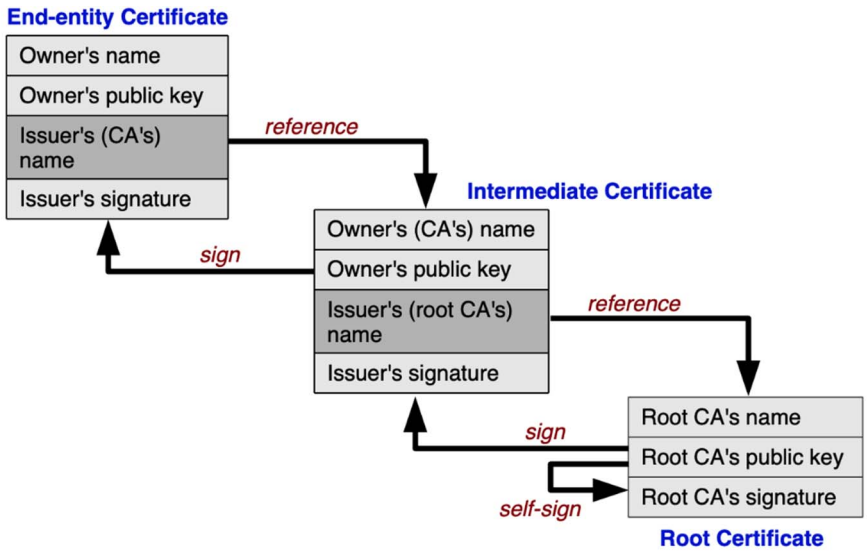


**Figure 4.5.1-1: Conventional PKI structure**
**(from https://upload.wikimedia.org/wikipedia/commons/d/d1/Chain_of_trust.svg)**

The following steps provide a guide to implementing certificate path building and validation. At minimum, a relying party needs to make checks consistent with the relying party's Certification Policy which should include the following technical steps (as defined in the Recommendation ITU-T X.509 specification [i.13]:

a) Check that the signature on the certificate is properly formatted and can be cryptographically verified by using the public key present in the issuer's PKC.

b) Check that the current date and time is within the validity period of the certificate. In particular check that the *notBefore* value is before the time at which the certificate is checked and that the *notAfter* value is after the time at which the certificate is checked.

c) Check that the value of the Issuer field of the certificate matches the value of the Subject field of the issuer's PKC.

d) Check that the *basicConstrains* extension is present in the issuer's PKC and that the value of the CA field is set to *TRUE*. Also, if the *pathLenConstraints* value of the extension is set, check that its value is present and set higher or equal to the current level of the certificate in the chain minus one. For example, in a three-level hierarchy (i.e. End-Entity - level 0, Intermediate CA - level 1, and Root CA - level 2), the value in the Intermediate CA's certificate (if present) should be equal to or greater than 0. For the Root CA's certificate, the value should be greater than or equal to 1. In order to provide flexibility, the *pathLenConstraints* is usually not present in Root CA's certificates.

e) Check for the presence of *authorityKeyIdentifier* extension. If present, and the *keyIdentifier* field is set, check that its value matches the *subjectKeyIdentifier* extension's value in the next certificate in the chain (if present). The values in these extensions are usually calculated by using the Method 1 as described in IETF RFC 5280 [i.17].

f) Check that the *keyUsage* extension in the next certificate in the chain supports certificate signing (i.e. the *keyCertSign* bit is set).

The PKIs may add attribute certificates to the PKC contents. The relying party may be required, in that case, to check for the presence of specific Object Identifiers (OID) in the *certificatePolicies* extension. Relying parties with specific policy requirements (such as subscribers' authentication servers or UE identifiers) should have a list of acceptable policy identifiers that should be used to verify the identifiers present in the certificates. In that case, the relying party should process the extension as follows:

a) Check that the *certificatePolicies* extension is present in the certificate. The value of this extension is a set of *certificatePolicy* values that should be check against the values set in the CP (if present). In particular, for each of the values, the relying party should check that:

- The required values of the *certPolicyId* field are present. For example, if the CP mandates for a specific value (*1.3.6.1.4.1.XXXX.YYY.ZZZ*) to be present in EE or SubCA certificates, the relying party should retrieve the content of the *certPolicyId* field of the *certificatePolicy* and check it against the required value.

- Although the use of *policyQualifiers* is discouraged as it might introduce interoperability issues, if the *policyQualifiers* field in the *certificatePolicy* extension is set, then the relying party should process the values according to their types as described in IETF RFC 5280 [i.17]. The detailed processing of these values is out of the scope of the present document.

To continue the chain building process, the relying party should repeat the steps above until one trusted certificate is reached.

# 5 Use cases for the use of certificates in NFV

## 5.1 VNF certificate use case

### 5.1.0 Introduction to use cases

VNFs are implemented with one or more VNFCs, which are internal component of a VNF providing a VNF Provider a defined sub-set of that VNF's functionality, with the main characteristic that a single instance of this component (i.e. VNFCI) maps 1:1 against a single Virtualisation Container. A VNF instance composed of various VNFCIs could have multiple logical identities, each of which is represented by a certificate, to communicate with different peers. In all of the use cases below, there is a pre-condition that a pre-established relationship exists between the communicating peers and the CA/RA respectively.

### 5.1.1 Use case #1: VNF management connection

A VNF instance should be configured and managed by both VNFM and EM, while it needs to be identified in order to be managed and configured. With the precondition that the VNFCI does not have a pre-established relationship with either the VNFM or EM, in order to ensure the security of this management path, a secure connection between a VNFCI and its corresponding VNFM or EM requires a VNFCI to have one or more certificates provisioned to attest its identity to the VNFM or EM to establish a secure connection between them.

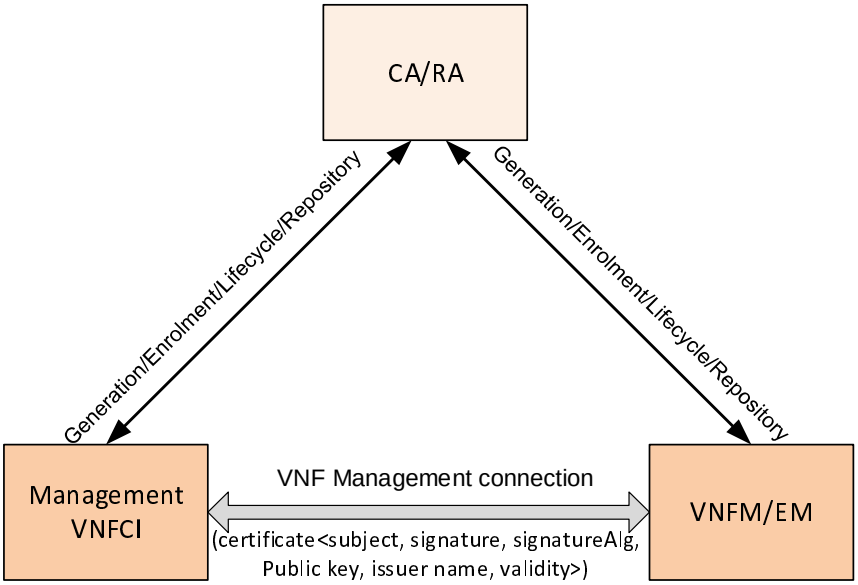


**Figure 5.1.1-1: Use case#1 VNFCI management connection**

Actors: VNFCI, VNFM/EM, CA/RA.

In this use case, VNFCI and VNFM or EM are validated by CA/RA and get the certificate(s) issued by CA/RA respectively. VNFCI sends its certificate to VNFM or EM, in which the attributes such as <subject, signature, signatureAlg, public key, issuer name, validity, etc.> are included. And VNFM/EM can verify VNFCI's identity via the certificate. And vice versa, VNFCI can validate VNFM/EM's identity in a similar fashion.

## 5.1.2 Use case #2: VNF transport connection

The VNFCI has the requirement to communicate with other entities, including other VNFCIs, PNFs, etc. With the precondition that the VNFCI does not have a pre-established relationship with either the VNFM or EM, a secure connection (e.g. Ipsec) between the VNFCI and the peer or a SeGW requires a VNFCI to have one or more certificates provisioned to attest its identity to the communication peers or SeGWs to establish secure connections between them.

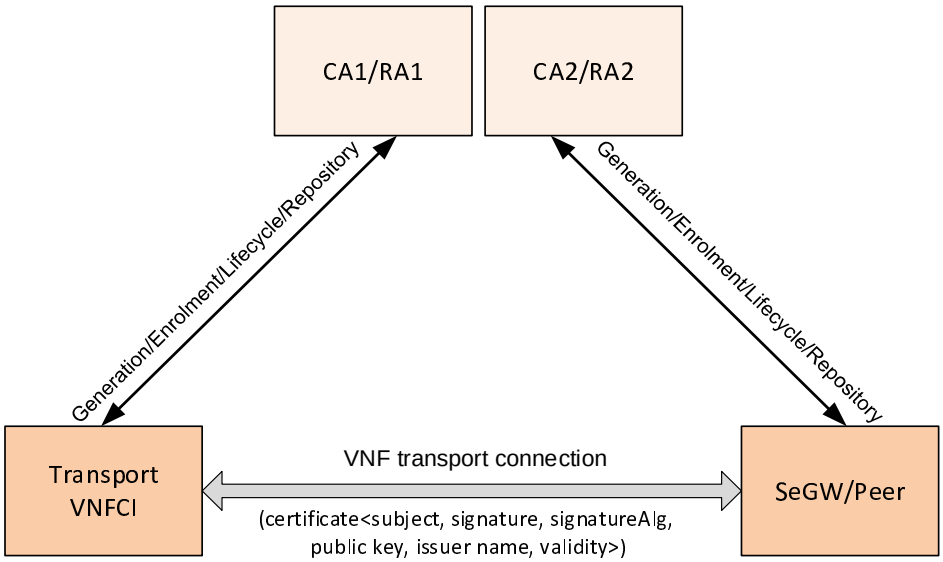


**Figure 5.1.2-1: Use case#2 VNFCI transport connection**

Actors: VNFCI, CA1/RA1, SeGW/Peer, CA2/RA2.

In this use case, VNFC is validated and gets the certificate(s) issued by CA1/RA1, and SeGW/Peer gets the certificate(s) issued by CA2/RA2. CA1/RA1 and CA2/RA2 may be the same one in some situations. VNFC sends its certificate to SeGW/Peer, in which the attributes such as <subject, signature, signatureAlg, public key, issuer name, validity, etc.> are included. And SeGW/Peer can verify VNFCI's identity via the certificate. And vice versa, VNFCI can validate SeGW/Peer's identity in a similar fashion.

