ETSI GR NFV-IFA 044 V5.1.1 (2023-02)

Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Report on Flexible VNF Deployment

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-IFA044

Keywords
flexible deployment, NFV, orchestration,
parameterization, virtualisation, VNF

ETSI

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

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

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

Contents

Intellectual Property Rights

Essential patents

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

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

Trademarks

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

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

Foreword

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

Modal verbs terminology

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

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

1 Scope

The present document studies potential improvements on the NFV architecture or information model for providing an increased richness of VNF deployment options and, therefore, a higher VNF deployment flexibility.

The study analyses use cases and the limitations that arise with the current information model in relation to them. It studies possible solutions to address the use cases and provides a set of recommendations on enhancements of the NFV specifications.

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-IFA 011: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; VNF Descriptor and Packaging Specification".

[i.2] ETSI GS NFV-IFA 007: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification".

[i.3] ETSI GS NFV-IFA 008: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification".

[i.4] ETSI GS NFV-IFA 013: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Os-Ma-nfvo reference point - Interface and Information Model Specification".

[i.5] ETSI GS NFV-IFA 014: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Network Service Templates Specification".

[i.6] ETSI GS NFV-IFA 010: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Functional requirements specification".

[i.7] ETSI GR 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 GR NFV 003 [i.7] apply.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GR NFV 003 [i.7] apply.

4 Overview

4.1 Introduction

With the use of VNF deployment flavours a VNFD can be designed to support different deployment requirements and avoid the need to create multiple VNF packages.

In particular, deployment flavours can properly address different sizes of the VNF (e.g. different VNFC scaling ranges or use of VDUs of different capacity) for a set of deployment scenarios with different needs.

Deployment flavours can also be well suited to address a small number of variations of topology. For example, a VNF might support a high availability configuration described in one flavour and a basic configuration without high availability described in another flavour. Eventually, the deployment flavour describing the high availability mode can include some VNFCs not included in the other deployment flavour.

In the general case, topology variations are orthogonal to size related requirements, i.e. scaling ranges or VDU size related attributes. Thus, if a VNFD uses deployment flavours to address topology variations and also size related variations the number of necessary deployment flavours to address all possible deployment scenarios can easily become very large.

In these circumstances, when the number of deployment variations that a VNF is expected to support explodes, deployment flavours become unsuitable to describe them.

This occurs more and more often with the advent of cloud native VNFs consisting of reusable microservices. The same microservice can be used by different functions and thus it is possible that one VNF implements a number of functions, as they all may use common microservices. Different deployments would then select different combination of functions and the number of possible combinations grows exponentially with the number of functions included in the VNF.

Moreover, for a particular VDU, whether VM based or container based, different deployments can have different needs, for example in terms of memory size or number of cpus, in order to optimize its usage. While different deployment flavours could be defined to cater for a reduced number of values sets, it becomes prohibitive if the number of possible values sets increases, especially if the configurations of the VDUs are independent of each other.

4.2 Objectives

The objective of the present document is to study how to enable the parameterization of VNF deployment flavours in order to reduce the number that a VNFD needs to specify, in particular:

- study how to enable the selection of which VDUs to enable for a particular VNF;
- study how to enable specific value parameterization for a defined selection;

- study the impacts on the granting processes;
- study the impacts of the parameterized deployment of a VNF on the VNF lifecycle and the operations that can be done on said VNF;
- study the relation of the selection and value parameterization with the parameterization of the cloud native templates;
- study the impacts at the NS, e.g. impact for other VNFs and definition of NS.

The study analyses potential enhancements in:

- the VNFD and NSD;
- VNF LCM, lifecycle granting and NS LCM interfaces;
- functional behaviour;

in order to support the objectives listed above.

5 Use cases

5.1 General

The following clauses describe use cases that illustrate a more flexible VNF deployment, in other words, a higher degree of parameterization in the VNF instantiation operation than what is currently supported.

While the focus is on VNF instantiation, the use cases also consider other aspects of the VNF lifecycle.

5.2 Bundling of multiple network functions in a VNF

A VNF vendor implements various network functions using a set of microservices or software modules, where the same microservice is used by multiple network functions. Thus, it decides to bundle these network functions in the same VNF. At deployment time the service provider will determine which of the network functions will be enabled and which not.

The VNF "ComboVnf" implements six network functions: NF1-NF6.

NF1 is a basic or supporting function and is always enabled when the VNF is deployed.

The rest of the NFs are optional functions. Whether they are enabled or not in a particular deployment depends on the particular service provider scenario.

The NFs are composed of VDUs as shown in table 5.2-1.

Table 5.2-1: Example of network function composition

NF1	VDU-A, VDU-B
NF2	VDU-C, VDU-D
NF3	VDU-C, VDU-E
NF4	VDU-F, VDU-G
NF5	VDU-H
NF6	VDU-I, VDU-J

Given the large number of different NF combinations, e.g. (NF1, NF2, NF3), (NF1, NF6), (NF1, NF2, NF3, NF4, NF5, NF6), etc., the VNF vendor specifies the NF composition in the VNFD as VDU groups and allows the selection of which of the NFs are enabled in a particular deployment. NF1 is indicated as not selectable, as it is mandatory.

The VNFD defines one single deployment flavour where all VDUs participate. The scaling ranges defined in the deployment flavour for every scaling aspects are able to cater for all deployment scenarios.

Deployment

Static selection (NS descriptor based):

A service provider creates an NSD that includes ComboVnf in a configuration that uses NF2, NF5 and NF6. This additional VNF information related to the NF selection will be included in the NSD.

When instantiating the ComboVnf, as a result of an NS LCM operation, the information about selected NFs (NF2, NF5 and NF6) will be included in the Instantiate VNF operation in Or-Vnfm. It is not necessary to indicate NF1 as this is a mandatory function.

The VNFCs corresponding to VDU-A, VDU-B, VDU-C, VDU-D, VDU-H, VDU-I and VDU-J will be instantiated.

Dynamic selection (interface based):

A service provider creates an NSD that includes ComboVnf without specifying which NFs will be enabled.

When invoking an NS LCM operation that will trigger the instantiation of ComboVnf, OSS/BSS will provide the information of which NFs are required, e.g. NF2, NF5 and NF6, via Os-Ma-Nfvo NS LCM interface parameters.

NFVO will invoke the Instantiate VNF operation in Or-Vnfm providing the information about selected NFs (NF2, NF5 and NF6).

The VNFCs corresponding to VDU-A, VDU-B, VDU-C, VDU-D, VDU-H, VDU-I and VDU-J will be instantiated.