## 5.2 MANO certificate use case

As entities with longer lifetime, MANO functional blocks (including NFVO, VNFM and VIM) need to communicate with each other, as well as with OSS/BSS, VNF, EM or NFVI. A secure connection (e.g. TLS) between the MANO and the peer requires a MANO functional block to have one or more certificates provisioned to attest its identity to the communication peer to establish a secure connection between them.

## 5.3 OSS/BSS/EM certificate use case

As traditional functional blocks, OSS/BSS need to communicate with EM and NFVO respectively, and OSS/BSS need to communicate with OSS/BSS, VNF and VNFM respectively. A secure connection (e.g. TLS) between these traditional functional blocks and the peer requires OSS/BSS or EM to have one or more certificates provisioned to attest its identity to the communication peer to establish a secure connection between them.

# 6 Recommendations

## 6.1 General

In order to eliminate or mitigate risks against attacks such as spoofing, tampering and information disclosure, secure connection can be established on all the new interfaces introduced by NFV scenario. IPsec and TLS mechanisms are widely deployed to protect the links between two communication entities using certificates as the credentials.

In NFV scenario, the functional blocks to be issued certificates include:

**NFV-MANO functional blocks and VNFCI**

[Recommendation-1] The NFV-MANO functional blocks and VNFCI should employ certificates which can be used in order to establish secure connections between them.

**Other functional blocks**

[Recommendation-2] OSS/BSS should employ certificates in order to establish secure management connections with NFVO.

[Recommendation-3] EMS should employ certificates in order to establish secure management connections with VNF or VNFM.

[Recommendation-4] NFVI (i.e. the control & admin agents in NFVI), should employ certificate(s) in order to establish secure connections with VIM.

## 6.2 Deployment recommendations

It is stated in NFV Requirements (ETSI GS NFV 004 [i.4]) that the NFV framework (ETSI GS NFV 002 [i.2]) is expected to implement appropriate security countermeasures to address protection of data transmitted via shared network resources and protection of new interfaces exposed by the interconnectivity among NFV architectural components (e.g. hardware resource, VNFs and management systems). In order to realize the authentication, data confidentiality and integrity protection, some cryptographic security algorithms may be employed.

Clause 5.1 of the ETSI GS NFV-SEC 001 [i.8] describes seven deployment scenarios:

- Monolithic Operator.

- Network Operator Hosting Virtual Network Operators.
- Hosted Network Operator.
- Hosted Communications Providers.
- Hosted Communications and Application Providers.
- Managed Network Service on Customer Premises.
- Managed Network Service on Customer Equipment.

Multi-tenant is a key factor in several of these scenarios. Because different providers and operators create different administrative domains, while each certificate applies to a specific security domain, multi-tenant scenarios need to be considered for certificate deployment.

It is possible to differentiate two identified administrative domains as defined in ETSI GS NFV-MAN 001 [i.5] for deployment scenarios, although additional administrative domains may exist. The two domains are by default separate PKIs:

- **Infrastructure Domain:** The Infrastructure Domain provides virtualised infrastructure resources such as computing, networking and storage or a composition of those resources via a service abstraction to a Tenant Domain, and is responsible for the management and orchestration of those resources.
- **Tenant Domain:** The Tenant Domain provides VNFs, and combinations of VNFs into Network Services, and is responsible for their management and orchestration, including their functional configuration and maintenance at application level.

By applying this two administrative domains approach to the seven NFV deployment scenarios in ETSI GS NFV-SEC 001 [i.8] using the NFV reference architectural framework as defined in ETSI GS NFV 002 [i.2], it is possible to envision associated certificate management scenarios.

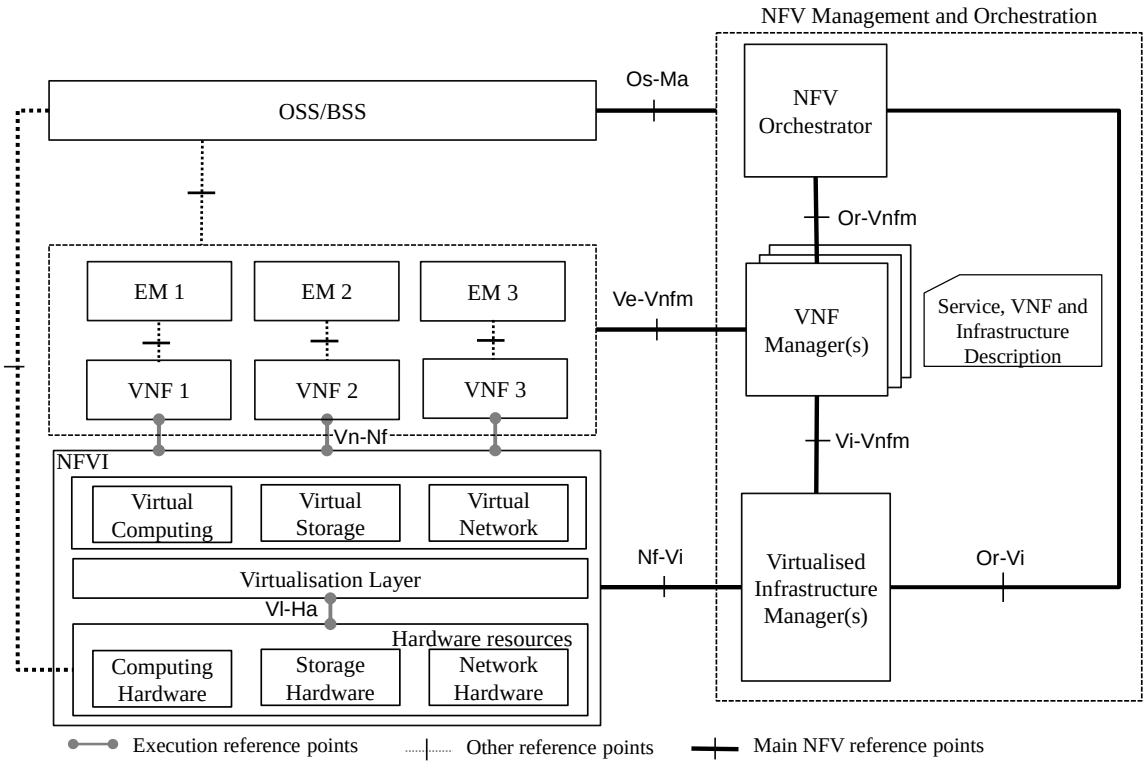


**Figure 6.2-1: NFV reference architectural framework (ETSI GS NFV 002 [i.2])**

**Monolithic Operator**

The same organization that operates the virtualised network functions deploys and controls the hardware and hypervisors they run on and physically secures the premises in which they are located.

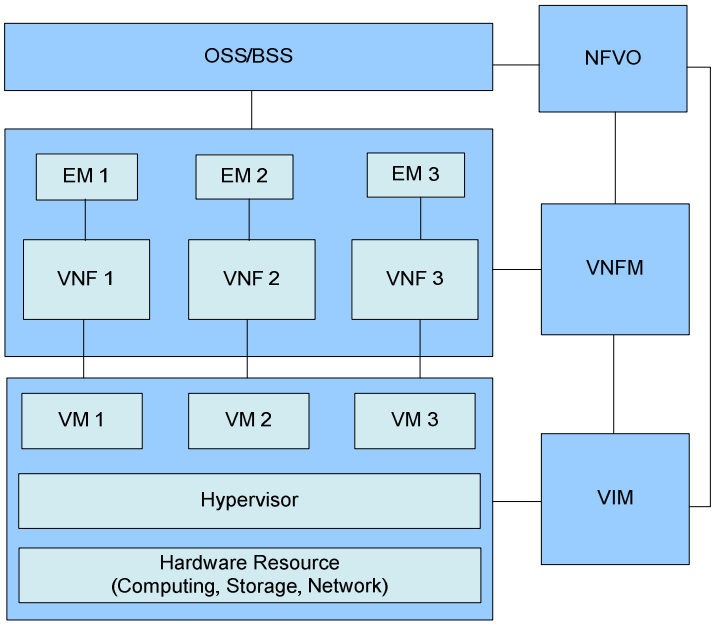


**Figure 6.2-2: Deployment scenario 1**

In this scenario, the entire NFV network is operated by one operator. This is the simplest deployment scenario. Since there is only one administrative domain, all the needed certificates can be issued by the CA of network operator domain.

**Network Operator Hosting Virtual Network Operators**

The network operator hosts VNFs for itself and other virtual network operators , within the same facility.

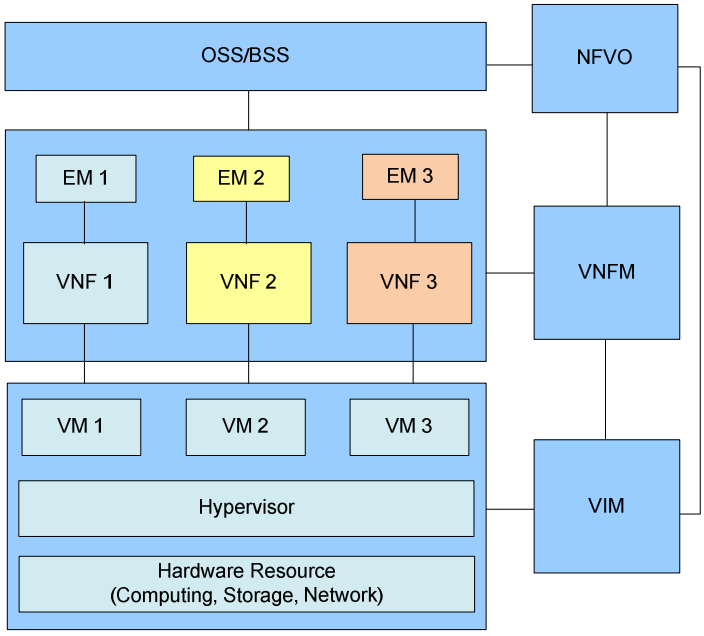


**Figure 6.2-3: Deployment scenario 2**

In this scenario, since VNF 1, VNF 2 and VNF 3 belong to different network operators respectively, the certificates issued to VNF 1, VNF 2 and VNF 3 should reside in different administrative domains.

**Hosted Network Operator**

An IT services organization operates the computer hardware, infrastructure network and hypervisors on which a separate network operator runs virtualised network functions.

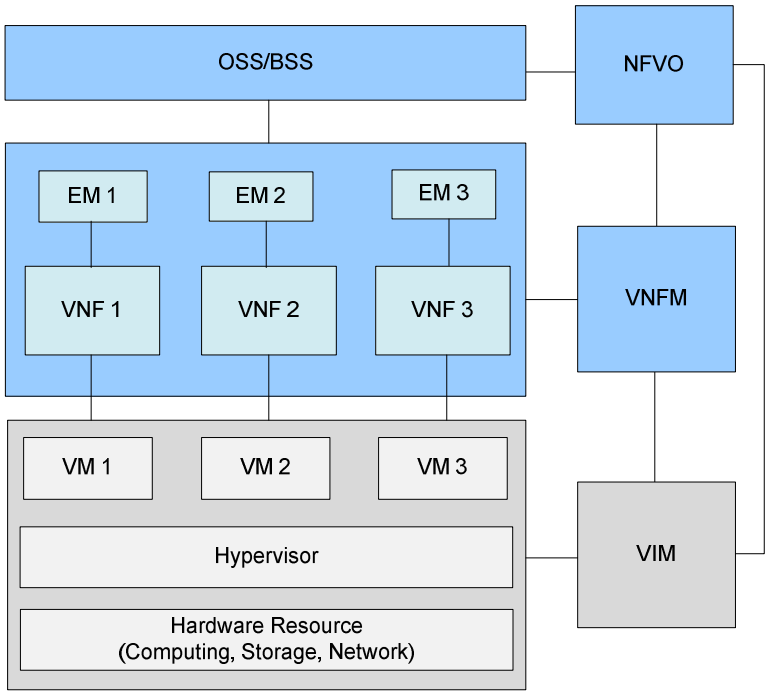


**Figure 6.2-4: Deployment scenario 3**

In this scenario, since the infrastructure and the above VNFs are provided and operated by different providers/operators, the certificates issued to the entities in Infrastructure Domain and Tenant Domain may reside in different administrative domains.

**Hosted Communications Providers**

This scenario is similar to the Hosted Network Operator scenario, except the IT services organization hosts multiple communications providers.

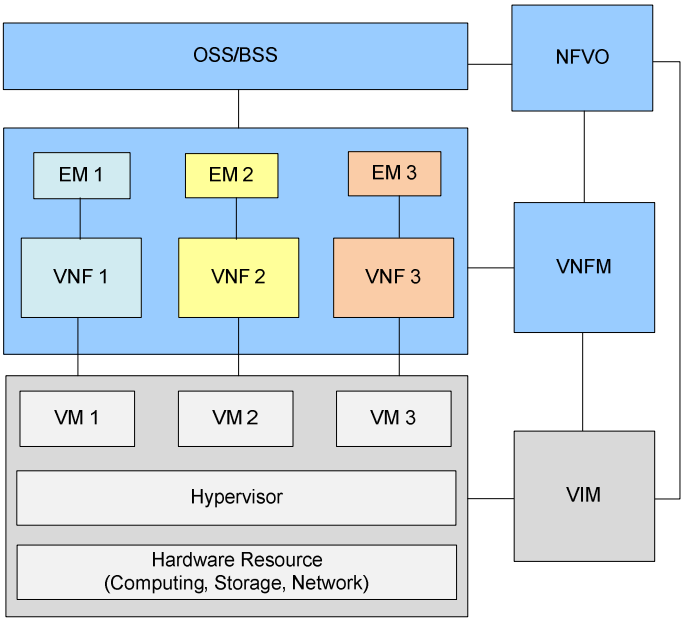


**Figure 6.2-5: Deployment scenario 4**

In this scenario, because infrastructure, VNF 1, VNF 2 and VNF 3 belong to different providers/operators respectively, the certificates issued to the entities in Infrastructure Domain and Tenant Domain may reside in different administrative domains.

**Hosted Communications and Application Providers**

This scenario is similar to the Hosted Communications Providers scenario, except servers in a data centre facility are offered to the public for deploying virtualised applications. Similarly, the certificates issued to the entities in Infrastructure Domain and Tenant Domain may reside in different administrative domains.

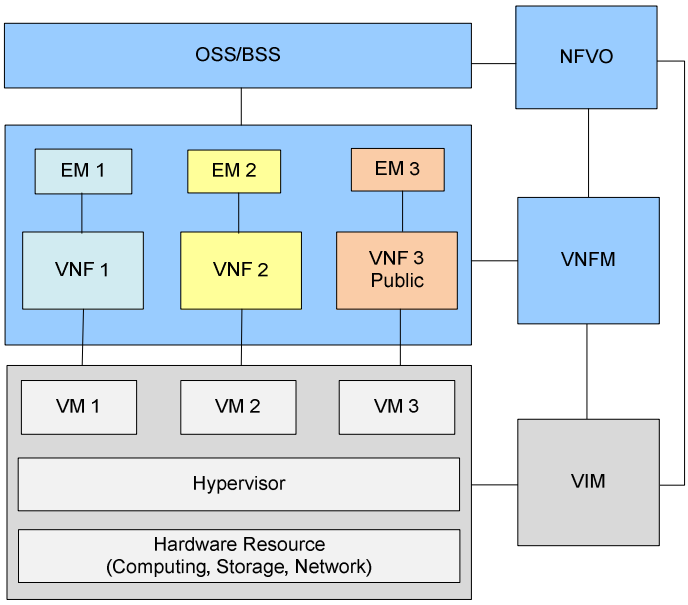


**Figure 6.2-6: Deployment scenario 5**

**Managed Network Service on Customer Premises**

In this scenario, a network operator runs virtualised network functions on its own generic server hardware located on a customer's premises and physically secured by the customer, normally under a contractual agreement between the network operator and the customer.

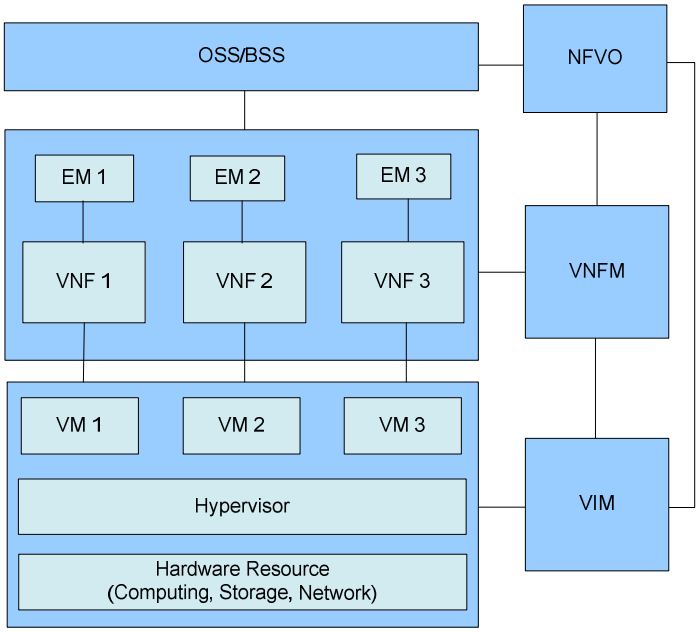


**Figure 6.2-7: Deployment scenario 6**

Because there is no impact on certificate deployment if the building belongs to network operator or customer, the certificate issuance is the same as that of a Monolithic Operator, i.e. all the needed certificates can be issued by the CA of network operator domain.

**Managed Network Service on Customer Equipment**

This scenario is similar to the Managed Network Service on Customer Premises scenario, except the computer hardware is supplied and operated by the customer rather than the network operator.

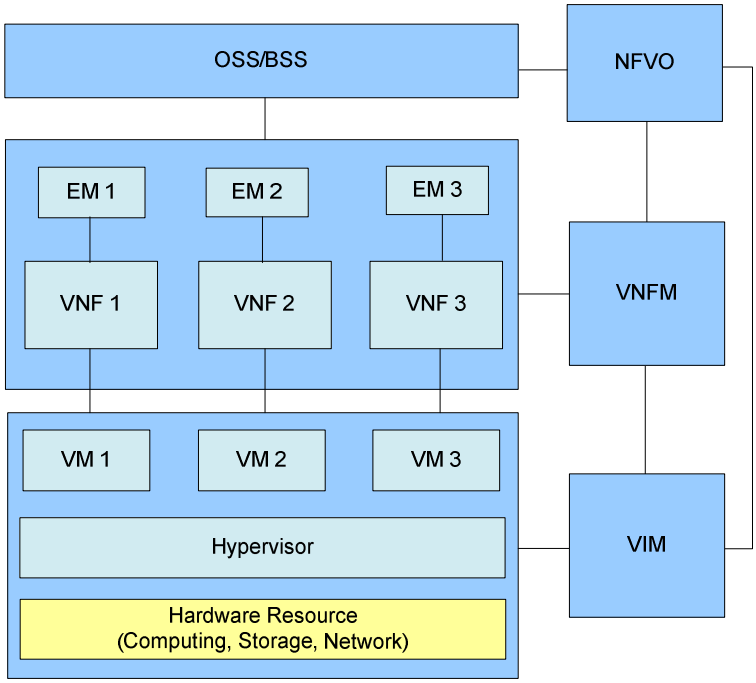


**Figure 6.2-8: Deployment scenario 7**

There may be a requirement to configure certificates for Lights-out Management (i.e. deploying a form of out-of-band management often using a dedicated management channel to allow monitoring and management of network-attached equipment regardless of whether the machine is powered on, or whether an operating system is installed or functional), which will be within the hardware management domain which is out of scope of the present document. The certificate issuance is similar to a Hosted Network Operator, i.e. the certificates issued to the hardware management domain and the entities in the Infrastructure and Tenant Domains may reside in different administrative domains.

# 7 Certificate management framework

## 7.1 Certificate hierarchy

Considering that the certificates may be deployed at two or more layers, however, the introduction of NFV brings new certificate deployment and management issues. A multi-layered certificate mechanism is desirable for NFV framework. The vertical layers that certificates need to be deployed could be as follows.