Change of selection

During VNF lifecycle the service provider decides to start using NF3. OSS/BSS sends an NS update operation including an updated NF selection for ComboVnf, this time consisting of NF3, NF2, NF5 and NF6. OSS/BSS optionally includes information about the desired scale level.

NFVO compares the new selection with the current one and determines that VDU-E is part of NF3 but is currently not enabled. NFVO issues a VNF LCM operation (e.g. a new update VNF operation) towards the VNFM to indicate the new selection.

The VNFCs corresponding to VDU-E are instantiated. The number of created instances is based on the scale level information provided in the NS Update operation or, if none provided, on the current instantiation level of the VNF.

5.3 Parameterizable VDUs

The Vnf "FlexSizeComponent" consists of 4 VDUs: VDU-A, VDU-B, VDU-C and VDU-D.

VDU-A includes a virtual block storage. The VNF implements several end-user services. One of the end-user services uses a significant amount of storage in comparison with the rest. Therefore, in deployment scenarios where this particular end-user service is enabled, it is beneficial to create the virtual block storage of VDU-A with a larger size than in all other cases.

This end-user service, however, does not impose any particular demands on the number of cpus or memory size of VDU-A, nor on other VDUs in the VNF.

For this purpose, the VNF vendor defines the 'sizeOfStorage' attribute of BlockStorageData in VDU-A in the VNFD as a configurable attribute.

Table 5.3-1: Example of a configurable attribute definition

Configurable attribute "VDU A block storage size"
Allowed values: [10 GiB .. 100 GiB] (in 10 GiB increments)

A configurable attribute is thus an attribute whose value will be determined at run-time instead of having a hardcoded value in the VNFD. However, the VNFD includes a default value.

Furthermore, VDU-B and VDU-C describe the VNFCs in charge of carrying out the heavier processing tasks of the VNF. The performance vs. resource utilization, i.e. costs, of the VNF can be optimized for different deployment scenarios by selecting one of three predefined sets of values for the number of cpus and memory size of these two VDUs.

For this purpose, the VNFD specifies a configurable attribute set with three predefined options.

Table 5.3-2: Example of a configurable attribute set definition

Configurable attribute set "VNF processing"	
Option 1	VDU-B numVirtualCpu: 1, virtualMemSize: 512 MiB
	VDU-C numVirtualCpu: 2, virtualMemSize: 512 MiB
Option 2	VDU-B numVirtualCpu: 4, virtualMemSize: 1024 MiB
	VDU-C numVirtualCpu: 4, virtualMemSize: 1024 MiB
Option 3	VDU-B numVirtualCpu: 4, virtualMemSize: 1024 MiB
	VDU-C numVirtualCpu: 8, virtualMemSize: 4096 MiB

A configurable attribute set is a list of options of predefined values for a set of attributes, usually that have some correlation among them. One of the options will be selected at run-time. A default option is included in the VNFD.

It can be noted that the "VDU A block storage size" configurable attribute and the "VNF processing" configurable attribute set are independent, i.e. any of the allowed values in the "VDU A block storage size" configurable attribute can be combined with any of the options in the "VNF processing" configurable attribute set.

VDU-D will use the same number of cpus, same memory size and same storage size in all deployment scenarios.

The VNFD defines furthermore three deployment flavours: "small", "large" and "large with high availability". Different scaling ranges differentiate the 'small" from the "large" and "large with high availability" flavours. The latter one in addition has a different topology which adds some functional redundancy and eliminates single points of failures.

All four VDUs participate in the three deployment flavours. The configurable attribute and the configurable attribute set are applicable to all three flavours.

Deployment

A service provider creates an NSD that includes the FlexSizeComponent VNFD.

When invoking an NS LCM operation that will trigger the instantiation of the FlexSizeComponent VNF, OSS/BSS will indicate the selected value of the "VDU A block storage size" configurable attribute and the selected option for "VNF processing" configurable attribute set.

NFVO will invoke the Instantiate VNF operation in Or-Vnfm providing the information about selected values for the configurable attribute and attribute set for VDU-A, VDU-B and VDU-C.

The VNFCs corresponding to VDU-A, VDU-B and VDU-C will be instantiated according to the selected options. The VNFCs corresponding to VDU-D will be instantiated according to the attribute values hardcoded in the VNFD.

Change of selection

During the VNF lifecycle the service provider activates the end-user service that has high storage consumption. In order to adapt the available storage to the service needs, OSS/BSS sends an NS Update operation including a new selected value for the "VDU A block storage size".

NFVO issues a VNF LCM operation towards the VNFM to indicate the new selection.

NOTE: The VNF LCM interface does not currently provide an operation enabling this procedure.

6 Operational aspects

This circumstance is likely to demand higher efforts from the VNF vendors related to testing activities, as it obviously remains the VNF vendor's responsibility to ensure the reliability of the VNF in all possible deployment scenarios which are allowed as a result of the higher parameterization of the VNF.

On the other hand, these enhancements open the possibility to enable or disable functions of a VNF and to tune its capacity and performance in a way that could potentially affect other VNFs in the NS. It is the responsibility of the service provider to perform these actions in a manner that allows the NS to fulfil its intended functionality and performance. Furthermore, the service provider has the means to, in the NSD, prevent that the consumer of the NS LCM interface modifies the selection of functions within the VNF or of its VDU capacity related attributes.

7 Potential solution and enhancements

7.1 General

The following clauses describe potential solutions for resolving the shortcoming identified in clause 5 and the possible enhancements in the current model (e.g. descriptors, interfaces, etc.) to realize such solutions.

7.2 Use case: bundling of multiple network functions in a VNF

7.2.1 Identification of gaps

The analysis of the use case leads to the identification of the following gaps:

1) The capability to define optional sets of VNFCs in the VNFD.

2) The capability to select at deployment time whether an optional set of VNFCs is deployed or not.

3) The capability to change the selection after deployment at a later point in time in the VNF lifecycle.

4) The capability to include in the NSD a selection of the optional sets of VNFCs of a constituent VNF to be deployed.

In the following clauses a more detailed analysis of these gaps is described as well as a potential solution to overcome them.

7.2.2 Optional sets of VNFCs in the VNFD

7.2.2.1 General

The deployment of an optional set of VNFCs within the full set that constitutes a VNF can be seen as realized by an intermediary grouping level between the VnfDf level and the VDU level as shown in Figure 7.2.2.1-1 where each optional set is represented by a Deployable Module (DM).