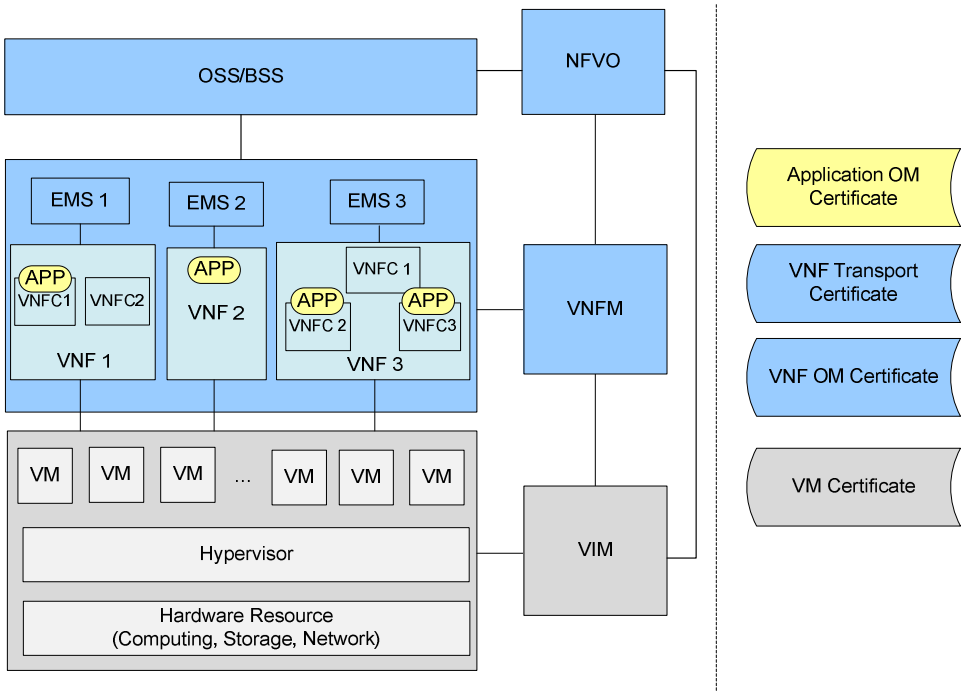


**Figure 7.1-1: NFV hierarchy**

- **Applications Environment Layer:** It provides orchestration and management functionality to application software or even 3rd party software installed on VNFs. It supports a flexible and efficient multi-tenancy run-time and hosting environment for Applications by providing both VNFaaS and VNPaaS facilities, allowing 3rd Party Application develops different degrees of control over processing power, memory, storage or operating system support. At this layer, each VNF can be configured with OM certificate(s) in order to manage the software installed on VNFs.

- **Execution Environment Platform Layer:** It provides the basic VNF functionalities. At this layer, each VNF can be configured with certificate(s) corresponding to the Tenant domain as defined by MANO.

- **Infrastructure Platform Layer:** It provides the hardware resources for the platform (e.g. CPU, memory, storage, acceleration devices, input/output devices etc.) together with the supporting operating system and virtualisation (hypervisor) software. At this layer, each VM can be configured with certificate(s) corresponding to the Infrastructure domain as defined by MANO.

Meanwhile, the horizontal layers could be defined between a variety of functionalities within the same layer, e.g. between VM and VIM, between VNF and VNFM, and between two VNFs hosted by the same or different operators.

At all these layers, each layer has a corresponding peer for function management, and each function at one layer may have a peer to communicate with. Certificates are needed to perform authentication to all the management protocols and peer-to-peer telecommunication protocols.

In the NFV multi-layered environment, the certificates need to be deployed at multiple layers, so it is quite complicated to consider where and how to deploy the certificates, and how to manage all the certificates deployed in different layers and different functionalities. The certificate management also needs to be embedded to the service procedures and make the trust relationship configurable and can be passed on reliably.

## 7.2 Certificate category

The above layering approach gives rise to corresponding certificate categories:

- **Application OM certificate:** It is used to establish management connection in order to perform management operations on VNFCIs.
- **VNFCI certificate:** It includes two types of certificates as below:
  - **VNFCI transport certificate:** It is configured to each VNFC instance which has the external communication requirement. It is used to establish secure connection with other peer entities (e.g. VNFCIs).
  - **VNF OM certificate:** It is used to establish management connection between VNFCI and VNF management entities (e.g. VNFM and EMS) in order to perform management operations to VNFCIs.
- **VM certificate:** It is configured to each VM in Infrastructure domain. It is used to establish management connection with VIM to perform management operations to VM.

Considering secure communication requirement on the new interfaces exposed by the interconnectivity among management systems, the MANO entities should also be configured with certificates to establish secure connection with the peer entities:

- **MANO certificate:** It is used to establish management connection with VM or VNF hosted on the VM. Moreover, secure connection can also be established between MANO entities in order to ensure secure communication.

Otherwise, some other management entities such as EMS also need to be configured with certificates in order to establish secure management links with VNFs. Because they are not new entities introduced by NFV scenario, no special description is needed to address this in the present document.

Due to the virtualisation, the introduction of NFV has a great impact on the deployment of VM certificate at Infrastructure Platform Layer and VNF certificate at Execution Environment Platform Layer. Compared to the traditional physical devices, the virtualisation makes VM and VNF created dynamically. Therefore, the present document aims to address the basic certificate management issues at Infrastructure Platform Layer and Execution Environment Platform Layer when VNF is instantiated initially.

# 8 NFV certificate lifecycle management

## 8.1 Certificate generation

### 8.1.1 Initial Credential

#### 8.1.1.1 Key pair generation

##### 8.1.1.1.1 Option 1: NFVI generates key pair

This option uses a key pair generation mechanism implemented by NFVI to generate the key pair. For this option, the VNFI acquires the key pair from the injection of the NFVI. Therefore, the MANO (NFVO&VNFM&VIM) does not know VNFI's private key. This key pair generation procedure is depicted in Figure 8.1.1.1.1-1.

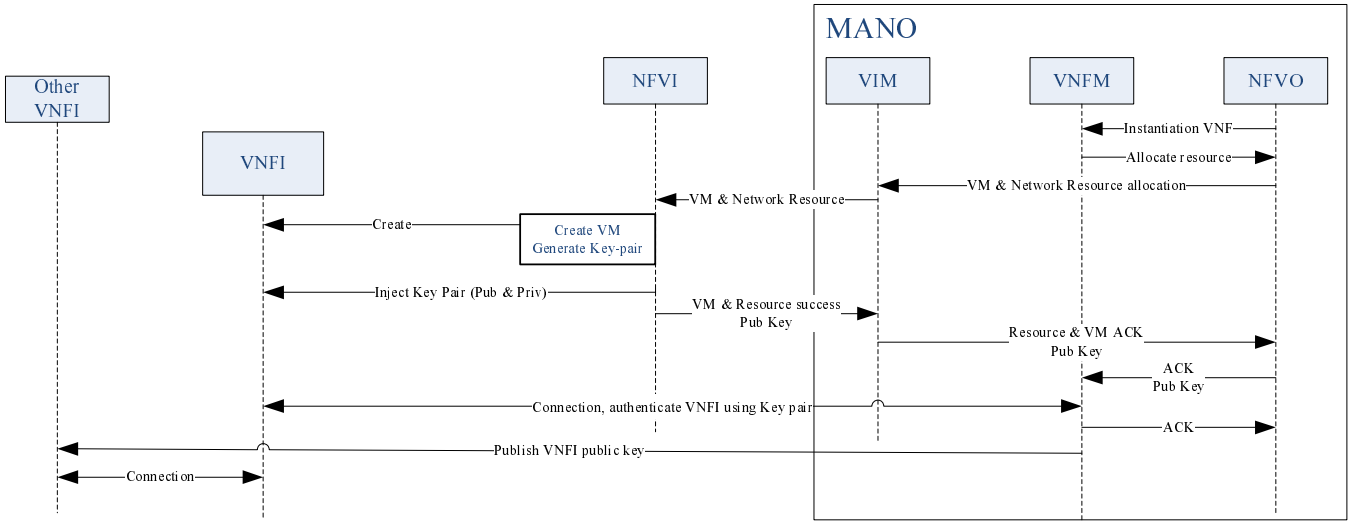


**Figure 8.1.1.1.1-1: NFVI generate key pair procedure**

1) NFVO calls VNFM to instantiate the VNF.
2) VNFM calls the NFVO for resource allocation.
3) NFVO requests allocation of resources to the VIM (VMs and network resources).
4) VIM forwards the resources allocation requests to NFVI.
5) NFVI generates the key pair and inject the key pair into the secure storage of the created VNFI (securely deleting the private key from its own storage). Meanwhile, NFVI confirms the successful instantiation VNFI back to the VIM, providing the VNF-ID and the public key as well.
6) VIM forwards the resources allocation response back to NFVO along with the public key.
7) NFVO acknowledges the completion of the resource allocation back to VNFM along with the public key.
8) VNFM establish trust relationship with the VNFI and configures the VNFI. Then VNFM acknowledges the completion of the VNFI instantiation back to the NFVO.
9) VNFM publishes the public key provided by NFVI.
10) Finally, the other network elements can establish trust relationship with the VNFI.

As the private key is only known to NFVI and the VNFI, the risk of the private key exposure is reduced.

#### 8.1.1.1.2 Option 2: HMEE generates key pair

This option uses a key pair generation mechanism implemented by HMEE to generate the key pair. For this option, HMEE is a secure part of the VNFI. NFVI cannot access the authentication function or read HMEE data. This key pair generation procedure is depicted in figure 8.1.1.1.2-1.

NOTE 1: A Hardware-Mediated Execution Enclave (HMEE) is defined as an area of process space and memory within a system environment within a computer host which delivers confidentiality and integrity of instructions and data associated with that enclave. See ETSI GS NFV-SEC 009 [i.10], clause 6.16.

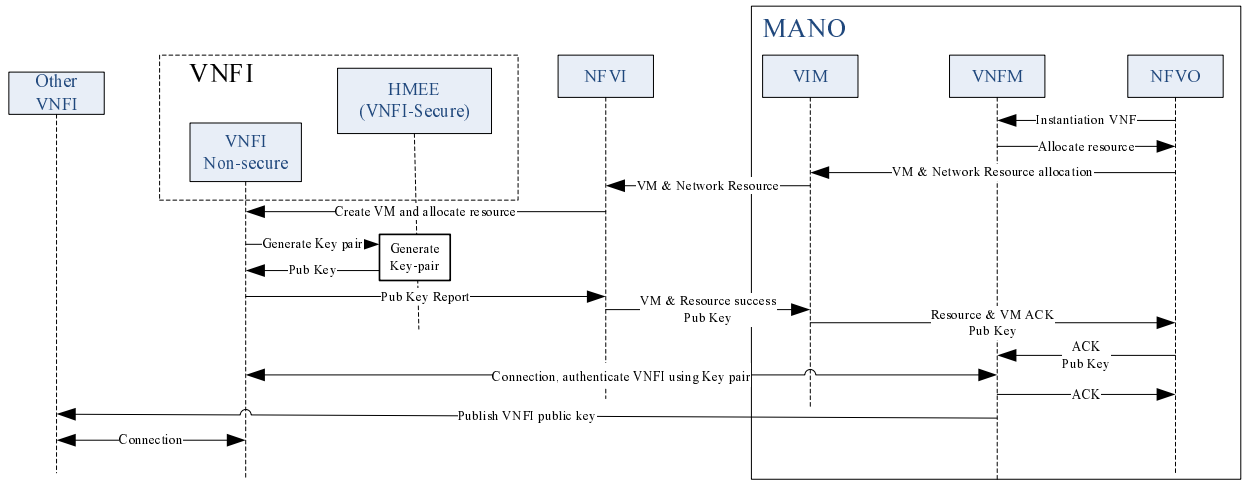


**Figure 8.1.1.1.2-1: HMEE generate key pair procedure1**

1) NFVO calls VNFM to instantiate the VNF.

2) VNFM calls the NFVO for resource allocation.

3) NFVO requests allocation of resources to the VIM (VMs and network resources).

4) VIM forwards the resources allocation requests to NFVI.

5) NFVI creates the VNFI with a HMEE protecting the sensitive part of the VNFI.

NOTE 2: The precise details of the creation and communication with the HMEE is outside the scope of the present document.

6) VNFI sends the key pair generation request message to HMEE.

7) The HMEE generates the key pair and reports the public key back to VNFI.

8) VNFI informs NFVI of the public key provided by HMEE.

9) NFVI confirms the successful initialization VNFI back to the VIM, providing the VNF-ID and the public key as well.

10) VIM forwards the resources allocation response back to NFVO along with the public key.

11) NFVO acknowledges the completion of the resource allocation back to VNFM along with the public key.

12) VNFM establish trust relationship with the VNFI and configures the VNFI. Then VNFM acknowledges the completion of the VNFI instantiation back to the NFVO.

13) VNFM publishes the public key provided by VNFI.

14) Finally, the other network elements can establish trust relationship with the VNFI.

As the private key is stored in the trust environment during the whole lifecycle, the risk of private key exposure does not exist.

#### 8.1.1.1.3 Option 3: HSM generates key pair

This option uses a key pair generation mechanism implemented by HSM to generate the key pair created using an ETSI GS NVF-SEC 012 [i.11] compliant random number generator. The HSM is linked to the HMEE where the VNFI is implemented through a secure channel established after a mutual authentication process. For this mutual authentication process, the public key of the HSM is expected to be loaded in the HMEE and the hardware (e.g. CPU) root certificate from which the HMEE certificate is generated is expected to be loaded in the HSM. How the public key of the HSM is introduced in the HMEE, and how the hardware root certificate is introduced in the HSM is out of scope of the present document.

NOTE 1: Hardware Secure Module (HSM) is defined in ETSI GS NFV-SEC 009 [i.10], clause 6.20.

NOTE 2: In the procedure described below, the VNFI contains a single component and is equivalent to a VNFCI.

The key pair generation procedure is depicted in figure 8.1.1.1.3-1.

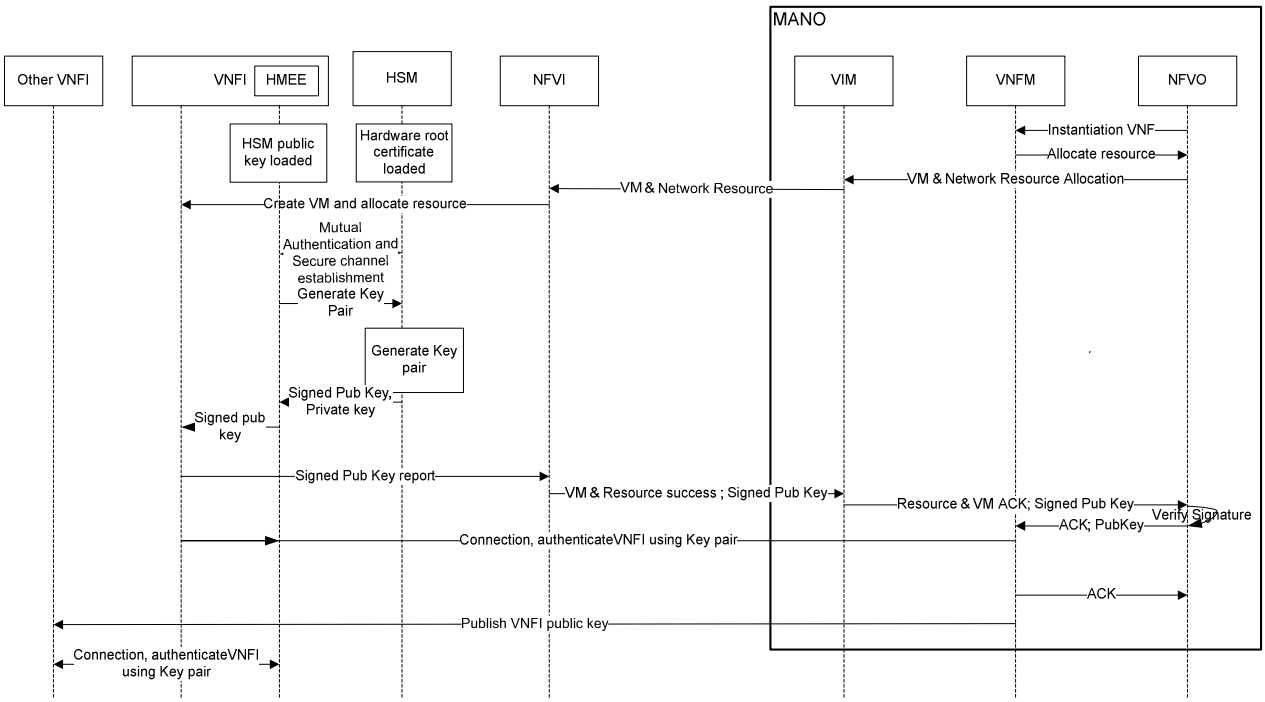


**Figure 8.1.1.1.3-1: HSM generate key pair procedure**

1) The public key of the HSM is introduced in the HMEE and the hardware root certificate is introduced in the HSM.

2) NFVO calls VNFM to instantiate the VNF.

3) VNFM calls the NFVO for resource allocation.

4) NFVO requests allocation of resources to the VIM (VMs and network resources).

5) VIM forwards the resources allocation requests to NFVI.

6) NFVI created the VNFI with a HMEE.

7) Mutual Authentication process between the HMEE and the HSM is done and a secure channel is established between HSM and HMEE.

8) VNFI through the HMEE sends the key pair generation request message to HSM.

9) The HSM generates the key pair and reports the signed public key and private key back to the HMEE. The public key is signed with the HSM key and the certificate may contain some other information (as the hash, version, etc.) of the VNFI.

10) VNFI informs NFVI of the signed public key provided by HSM.

11) NFVI confirms the successful initialization VNFI back to the VIM, providing the VNF-ID and the signed public key as well.

12) VIM forwards the resources allocation response back to NFVO along with the signed public key of the VNFI.

13) NFVO verifies the certificate of the signed public key and acknowledges the completion of the resource allocation back to VNFM along with the public key. The signature gives assurance to the NFVO on the key pair quality generated by a reliable source and information in the certificate assurance on version, integrity of the code of VNFI and the implementation of the VNFI in a HMEE.

14) VNFM establish trust relationship with the VNFI and configures the VNFI. Then VNFM acknowledges the completion of the VNFI instantiation back to the NFVO.

15) VNFM publishes the public key provided by VNFI.

16) Finally, the other network elements can establish trust relationship with the VNFI.

As the private key is stored in the trust environment during the whole lifecycle, the risk of private key exposure does not exist.

## 8.1.2 VNFCI Certificate

### 8.1.2.0 Introduction to VNFCI certificate issuance

A VNFCI certificate is issued by operator CA after a VNFCI is successfully instantiated. It is requested by the VNFCI that takes the initial credential as proof to Operator CA in the following CMPv2 certificate enrolment procedure.

The certificate configured to VNF should be aimed at VNFCI which has the communication requirement with external entities. Some options are listed for the formal certificate issuance as shown below.

All the following CMPv2 certificate enrolment procedures are in line with the specification of IETF RFC 4210 [i.18].

### 8.1.2.1 Option 1: VNFCI generates key pair, constructs and signs certificate request

This option uses a certificate enrolment mechanism implemented directly by VNFCI itself to request a certificate from Operator CA.

For this option, the VNFCI generates public-private key pair, constructs and signs the certificate request message. VNFCI needs to use the initial credential to request formal certificate issued by Operator CA according to the certificate enrolment procedure. Operator CA verifies the VNFC's identity using the initial credential.

Operator root CA certificate could be transferred to VNFCI either in the certificate enrolment procedure or in the previous instantiation procedure installed by NFVI. As defined by ETSI TS 133 310 [i.19], the protection of the operator root CA certificate during provisioning may be based on operator security policy. If an operator root CA certificate provisioned prior to the CMPv2 protocol run is available the VNFCI uses it. Otherwise, the VNFCI uses the operator root CA certificate provisioned during the CMPv2 run. If no operator root CA certificate is provisioned at all then the VNFCI cannot continue with the certificate enrolment procedure.

This CMPv2 enrolment procedure is depicted in figure 8.1.2.1-1.

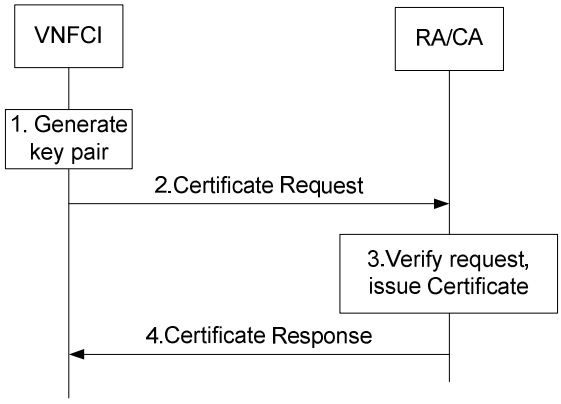


**Figure 8.1.2.1-1: Direct VNFCI certificate enrolment procedure**

1) After the successful instantiation, by the indication of certificate installed during instantiation, the VNFCI generates a public-private key pair to be certified by the Operator RA/CA.

2) The VNFCI constructs and sends Certificate Request message to Operator RA/CA to request a certificate, providing the generated public key. In this message the initial credential is used to authenticate the message by Operator RA/CA. In order to provide proof of possession, the VNFCI generates the signature for the POPOSigningKey field of the CertReqMsg using the private key.

3) Operator RA/CA verifies the Certificate Request message based on the initial credential, and the POP. If the verification is successful, it issues VNFCI a certificate.

4) RA/CA responds to the enrolment request, providing the new certificate. The VNFCI verifies the received message, if successful, installs the received certificate.

Additional policy could be applied to enhance the security, such as invalidating the initial credential once VNFCI installs a formal certificate successfully.

NOTE: This initial credential can be used for all VNFCIs, or just for a Master VNFCI (e.g. first VNFCI established during the instantiation procedure) which can be a longer-lived entity compared to other VNFCs in a VNF.