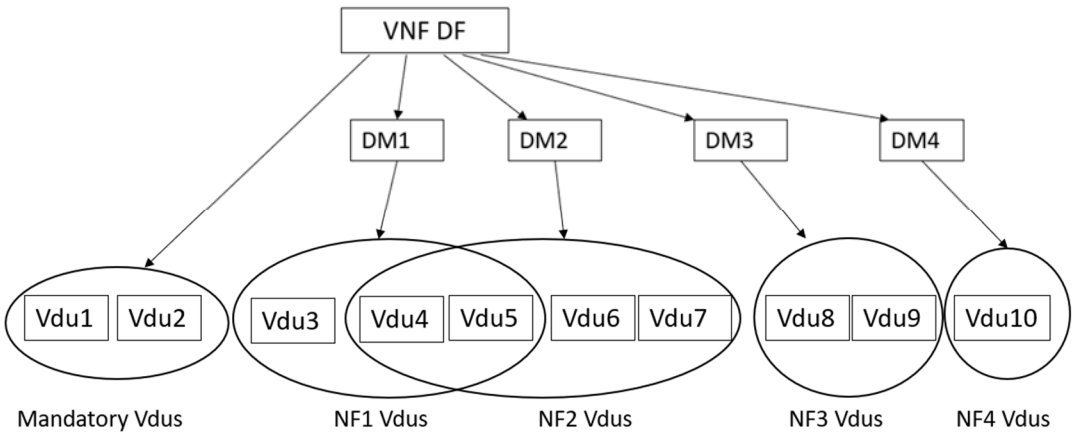

Figure 7.2.2.1-1: Optional VNF sets as deployment grouping level

An optional set of VNFCs has the following characteristics:

- It is specified in the VNFD, i.e. it is a design time decision, per VNF deployment flavour.
- A set consists of one or multiple VNFCs.
- A VNFC is part of zero, one or more sets.
- Sets are independent, i.e. the decision to deploy each set is independent of the other sets.
- From VNF perspective, the decision whether or not to deploy an optional set is taken at run time.

7.2.2.2 VNFD

A possible solution is to add a new IE in ETSI GS NFV-IFA 011 [i.1] to model this optional set, e.g. a "deployableModule". VduProfiles that participate in a VnfDf would optionally refer to one or more of these deployableModules. If a VduProfile does not refer to any deployableModule the corresponding VDU is deployed mandatorily in accordance with the requested instantiation level or scale info. If a VduProfile refers to one or more deployableModules the related VDU is only deployed if at least one of these deployableModules is selected.

As deployableModules represent independent functions, ScalingAspects will not span across deployableModules, i.e. they will not contain Vdus belonging to different deployableModules. A ScalingAspect is a group of VNFCs and other elements that are scaled together, usually because there is some correlation between the capacity required from the members of one ScalingAspect. Network functions, represented by deployableModules, have different scaling needs and defining a ScalingAspect that includes elements from more than one network function has the potential to produce undesirable effects in one network function when it is scaled based on the need of another network function. This also means that if a VDU belongs to several deployableModules, like Vdu4 and Vdu5 in Figure 7.2.2.1-1 which belong to DM1 and DM2, it will not be part of a ScalingAspect together with other VDUs that are not part of exactly the same DMs. In the example in Figure 7.2.2.1-1 the shared VDUs Vdu4 and Vdu5, if scalable, will be defined in their own ScalingAspect, either both VDUs in the same scalingAspect, or each VDU in a separate one.

The definition of deployableModules in a VNFD also has an implication in the composition of the MCIOPs. An MCIOP will only describe workloads related to VDUs that are included in the same deployableModule. A workload related to a VDU that is included in more than one DM will not be described in the same MCIOP as a workload related to another VDU that is not included in the same DMs. Workloads related to mandatory VDUs, i.e. which are not included in any deployableModule, and workloads related to VDUs which are included in a deployableModule will not be described by the same MCIOP. In the example represented in Figure 7.2.2.1-1 a minimum of six MCIOPs will be included to describe:

- MCIOP-1: workloads related to Vdu1 and Vdu2.
- MCIOP-2: workload related to Vdu3.
- MCIOP-3: workloads related to Vdu4 and Vdu5.
- MCIOP-4: workloads related to Vdu6 and Vdu7.
- MCIOP-5: workloads related to Vdu8 and Vdu9.
- MCIOP-6: workload related to Vdu10.

NOTE: The solution does not preclude a more fine-grained distribution of Vdus than the one listed above. For example, workloads related to Vdu1 and to Vdu2 might be included in MCIOP-1a and MCIOP-1b respectively.

If the VNFCs that constitute a deployableModule expect external connectivity in order to operate, either at least one of the VDUs belonging to that deployableModule will contain a VnfExtCpd or the external connectivity is realized by a VnfExtCpd belonging to a VDU that is not included in any deployableModule, i.e. a mandatory VDU. The external connectivity of a deployableModule cannot be realized through the external connectivity provided by another deployableModule, as this would create a dependency between the deployableModules.

The set of deployableModules may change between versions of a VNFD. For example, a VNF vendor may decide to merge two deployableModules (DM1 and DM2) that are separate in version n-1 of the VNFD into one single deployableModule DM1 in version n of the VNFD. Therefore, the solution will also include the possibility to specify in the ComponentMapping information element how deployableModules in the source VNFD map to deployableModules in the destination VNFD for a change current VNF package operation. For the same example, two mappings would be included for the change path from version n-1 to version n of the VNFD: from DM1 in source VNFD to DM1 in destination VNFD and from DM2 in source VNFD to DM1 in destination VNFD.

7.2.3 Selection of optional sets

7.2.3.1 General

The realization of the use case assumes a new capability to indicate in the VNF LCM interface whether each optional set is to be deployed or not.

This implies the ability to select components at a finer granularity than what is supported today. Today the selection is based on VNF deployment flavour and either an instantiation level or scale info.

The selected optional sets are primarily indicated in the Instantiate VNF operation, as it is desirable that the selection is applicable from the first instants of the VNF lifecycle. However, there is also a need to communicate a selection to the VNFM during any later point in time.

The selection is performed by the service provider and ultimately communicated by the VNF LCM interface consumer to the VNFM. There are several plausible ways how the NFVO can determine the selected modules to communicate to the VNFM: they are specified in the NSD as part of an NS deployment flavour, they are communicated by OSS/BSS in the NS LCM interface or they are introduced in the NFVO by means not specified in NFV, e.g. via a user interface. Consequently, two new capabilities are assumed:

- to indicate in the NS LCM interface a selection of optional sets for a constituent VNF: this selection is primarily to be indicated in the NS LCM operation that results in the instantiation of the VNF but is also useful in an NS LCM operation that results in a modification of the VNF;
- to indicate in the NSD, per NS deployment flavour, the desired optional sets of a constituent VNF.

Only the optional sets applicable to the selected VNF deployment flavour are eligible to be deployed.