#### 8.1.2.2 Option 2: VNFCI generates key pair, constructs certificate request, and VNFM/NFVO signs certificate request

This option uses a certificate enrolment agent mechanism to leverage a VNFC instance to obtain its certificate issued by Operator CA with the help of VNFM that acts as an agent of the VNFCI.

For this option, the VNFCI generates public-private key pair, and constructs the public key certificate request message. VNFCI sends certificate request message to VNFM, using the initial credential to request formal certificate issued by Operator CA. And VNFM authenticates the message based on the initial credential, signs the message with the private key corresponding to VNFM certificate, and then acts as an agent to request a formal certificate to CA according to the certificate enrolment procedure. Operator CA issues VNFCI the formal certificate and sends it to VNFCI via the redirection of VNFM.

Operator root CA certificate could be transferred to VNFCI either in the certificate enrolment procedure or in the previous instantiation procedure installed by NFVI. As defined by ETSI TS 133 310 [i.19], the protection of the operator root CA certificate during provisioning may be based on operator security policy. If an operator root CA certificate provisioned prior to the CMPv2 protocol run is available the VNFCI uses it. Otherwise, the VNFCI uses the operator root CA certificate provisioned during the CMPv2 run. If no operator root CA certificate is provisioned at all then the VNFCI aborts the procedure.

This enrolment procedure is depicted in figure 8.1.2.2-1.

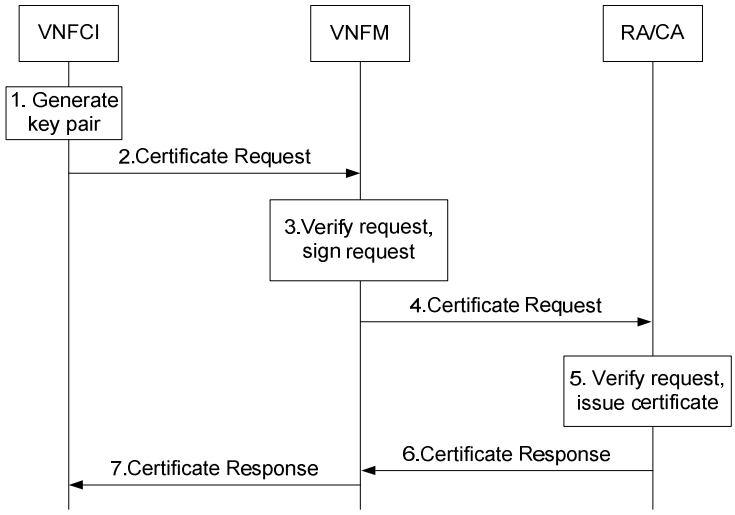


**Figure 8.1.2.2-1: VNFCI certificate agent enrolment procedure**

1) After the successful instantiation, by the indication of certificate installed during instantiation, the VNFCI generates a public-private key pair to be certified by the Operator RA/CA.

2) The VNFCI constructs and sends Certificate Request message to VNF Manager to request a certificate, providing the generated public key. In this message the initial credential is used to authenticate the message by VNFM or RA/CA. In order to provide proof of possession, the VNFCI generates the signature for the POPOSigningKey field of the CertReqMsg using the private key.

3) Based on the initial credential, VNFM verifies the Certificate Request message. If the verification is successful, VNFM acts as an agent of the VNFCI, and constructs and signs the Certificate Request message using VNFM certificate.

4) VNFM forwards the Certificate Request message to RA/CA.

5) RA/CA verifies the Certificate Request message and the POP, if successful, it issues VNFCI a certificate.

6) RA/CA responds to the enrolment request, providing the new certificate.

7) The VNFM verifies the Certificate Response message, and constructs and forwards Certificate Response message back to the VNFCI. The VNFCI verifies the received message, if successful, installs the received certificate.

NOTE: Authentication of VNFCI using the initial credential can be performed either by VNFM or by RA/CA, since both VNFM and RA/CA share the secret information. The above procedure assumes this authentication is performed by VNFM.

Additional policy could be applied to enhance the security, such as invalidating the initial credential once VNFCI installs a formal certificate successfully.

## 8.2 Certificate update

A certificate is updated as a new certificate issuance before the current certificate expires, that may also include name update, attribute update, public key update, expiration update, etc. Certificate update can initiate by VNFCI or by VNFM on behalf of the VNFCI.

Once the CA has issued a formal certificate for the VNFCI, the VNFCI is able to either request the CA directly or through the agent of VNFCI (i.e. VNFM) for any subsequent updates. The PKI should support either case for certificate update. If the public-private key pair needs to be updated, either VNFCI or NFVI could generate the key pair, which is determined by implementation.

If the option 1 in clause 8.1.2.1 is supported, the new updated certificate can be retrieved directly from CA. In such case IETF RFC 4210 [i.18] gives a practical example of certificate update mechanism and is the most popular protocol in certificate management.

If the option 2 in clause 8.1.2.2 is supported, the new updated certificate retrieved via the VNFM as an agent from CA may use the similar mechanism defined in clause 8.1.2.2.

# 9 NFV Certificate Management

## 9.0 Introduction

All NFV functional blocks should configure certificates, including MANO entities, VNFCI, VNF and some other O&M entities (e.g. EMS and OSS/BSS). Manual and automatic configurations are common ways used in certificate deployment. In order to support the requirement that the NFV framework incorporates mechanisms for automation of operational and management functions automation, the clauses below focus on automatic mechanisms for certificate management.

# 9.1 MANO and other functional blocks

The MANO functional blocks defined by ETSI GS NFV 002 [i.2] include:

- NFV Orchestrator (NFVO).
- VNF Manager (VNFM).
- Virtualised Infrastructure Manager (VIM).

Furthermore, some other functional blocks, such as EM and OSS/BSS, are entities exchanging information with NFV-MANO functional blocks or VNFs to perform management operation. Therefore, they are also described here.

Since MANO and other O&M functional blocks (i.e. EM and OSS/BSS) are long-lived entities, the certificate deployment is similar with the traditional non-virtualised entities, e.g. by manual or automatic configuration.

# 9.2 Tenant domain

## 9.2.1 VNF certificate

### 9.2.1.0 Introduction

The certificate in VNF is for the logical functions of VNF. This clause discusses the certificate management and the relationship between certificate lifecycle and VNF lifecycle.

### 9.2.1.1 ID and certificate management in VNF

One VNF could have several functionalities and several logical interfaces, and one VNF could have several identities for different functionality, for different communication interface. According to the policies, different interfaces/functionalities could be assigned either same ID or different IDs. The ID and the functionalities/interfaces/communication-sessions could be a multiple-to-multiple mapping, as shown in figure 9.2.1.1-1.

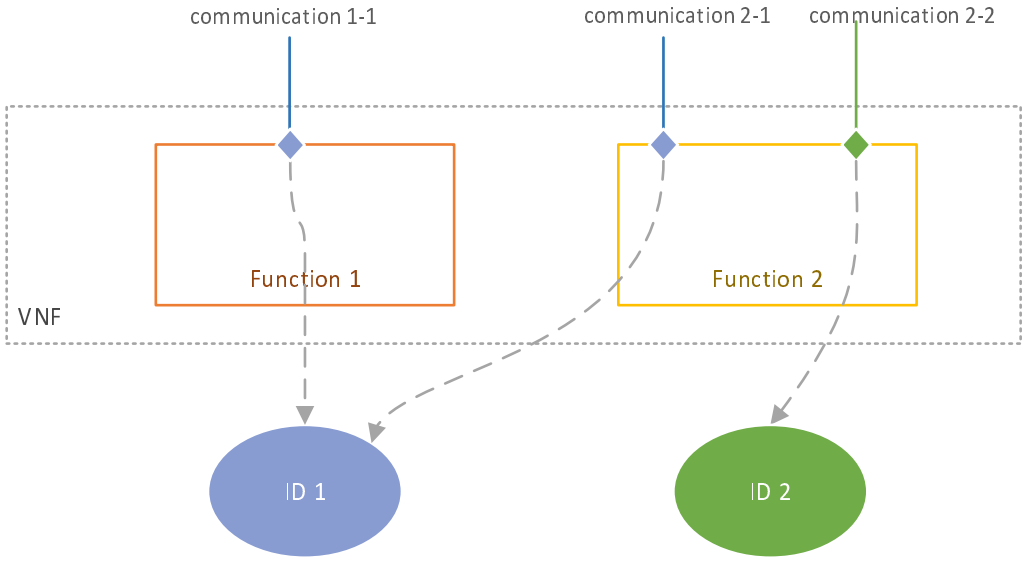


**Figure 9.2.1.1-1: Mapping between function and ID**

For example, the Mobility Management Entity (MME) in 3GPP network may connect to EMS, to eNodeB, to Serving Gateway (SGW), to other MME belonging to the same or different operators. All those connections could use a same ID or use different IDs in different interfaces.

A VNF consists of several VNF components (VNFC). A VNF instance is composed by several VNFC instances (VNFCI). A VNFC could be instantiated to multiple VNFCI.

A VNFCI refers to one realization of a defined VNFC. In current stage, a common understanding is *one VNFCI corresponds to one container or to one VM*, though there is no formal document to describe this mapping.

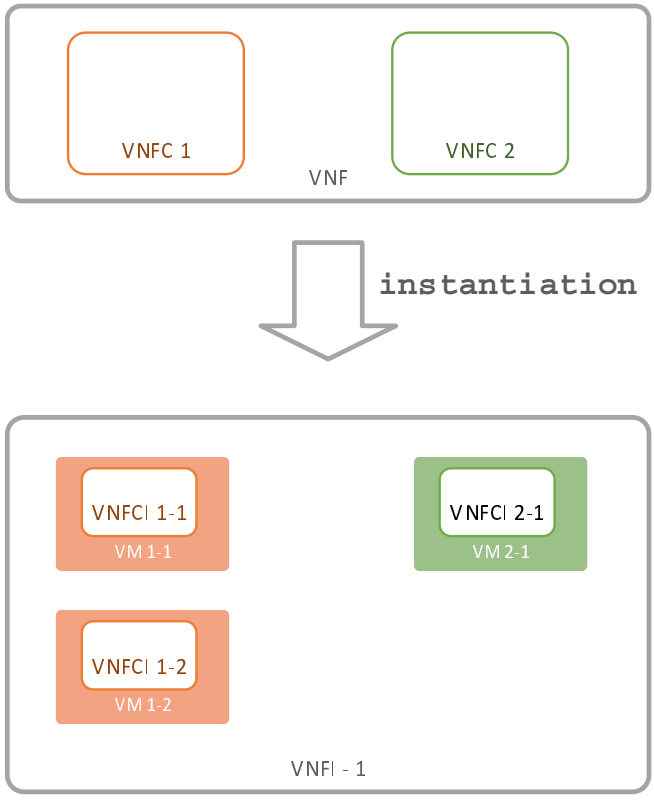


**Figure 9.2.1.1-2: VNF instantiation: one VNFC could have multiple VNFCI**

The ID in one VNF could be assigned to multiple functions, and each function could have multiple instances (VNFCI). Therefore, one ID will have multiple users, i.e. multiple VNFCI. There could be different policies for the binding between credential and ID. In the present document, the credential(s) bound to one ID are the certificate and the corresponding private key. The policies are:

**P1.** One ID have multiple credentials, each VNFCI using unique credential.

**P2.** One ID have one credential, all the VNFCI using same credential for one ID.

**P3.** One ID have multiple credentials, some of them are unique and some of them are bound to multiple VNFCI.

In case P2, the VNF only applies one certificate from CA/RA, for one ID. In case P1 and P3, the VNF needs to apply multiple certificate from CA/RA, for one ID. It is not restricted that one ID is bound to ONE certificate. IETF RFC 5280 [i.17] says:

*"A CA MAY issue more than one certificate with the same DN to the* ***same subject entity****."*

The *subject entity* equals to the ID being discussed in the present document.

The operator can select the policy. However, it should be emphasized that *issuing multiple certificate to the same entity* is an optional feature for the CA capacity.

**One certificate for one ID**

Figure 9.2.1.1-3 shows an example for the policy "one certificate one ID".

NOTE: This policy does not mean the certificate has to have multiple copies rather that in virtual environments, there are solutions that do not need to copy the certificate into each VNFCI.

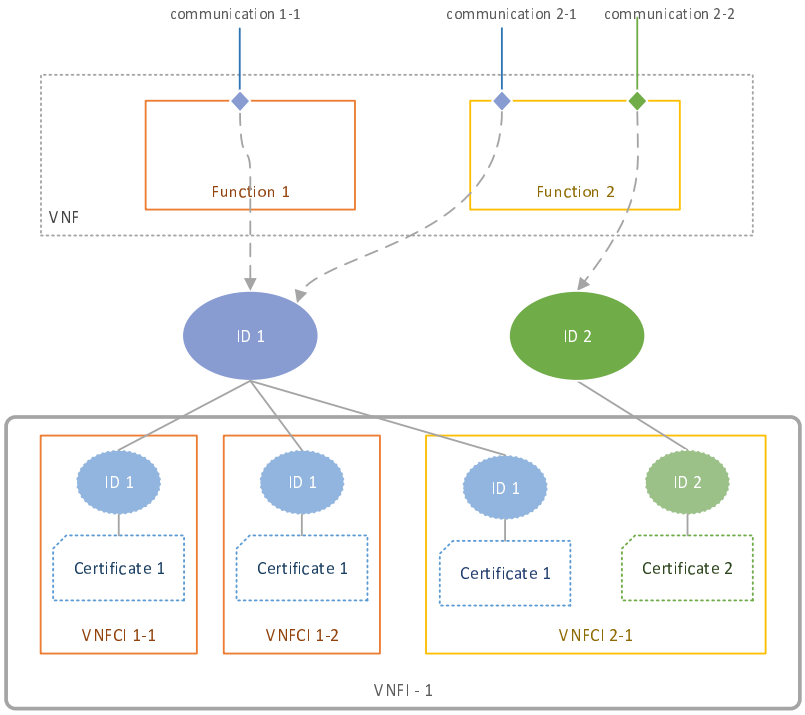


**Figure 9.2.1.1-3: Example of certificate sharing**

The traditional network NE has a secure internal physical environment. The communication between the components of one NE is typically regarded as a secure zone. The private key and certificate could be used by multiple components, that is, multiple physical boards. If only the private key is not disclosed on the network communication, the share is considered as safe.

In the virtual environment, the NE is distributed in the virtual environment and there is no physical boundary for internal communication. If the private key has multiple copies, the risk of key disclosure is increased: transfer the private key faces the communication attack, and, multiple copy or multiple access means one copy disclosure will threat all VNFCI using that certificate. Therefore, sharing private key and certificate should be carefully implemented.

Table 9.2.1.1-1 lists the requirement of implement certificate sharing in one VNF.

**Table 9.2.1.1-1: Requirement of implement certificate sharing in one VNF**

| | Risk | Mitigation |
|---|---|---|
| Confidential | • Disclosure of private key in communication<br>• Storage and memory hack to get private key | • Communication encryption when private key is transmitted<br>• Storage encryption in all VNFCI storing private key<br>• Storage encryption in network or platform storage |
| Integrity | • Storage modification | • Storage integrity protection in all VNFCI storing private key<br>• Storage integrity protection in network or platform storage |
| Access control | • Spoofing to get private key copy<br>• Communication message modification/replay | • Communication & storage authentication and authorization<br>• Communication integrity protection |
| Audit | • Repudiation | • Log |
| Availability | • DOS or DDOS (happens each request) | • Security domain for internal communication in VNF |

In one word, it should build a secure zone for the private key storage and transmission on the network. In the case of a VNF instance, the VNF instance is itself a trust domain, and any VNFCIs which comprise are therefore within the boundary of that trust domain. It follows that any communications and shared storage between those VNFCIs are expected to be protected at the level of the VNF instance, to maintain the trust boundary. There may, of course, be different levels of trust domains, including some with multiple VNFs. A trust domain is not congruent to the shared domain of a private key.

**Unique certificate for each ID in each VNFCI**

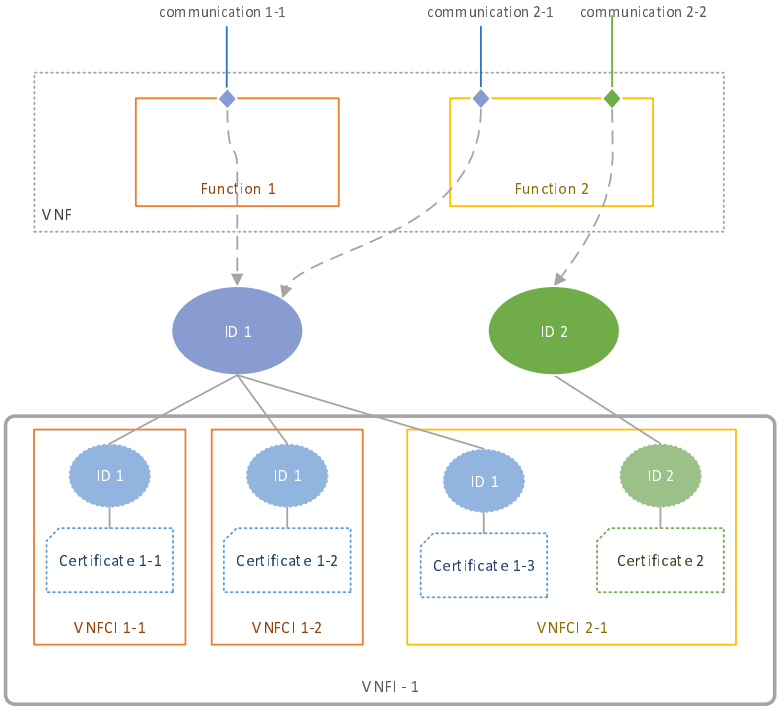


**Figure 9.2.1.1-4: Example of unique certificate for each VNFCI**

This policy has a significant advantage: the private key is held in each VNFCI and no inter-VNFCI communication to transfer any private keys. The life cycle of the certificate is along with the container VM lifecycle. When one instance is created on a new VM, e.g. VNF instantiation and scaling out, a new certificate is applied from CA. When one instance is terminated on a VM, e.g. VNF termination and scaling in, the certificate that stored on the VM is revoked from CA. When one instance is terminated and recreated in another location, e.g. VNF migration, the certificate in the old VM is revoked and a new certificate is applied for the instance in the new VM.

For the non-certificate sharing case, multiple certificates for one subject entity brings some complexities for management, because the "subject name" and "public key" is a one to many mapping. The only unique index in the certificate is the "issuer + serial number". This leads to the management complexity as follows:

a) Complexity in operation and maintenance, e.g.:

   i) Complex to configure the mapping between certificates and the functions, between certificates and function instances.

   ii) Configuration change or software upgrade may cause certificate application, update or revocation.

b) Complexity for certificate management function in VNF. The certificate lifecycle management should be distributed in all VNFCI that uses the certificate. Each VNFCI should support or partly support certificate initial application, update and revocation. At least, the private key generation, certificate application file generation and signature should be supported.

c) Complexity and inefficiency to NFV dynamic orchestration. Each time the NFV scales and each time the NFV container migrates, the certificate is expected to be either initially applied or revoked. The application and revoke are network communication and will lead to time latency for the orchestration.

d) Complexity in updating/revoking a certificate. The CA/RA workload is increased and the CA/RA is expected to support an "optional feature" in IETF standard, i.e. *A CA MAY issue more than one certificate with the same DN to the **same subject entity.***

### 9.2.1.2 Certificate lifecycle and VNF lifecycle

Certificates have their lifecycles, i.e. certification applying, certificate update and certificate revocation, as discussed in clause 8. A VNF has its lifecycle, i.e. VNF instantiation, scaling, update, upgrade and termination. If a certificate is assigned to one or multiple VNFCI, this certificate's lifecycle will have interaction to the VNF lifecycles. For example, one or multiple certificates should be applied from CA when the VNF instantiation is operated, and the corresponding certificates should be revoked from CA when one VNF instance is terminated.

The relation between VNF operation and VNF certificate management can be summarized in figure 9.2.1.2-1. Although the VNF migration is not purely a VNF operation, it is listed in this figure since this operation impacts the certificate lifecycle. In case of non-shared case, certificate management only happens between VNF and CA, as discussed in clause 8. In case of certificate sharing case, the certificate storage and sharing are also impacted by the VNF management.