7.2.3.2 VNF LCM interface

A possible solution to allow the VNF LCM interface consumer to communicate the selected optional sets of VNFCs is to add a new parameter to the Instantiate VNF operation in ETSI GS NFV-IFA 007 [i.2] and ETSI GS NFV-IFA 008 [i.3], e.g. 'selectedDeployableModules' and to add a new operation to allow a change of the selected sets, e.g. 'Change VNF optional sets'.

The number of instances of each VNFC is determined at VNF instantiation based on an instantiation level, either a default one or one indicated in the Instantiate VNF operation, or in scale info provided in the operation.

Instantiation levels encompass all VDUs participating in a VnfDf. When a VDU is not mandatory, i.e. belongs to one or more optional sets, and none of these sets is selected, the information in the applicable instantiation level, either received in the VNF LCM operation or the default instantiation level defined in the VNFD, concerning this VDU is ignored during instantiation, but stored in the VNFM in case that, at a later time, the selection is changed and this VDU has to be deployed.

Likewise for the scale info, which is an alternative way to express the number of instances at instantiation: if the scale info provided at VNF instantiation contains information, or in a later scale VNF operation, about a VDU that is not to be deployed, it is ignored but stored in the VNFM.

If, at a later time, the selection is changed by means of a new operation and the VDU is to be deployed, the operation that invokes the change of selected sets can contain new scale info concerning the VDUs to be deployed. If the operation does not contain scale info, the current scale level, as determined by the instantiate VNF operation, of the scaling aspect where the VDU belongs determines the number of instances

The solution will also support changing the selected optional sets in the Change Current VNF Package operation since different versions of a VNFD might define different optional sets of VNFCs.

The solution will also support changing the selected optional sets in the Change VNF flavour operation, as different sets might be defined for different flavours. For example, a VNF vendor might define a deployment flavour without any optionality.

7.2.3.3 NS LCM interface

A possible solution to allow the NS LCM interface consumer to communicate the selected optional sets of VNFCs when a VNF is instantiated is to add a new parameter, e.g. 'selectedDeployableModules', to those IEs in ETSI GS NFV-IFA 013 [i.4] related to a VNF to be created; e.g. ParamsForVnf, InstantiateVnfData and ParamsForNestedNs.

In addition a possible solution to allow to communicate the selected sets when changing the current VNF package, changing the VNF deployment flavour, changing the NS deployment flavour or associating a new NSD to an NS instance is to add the same new parameter to the related IEs ChangeVnfFlavourData, ChangeVnfPackageData, ChangeNsFlavourData and AssocNewNsdVersionData.

A possible solution for changing the selected optional sets of VNFCs of a constituent VNF in an NS instance at any time is to add a new task to the NS update operation, e.g. 'Change VNF optional sets'.

In all cases, in order to allow the user to specify the selected deployable modules per VNF instance, the selection will always apply to a particular VnfProfile, or a particular VNF instance, if that instance already exists. This is achieved by either placing the parameter in an IE that refers to a VnfProfile (e.g. ParamsForVnf, InstantiateVnfData) or by wrapping the parameter in a new IE that refers to a VnfProfile or a VNF instance.

7.2.3.4 VNF lifecycle operation granting interface

The selected optional sets, either indicated in the instantiation or in a change operation, are considered in the granting procedures as they affect the resource utilization.

7.2.3.5 NSD

A possible solution is to add a new attribute to the VnfProfile IE in ETSI GS NFV-IFA 014 [i.5] to specify the selected optional sets of a particular VNF for a particular NS deployment flavour.

In addition, in order to specify the selected optional sets of a particular VNF belonging to a nested NS, a possible solution is to add a new attribute to the NsProfile IE in ETSI GS NFV-IFA 014 [i.5].

7.2.3.6 VNFM interaction with CISM and VIM

When interworking with the CISM in order to install or upgrade MCIOPs, the VNFM will only install or upgrade those MCIOPs that describe mandatory VDUs and VDUs belonging to selected sets, regardless of which MciopProfiles are included in the VNF deployment flavour.

When interworking with the VIM in order to create resources, the VNFM will only create those related to mandatory VDUs and VDUs belonging to selected sets, regardless of which VduProfiles are included in the VNF deployment flavour.

7.3 Use case: parameterizable VDUs

7.3.1 Identification of gaps

The analysis of the use case leads to the identification of the following gaps related to the support of "configurable attributes":

1) The capability to specify in the VNFD for certain VDU attributes related to resource capacity requirements, e.g. storage size:

   a) that the attribute is configurable: the value is selected at deployment time and changeable during the VNF lifecycle;

   b) a range or a set of valid values.

2) The capability to select at deployment time the values for those VDU attributes specified as configurable in the VNFD.

3) The capability to change the values after deployment at a later point in time in the VNF lifecycle for those configurable attributes.

These gaps are addressed by the potential solutions described in the following clauses.

The use case also makes use of "configurable attribute sets". Configurable attributes and configurable attribute sets can be used independently. The realization of configurable attribute sets is based on the following capabilities:

1) The capability to specify in the VNFD a group of VDU attributes related to resource capacity requirements, belonging to one or to various VDUs, and to specify for that group:

   a) Two or more options, where each option consists of a list of fixed values, one for each of the attributes in the group.

2) The capability to select at deployment time one of the options defined in the VNFD, i.e. a list of fixed values for the group of attributes.

3) The capability to change to another option, of those defined in the VNFD, for a group of attributes at a later point in time in the VNF lifecycle.

The latter set of capabilities is supported within the actual framework if a different VNF deployment flavour is specified for each of the options, whereby different deployment flavours specify different values for the VDU attributes. Because of that they are not classified as gaps.

The number of configurable attributes sets as well as the number of options for one set is expected to be low in real cases. Therefore, the use of VNF deployment flavours is a viable method for providing this capability and no other solution is investigated in the present document. This is not the case for configurable attributes: just a few configurable attributes, each with a set of valid values, result in a large number of possible combinations which make the use of deployment flavours unviable for their modelling.

7.3.2 Specification of configurable attributes in the VNFD

VDU attributes used to express requirements related to resource capacity are:

In OsContainerDesc:

- requestedCpuResources
- requestedMemoryResources
- requestedEphemeralStorageResources
- cpuResourceLimit
- memoryResourceLimit
- ephemeralStorageResourceLimit
- extendedResourceRequests (only the part of the attribute indicating quantity)
- hugePageResources (only the part of the attribute indicating quantity)

In VirtualComputeDesc:

- virtualCpu.numVirtualCpus
- virtualMemory.virtualMemSize
- virtualDisk.sizeOfStorage

In VirtualStorageDesc:

- blockStorageData.sizeOfStorage

- objectStorageData.maxSizeOfStorage
- fileStorageData.sizeOfStorage

NOTE: The attributes above are based on the content of ETSI GS NFV-IFA 011 [i.1].