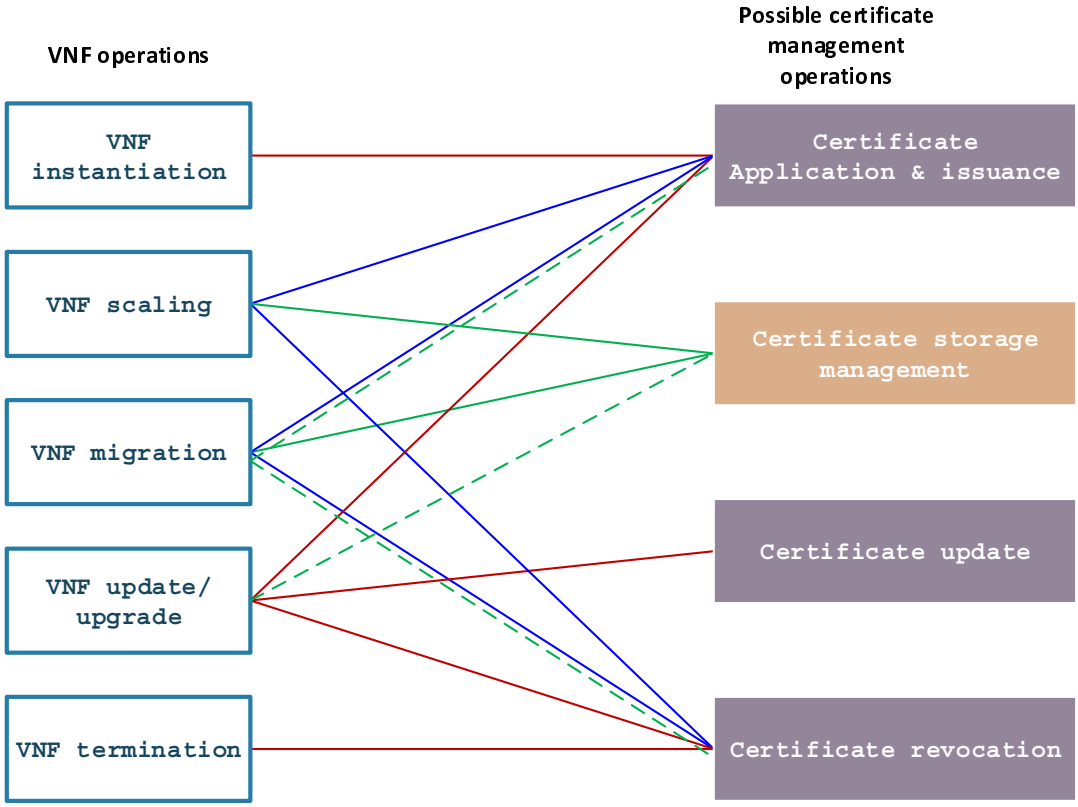


**Figure 9.2.1.2-1: The relation between VNF status and VNF certificate operations**

In figure 9.2.1.2-1, the green line is applicable in "certificate sharing case", the blue line is applicable to "non-certificate sharing case", and the red line is applicable to both cases. The dashed green line is optional in some implementations or some cases.

The following clause will explain figure 9.2.1.2-1 case by case.

### 9.2.1.3 VNF instantiation

When an VNF is instantiated, all the certificate required are applied from CA. If the policy P1 (non-sharing) is deployed, each VNFCI should apply its certificate from CA, even they are assigned to the same ID. If policy P2 (certificate sharing) is deployed, only one certificate application is proceeded, normally by the "certificate-managing VNFCI". Other VNFCI shares the private key and certificate via various solutions.

### 9.2.1.4 VNF scaling

During the lifecycle of a VNF, the VNF Management functions may monitor KPIs of a VNF, if such KPIs were captured in the deployment template. The management functions may use this information for scaling operations. Scaling may include changing the configuration of the virtualised resources (scale up, e.g. add CPU, or scale down, e.g. remove CPU), adding new virtualised resources (scale out, e.g. add a new VM), shutting down and removing VM instances (scale in), or releasing some virtualised resources (scale down).

The scaling action and corresponding certificate operations as shown in table 9.2.1.4-1.

**Table 9.2.1.4-1**

| VNF scaling action | Possible VNF certificate operations (certificate sharing) | Possible VNF certificate operations (certificate not sharing) |
|---|---|---|
| **Scale UP:** increase virtualised resource, e.g. add CPU or memory **Scale down:** release virtualised resources from existing instances, e.g. remove CPU | The resource change is transparency to certificate storage and usage. | The resource change is transparency to certificate storage and usage. |
| **Scale out:** add new virtualised resources, e.g. add a new VM instances | **certificate storage management:** assign the certificate copy or access right to the newly created VNFCI. | **Certificate initialization:** newly created VNFCI applies new certificate from CA. |
| **Scale in:** release some virtualised resources, e.g. shut down and remove VM instances | **certificate storage management:** remove the certificate copy or access right from the obsoleted VNFCI. | **Certificate revocation:** the certificate used by obsoleted VNFCI is revoked from CA, by the VNFCI itself or by the certificate management function in VNFCI, or by the manager NE. |

### 9.2.1.5 VNF migration

VNF or part of VNF change the resources, e.g. the container VM. See table 9.2.1.5-1 for details.

**Table 9.2.1.5-1**

| VNF migration | Possible VNF certificate operations (certificate sharing) | Possible VNF certificate operations (certificate not sharing) |
|---|---|---|
| VNF or part of VNF change the resources, e.g. the container VM | • If the certificate-managing VNFCI is not migrated, change the share assignment: remove the copy or assign the access right to the new VNFCI and remove the copy or the access right from the obsoleted VNFCI.<br>• If the communication for context migration is safe, the certificate could be copied from old resources to new resources.<br>• If the sharing is via network storage, just re-mount the storage.<br>• Otherwise, apply from CA a new certificate for all migrated VNFCI and revoke the certificate used by obsoleted VNFCI. | Apply from CA each certificate for each migrated VNFCI. |

### 9.2.1.6 VNF update/Upgrade

VNF update/upgrade supports VNF software and/or configuration changes of various complexities, including addition of new functions, removal of current functions and modification of current functions. The function addition may need certificate and the function removal may obsolete certificate. The addition, removal and the modification may need to change some contents of existing certificate, e.g. the Key-usage or the validity period. The VNF certificate management should have corresponding actions in accordance with the VNF changes. See table 9.2.1.6-1 for details.

**Table 9.2.1.6-1**

| VNF update/upgrade | Possible VNF certificate operations (certificate sharing) | Possible VNF certificate operations (certificate not sharing) |
|---|---|---|
| Add new function that need certificate | • If required certificate does not available in VNFI, apply new one, otherwise.<br>• If required certificate exists in VNFI, but the contents do not satisfy the requirement, update the certificate and manage the sharing, otherwise.<br>• If required certificate exists in VNFI, manage the sharing. | • If required certificate does not available in VNFCI, apply new one, otherwise.<br>• If required certificate exists in VNFCI, but the contents do not satisfy the requirement, update the certificate and manage the sharing. |
| Remove a function that was assigned a certificate | • If the certificate is required by other function, manage the sharing<br>• If the certificate is not required by other function, revoke it. | • Revoke the certificate. |
| Change the function related to a certificate | • If the certificate does not satisfy the requirement of the function change, update.<br>• if the certificate is no longer needed by other functions in this VNFI, change the sharing or revoke.<br>• if the certificate is still used by other functions in this VNFI, check the contents of certificate. If the contents of the certificate need change, update. | • If the certificate does not satisfy the requirement of the function change, update.<br>• if the certificate is no longer needed by other functions in this VNFCI, revoke.<br>• if the certificate is still used by other functions in this VNFCI, check the contents of certificate. If the contents of the certificate need change, update. |
| Add or Remove a function without communication requirement | NULL | NULL |

### 9.2.1.7 VNF termination

NFV Orchestrator can receive a request to terminate an existing VNF instance. This request may be triggered by OSS or MANO. In this case, VNFI or MANO or OSS should invoke certificate revocation for the VNF before it is gracefully terminated. If VNF is terminated abnormally, the MANO or OSS should initiate a certificate revocation procedure as soon as it detects the abnormality of VNFI.

It is highly recommended to clear the local or network storage of the private key and the certificate before termination. The clearing could be done by VNF, if it is gracefully terminated, or by MANO if the VNFI is abnormally terminated. The clearing is very important because the PKI relies on CRL to prevent certificate misuse or abuse, but the CRL is periodically updated. Attack could success among the CRL update gaps, if someone could get the copy of private key and the certificate file.

## 9.3 Certificate Provisioning

In the traditional NE, certificates and associated keys could be provisioned by:

1) Manually copying the private key and certificate via USB.

2) Connect to the local maintenance port and copy certificate.

3) The NE is provisioned with a device certificate issued by vendor or by third- party entity:

   a) this certificate is used as the NE certificate;

   b) using this certificate as an initial credential to apply a new certificate.

4) The NE is provisioned with some credentials, e.g. token or pre-shared key, and the new certificate is applied using this credential to access CA.

All of those solutions could be implemented for the NFVI creation, e.g. provisioning the certificate for the switches, routers and the host OS, etc. If the VM is created to have direct access right to the physical interfaces, e.g. USB port or Ethernet port, those solutions are applicable for VNF layer certificates provisioning.

In VNF layer, solution 1 is not applicable, because neither the USB interface nor the manual operations are available in most cases. Solution 2 is not applicable, because the local maintenance port is not available in most cases. The device certificate is no longer available, since the vendor only delivers software package to the operator. The VM and the NE are created on virtual resources. And therefore solution 3 is not applicable. The solution 4 requires provisioning credentials via USB or local maintenance interface, thus it is not applicable for virtual environment.

Provisioning certificate in NFV needs to be automated and manual operations avoided as much as possible. Some of the possible solutions have been discussed in clause 8.1. The main philosophy is injecting credentials to the instance. If the credentials are certificate, they could be used as the formal certificate or they could be as the temporary credential to apply formal certificates from CA. It could be the VNFM or VIM to do the injection.

## 9.4 Trust list management

The certificate management systems have to be validated by VNFs, which can be realized by provisioning trust certificate lists or PSKs in those VNFs. Specifically, the trust certificate or PSK can be provisioned during instantiation procedure. And VNFs can use these provisioned information to validate the PKI system.

VNF should maintain a trust list of root CA certificate to validate the certificates issued by CA. The root CA certificate may be provisioned in the VNF during the VNF instantiation procedure, or be obtained using certificate management protocol, e.g. CMP. The VNF that accepts the root certificate during provision should authenticate the CA according to security policy.

If the root CA certificate is provisioned during instantiation procedure, it should be configured by VNFM and securely transmitted via VIM and NFVI, then injected to the VNF.

If the root CA certificate is provisioned using a certificate management protocol, there should be some mechanism to ensure VNF can trust the CA's identity, e.g. a PSK is shared between VNF and CA in order to validate that CA in the following certificate management protocol for the VNF.

# Annex A:
# Authors & contributors

The following people have contributed to the present document:

**Rapporteur:**

Fei Li, Huawei, Huawei

**Other contributors:**

Scott Cadzow, Cadzow

Anne-Marie Praden, Gemalto N.V.

Alex Leadbeater, British Telecommunications plc

Leslie Willis, British Telecommunications plc

Steve Goeringer, CableLabs

Michael Bilca, OTD

Massimiliano Pala, CableLabs

# History

| Document history | | |
|---|---|---|
| V1.1.1 | January 2019 | Publication |
| | | |
| | | |
| | | |
| | | |