These attributes can be specified as configurable in the VNFD (in the case of the extendedResourceRequests and the hugePageResources only the part of the attribute that indicates the amount would be configurable).

A possible solution is to add for each of these attributes an optional sibling attribute that indicates a list of values or a range of values. The presence of this sibling parameter indicates:

- that the parameter is configurable, i.e. the actual value is not hardcoded in the VNFD, but is determined at deployment time;
- the valid values: expressed as a list or as a range.

For example, to be able to specify that the size of the block storage of a VirtualStorageDesc is configurable, a new attribute 'configurableSizeOfStorage' with two values indicating the range boundaries is added to BlockStorageData in the VNFD specification ETSI GS NFV-IFA 011 [i.1]. If a particular VirtualStorageDesc in a VNFD includes this attribute the user, i.e. the consumer of the VNF LCM interface, can determine the size at instantiation.

Whenever a VNFD defines an attribute as configurable, by means of presence of the sibling attribute that indicates valid values, a default value is also provided in the VNFD in the capacity related attribute that exists today. In other words, the sibling attribute is always accompanied by the existing capacity related attribute.

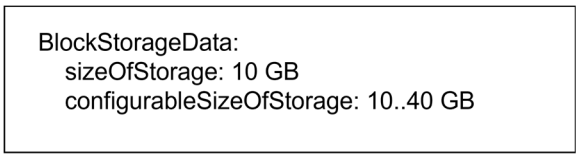

Figure 7.3.2-1: Size of storage as a configurable VDU attribute

Figure 7.3.2-1 shows how the sizeOfStorage in a VirtualStorageDesc of type block storage can be specified as configurable in the VNFD. When the VNFC is instantiated, the VNF LCM consumer can indicate the size of the storage, within the range specified in the descriptor. If the consumer does not indicate any value, the storage is instantiated with the size indicated as default by the sizeOfStorage attribute: 10 GB.

7.3.3 Selection of values for configurable VDU attributes

7.3.3.1 General

The realization of the use case assumes a new capability to indicate in the VNF LCM interface selected values for those VDU attributes related to capacity requirements that are declared as configurable in the VNFD.

The selected values are primarily indicated in the Instantiate VNF operation, as it is desirable that the selection is applicable from the first instants of the VNF lifecycle. However, the ability to communicate a selection to the VNFM during any later point in time is also expected.

The selection is performed by the service provider and ultimately communicated by the VNF LCM interface consumer to the VNFM. There are two plausible ways how the NFVO can determine the selected values to communicate to the VNFM: they are communicated by OSS/BSS in the NS LCM interface or they are introduced in the NFVO by means not specified in NFV, e.g. via a user interface.

Consequently, the new capability to indicate in the NS LCM interface selected values for configurable VDU attributes for a constituent VNF is also assumed. This selection is primarily to be indicated in the NS LCM operation that results in the instantiation of the VNF but is also useful in an NS LCM operation that results in a modification of the VNF.

NOTE: Theoretically it would also be possible to specify the selected values in an NSD, similar to the use case 'VNF as network functions bundle'. However, while in that use case the selection is more related to functionality (and thus it may be determined already at NSD design time) in this use case the selection is related to capacity and determining it at NSD design time restricts the applicability of the NSD. Therefore, this option is not pursued in this analysis.

Additional methods that allow the NFVO to determine the selected values are for further study. For instance, by means of policies evaluated and enforced by the NFVO.

7.3.3.2 VNF LCM

A possible solution to allow the VNF LCM interface consumer to communicate the selected values for configurable VDU attributes is to add a new parameter to the Instantiate VNF operation in ETSI GS NFV-IFA 007 [i.2] and ETSI GS NFV-IFA 008 [i.3], e.g. 'configurableCapacity'. The parameter will be structured per VDU, and within the VDU, per virtualComputeDesc or osContainerDesc and virtualStorageDesc.

At the virtualComputeDesc, osContainerDesc and virtualStorageDesc level, the 'configurableCapacity' parameter will indicate, for those attributes that are defined as configurable in the VNFD, the selected value. If no value is indicated, the default value indicated in the VNFD is used for the instantiation of the VNF.

This is illustrated in the following example. Figure 7.3.3.2-1 shows two VDUs of a containerized VNF. One of them consists of a single OS container descriptor, the second one of two OS container descriptors.

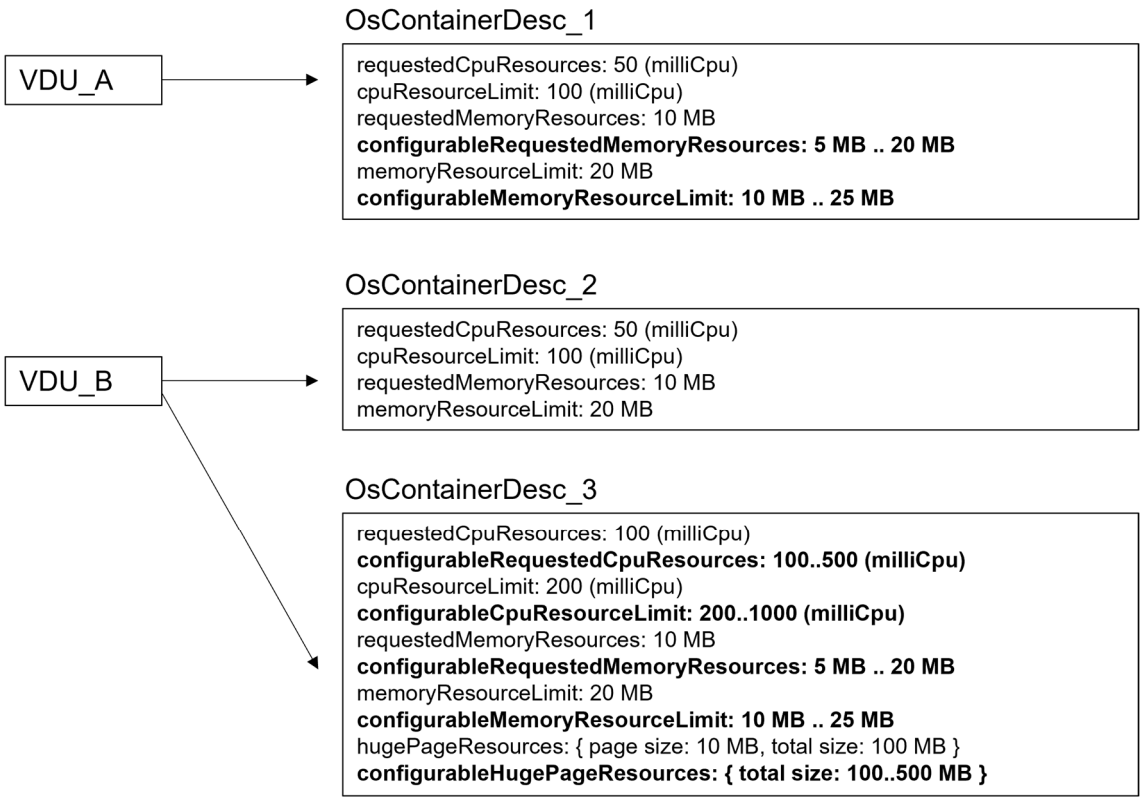

Figure 7.3.3.2-1: Configurable VDU capacity related attributes in the VNFD

Several of the attributes in the container descriptors are declared as configurable in the VNFD:

- In OsContainerDesc_1: requestedMemoryResources, memoryResourceLimit
- In OsContainerDesc_3: requestedCpuResources, cpuResourceLimit, requestedMemoryResources, memoryResourceLimit, hugePagesResources (total size)

while others have a single fixed value and are not configurable:

- In OsContainerDesc_1: requestedCpuResources, cpuResourceLimit

- In OsContainerDesc_2: requestedCpuResources, cpuResourceLimit, requestedMemoryResources, memoryResourceLimit

At instantiation of the VNF, the consumer of the interface includes the configurableCapacity parameter with the information shown in Figure 7.3.3.2-2.

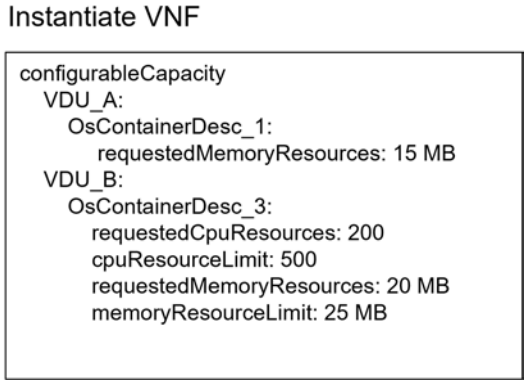

Figure 7.3.3.2-2: Selection of capacity values in the Instantiate VNF LCM operation

As a result, the VNF is instantiated with the attribute values indicated in Figure 7.3.3.2-3.

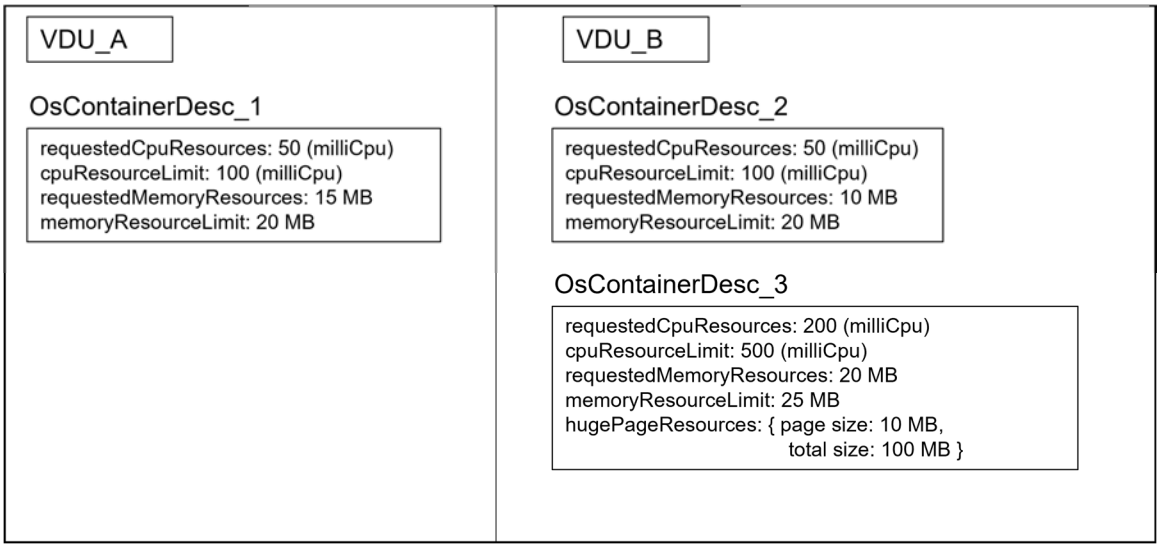

Figure 7.3.3.2-3: Capacity values applied in the instantiate operation

It can be observed that:

- configurable attributes for which a value is indicated in the instantiate VNF operation take this value:
  - In OsContainerDesc_1: requestedMemoryResources 15 MB
  - In OsContainerDesc_3: requestedCpuResources 200 milliCpu, cpuResourceLimit 500 milliCpu, requestedMemoryResources 20 MB, memoryResourceLimit 25 MB
- configurable attributes for which no value is indicated in the instantiate VNF operation take the default value indicated in the descriptor:
  - In OsContainerDesc_1: memoryResourceLimit 20 MB
  - In OsContainerDesc_3: hugePageResources { page size: 10 MB, total size: 100 MB}

- non configurable attributes take the value indicated in the descriptor:
  - In OsContainerDesc_1: requestedCpuResources 50 milliCpu, cpuResourceLimit 100 milliCpu
  - In OsContainerDesc_2: requestedCpuResources 50 milliCpu, cpuResourceLimit 100 milliCpu, requestedMemoryResources 10 MB, memoryResourceLimit 20 MB

It is worth noticing that there is a correlation between the values of some parameters, e.g. the value of cpuResourceLimit is supposed to be greater (or equal) than the requestedCpuResource value. For OsContainerDesc_3 these two values are configurable and the allowed range for the requestedCpuResource is not entirely smaller than the allowed range for the cpuResourceLimit: the upper boundary of the configurableRequestedCpuResources (500 milliCpu) is greater than the lower boundary of configurableCpuResourceLimit (200 milliCpu). The reason is to allow the user to set low cpu utilization limits when it also sets a low value for the cpu requested resources, e.g. requestedCpuResources: 200 milliCpus, cpuResourceLimit: 300 milliCpus. If the VNF LCM consumer provides values for these two attributes, it is its responsibility to indicate values that fulfill that condition (cpuResourceLimit greater or equal than requestedCpuResource).

In order to modify a value for a configurable attribute after VNF instantiation, a possible solution is to add a new VNF LCM operation 'Change configurable capacity' that contains the 'configurableCapacity' parameter introduced for the instantiate VNF operation.

NOTE: The potential solution for the use case "bundling of multiple network functions in a VNF" also proposes a new VNF LCM operation 'Change VNF optional sets' to modify the selection of optional sets during the VNF lifecycle in clause 7.2.3.2. The same operation with a more generic name can be used also for this purpose.

Another possible solution is to use the Scale VNF operation to convey the 'configurableCapacity' parameter.

The 'configurableCapacity' parameter would also be added to Change VNF Flavour and Change Current VNF Package operations.

7.3.3.3 NS LCM interface

A possible solution to allow the NS LCM interface consumer to communicate the selected values of configurable capacity related attributes when a VNF is instantiated is to add a new parameter, e.g. 'configurableCapacity', to those IEs in ETSI GS NFV-IFA 013 [i.4] related to a VNF to be created; e.g. ParamsForVnf, InstantiateVnfData and ParamsForNestedNs.

Figure 7.3.3.3-1 shows how the selected values are communicated in the NS LCM interface, in this case as part of the Instantiate NS operation, for a to be created VNF instance.

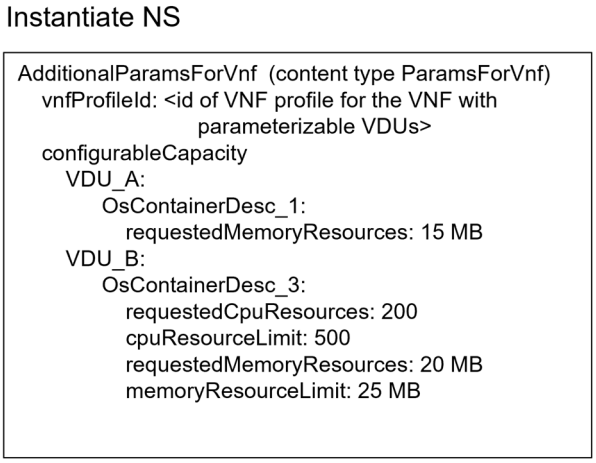

Figure 7.3.3.3-1: Selection of capacity values in the Instantiate NS operation in the NS LCM interface

In addition, a possible solution to allow to communicate the selected attribute values when changing the current VNF package, changing the VNF deployment flavour, changing the NS deployment flavour or associating a new NSD to an NS instance is to add the same new parameter to the related IEs ChangeVnfFlavourData, ChangeVnfPackageData, ChangeNsFlavourData and AssocNewNsdVersionData.

A possible solution for changing the selected VDU attribute values of a constituent VNF in an NS instance at any time is to add a new task to the NS update operation, e.g. 'Change VNF capacity'.

In all cases, in order to allow the user to specify the configurable capacity per VNF instance, the selection will always apply to a particular VnfProfile, or a particular VNF instance, if that instance already exists. This is achieved by either placing the parameter in an IE that refers to a VnfProfile (e.g. ParamsForVnf, InstantiateVnfData) or by wrapping the parameter in a new IE that refers to a VnfProfile or a VNF instance.

NOTE: The potential solution for the use case "bundling of multiple network functions in a VNF" also proposes a new task 'Change VNF optional sets' to the NS update operation to modify the selection of optional sets during the NS lifecycle in clause 7.2.3.3. The same task with a more generic name can be used also for this purpose.

Another possible solution is to use the Scale NS operation to convey the 'configurableCapacity' parameter as part of ScaleVnfData.

7.3.3.4 VNF lifecycle operation granting interface

The selected values for the VDU configurable capacity related attributes, either indicated in the instantiation or in a change operation, are considered in the granting procedures as they affect the resource utilization.

8 Recommendations

8.1 Recommendations on functional behaviour

This clause provides recommendation for new requirements on functional behaviour.

Table 8.1-1: Recommendations on new functional requirements on VNFM

Identifier	Recommendation	Comment
Vnfm.DepMod.001	It is recommended to specify a requirement on VNFM to support the capability to modify the set of instantiated VNFCs (adding VNFCs that were not instantiated or removing VNFCs that were instantiated) during the lifetime of a VNF to match those VDUs that are part of the selected deployableModules.	See clause 7.2.3.2.
Vnfm.ConfAttr.001.	It is recommended to remove the restrictions in notes 1 and 2 in table 7.2.3-1 of ETSI GS NFV-IFA 010 [i.6] which state that scaling up and down are not supported in the present release.	See clause 7.3.3.2. Regardless of whether the scale operation is used to modify VDU attribute values related to resource capacity or a new VNF LCM operation is introduced for that purpose, the vertical scaling behaviour would be de facto supported.

Table 8.1-2: Recommendations on new functional requirements on NFVO

Identifier	Recommendation	Comment
Nfvo.DepMod.001	It is recommended to specify a requirement on NFVO to support the capability to request the modification of the set of instantiated VNFCs (adding VNFCs that were not instantiated or removing VNFCs that were instantiated) during the lifetime of a VNF to match those VDUs that are part of the selected deployableModules.	See clause 7.2.3.3.

8.2 Recommendations on descriptors

This clause provides recommendation for normative work on NFV descriptors.

Table 8.2-1: Recommendations related to VNF descriptors

Identifier	Recommendation	Comment
VNFD.DepMod.001	It is recommended to introduce the concept of 'deployableModule', with the meaning of an optional set of VDUs, in the VNF deployment flavour.	See clause 7.2.2.2 The recommended modelling consists of a new IE to represent a deployableModule and a new attribute in the VduProfile containing references to the IE.
VNFD.DepMod.002	It is recommended to introduce an attribute in the VNF deployment flavour IE to indicate in which operations the modification of the deployableModules is supported by the VNF.	Only the new operation referred to in Vnflcm.DepMod.002 in table 8.3-1, the change current VNF package operation and the change VNF flavour operation can be listed.
VNFD.DepMod.003	It is recommended to add the 'deployableModule' type to the list of values in the componentType attribute in the ComponentMapping information element.	See clause 7.2.2.2
VNFD.ConfAttr.001	It is recommended to introduce a sibling attribute for each VDU attribute related to resource capacity and model this attribute to allow the definition of a set of values or a value range depending on the nature of the attribute.	See clause 7.3.2
VNFD.ConfAttr.002	It is recommended to introduce an attribute in the VduProfile IE to indicate in which operations the modification of the values of VDU attributes related to resource capacity is supported by the VNF.	Only the scale VNF or the new operation referred to in Vnflcm.ConfAttr.002 in table 8.3-1, the change current VNF package operation and the change VNF flavour operation can be listed.

Table 8.2-2: Recommendations related to NS descriptors

Identifier	Recommendation	Comment
NSD.DepMod.001	It is recommended to introduce an attribute in the VnfProfile to list the selected deployableModules of the VNF for a particular NS deployment flavour.	See clause 7.2.3.5
NSD.DepMod.002	It is recommended to introduce an attribute in the NsProfile to list per constituent VNF of a nested NS the selected deployableModules of the VNF for a particular NS deployment flavour.	See clause 7.2.3.5
NSD.DepMod.003	It is recommended that the attribute described in NSD.DepMod.001 include a flag to indicate whether the selection is allowed to be overridden or not (by means of information in the NS LCM interface or by means of information in an NsProfile of a parent NSD when the VNF is part of a nested NS).	See clause 6.
NSD.DepMod.004	It is recommended that the attribute described in NSD.DepMod.002 include a flag to indicate whether the selection is allowed to be overridden or not (by means of information in the NS LCM interface).	See clause 6.
NSD.ConfAttr.001	It is recommended to introduce an attribute in the VnfProfile to indicate per VDU whether it is allowed or not to modify the values of its attributes related to resource capacity by means of information in the NS LCM interface.	See clause 6.

8.3 Recommendations on interfaces

This clause provides recommendation for normative work on interfaces.

Table 8.3-1: Recommendations on the VNF LCM interface

Identifier	Recommendation	Comment
Vnflcm.DepMod.001	It is recommended to add new parameters in the 'instantiate VNF' operation to convey the selection of deployableModules.	See clause 7.2.3.2.
Vnflcm.DepMod.002	It is recommended to add a new VNF LCM operation to support the modification of the selection (addition or removal) of deployableModules.	See clause 7.2.3.2. Adding or removing deployableModules imply changes in the resource utilization and in the VNF structure. Therefore, the use of the Modify VNF Information operation is not proposed for this purpose.
Vnflcm.DepMod.003	It is recommended to add new parameters in the 'change VNF flavour' operation to convey the selection of deployableModules.	See clause 7.2.3.2.
Vnflcm.DepMod.004	It is recommended to add new parameters in the 'change current VNF package' operation to convey the selection of deployableModules.	See clause 7.2.3.2.
Vnflcm.ConfAttr.001	It is recommended to add new parameters in the 'instantiate VNF' operation to convey selected values for VDU attributes related to resource capacity.	See clause 7.3.3.2.
Vnflcm.ConfAttr.002	It is recommended to add new parameters in the 'scale VNF' operation or to add a new VNF LCM operation to support the modification of selected values for VDU attributes related to resource capacity.	See clause 7.3.3.2. A solution based on the scale VNF operation would result in the support of VNFC vertical scaling. In case a new VNF LCM operation is introduced, whether or not to use the same VNF LCM operation as for requirement Vnflcm.DepMod.002 is left for the information model design phase.
Vnflcm.ConfAttr.003	It is recommended to add new parameters in the 'change VNF flavour' operation to convey selected values for VDU attributes related to resource capacity.	See clause 7.3.3.2.
Vnflcm.ConfAttr.004	It is recommended to add new parameters in the 'change current VNF package' operation to convey selected values for VDU attributes related to resource capacity.	See clause 7.3.3.2.

Table 8.3-2: Recommendations on the NS LCM interface

Identifier	Recommendation	Comment
Nslcm.DepMod.001	It is recommended to enhance existing parameters in the 'instantiate NS' operation to support conveying the selection of deployableModules on a per instance basis of the constituent VNFs.	See clause 7.2.3.3.
Nslcm.DepMod.002	It is recommended to enhance existing parameters in the 'instantiate NS' operation to support conveying the selection of deployableModules on a per instance basis of the constituent VNFs of a nested NS.	See clause 7.2.3.3.
Nslcm.DepMod.003	It is recommended to enhance existing parameters in the 'update NS' operation to support conveying the selection of deployableModules on a per instance basis of constituent VNFs to be instantiated and added to the NS as part of this operation.	See clause 7.2.3.3.
Nslcm.DepMod.004	It is recommended to enhance existing parameters in the 'update NS' operation to support conveying the selection of deployableModules on a per instance basis of constituent VNFs for which the flavour is changed as part of this operation.	See clause 7.2.3.3.

Identifier	Recommendation	Comment
Nslcm.DepMod.005	It is recommended to enhance existing parameters in the 'update NS' operation to support conveying the selection of deployableModules on a per instance basis of constituent VNFs for which the current VNF package is changed as part of this operation.	See clause 7.2.3.3.
Nslcm.DepMod.006	It is recommended to enhance existing parameters in the 'update NS' operation to support conveying the selection of deployableModules on a per instance basis of the constituent VNFs when the NS is associated to a new NSD version as part of this operation.	See clause 7.2.3.3.
Nslcm.DepMod.007	It is recommended to enhance existing parameters in the 'update NS' operation to support conveying the selection of deployableModules on a per instance basis of the constituent VNFs when the NS flavour is changed as part of this operation.	See clause 7.2.3.3.
Nslcm.DepMod.008	It is recommended to add a new update type to the 'update NS' operation to support the modification of the selection of deployableModules on a per instance basis of the constituent VNFs or the constituent VNFs of a nested NS.	See clause 7.2.3.3.
Nslcm.ConfAttr.001	It is recommended to enhance existing parameters in the 'instantiate NS' operation to support conveying selected values for VDU attributes related to resource capacity on a per instance basis of the constituent VNFs.	See clause 7.3.3.3.
Nslcm.ConfAttr.002	It is recommended to enhance existing parameters in the 'instantiate NS' operation to support conveying selected values for VDU attributes related to resource capacity on a per instance basis of the constituent VNFs of a nested NS.	See clause 7.3.3.3.
Nslcm.ConfAttr.003	It is recommended to enhance existing parameters in the 'update NS' operation to support conveying selected values for VDU attributes related to resource capacity on a per instance basis of constituent VNFs that are to be instantiated and added to the NS as part of this operation.	See clause 7.3.3.3.
Nslcm.ConfAttr.004	It is recommended to enhance existing parameters in the 'update NS' operation to support conveying selected values for VDU attributes related to resource capacity on a per instance basis of constituent VNFs for which the flavour is changed as part of this operation.	See clause 7.3.3.3.
Nslcm.ConfAttr.005	It is recommended to enhance existing parameters in the 'update NS' operation to support conveying selected values for VDU attributes related to resource capacity on a per instance basis of constituent VNFs for which the current VNF package is changed as part of this operation.	See clause 7.3.3.3.
Nslcm.ConfAttr.006	It is recommended to enhance existing parameters in the 'update NS' operation to support conveying selected values for VDU attributes related to resource capacity on a per instance basis of the constituent VNFs when the NS is associated to a new NSD version as part of this operation.	See clause 7.3.3.3.
Nslcm.ConfAttr.007	It is recommended to enhance existing parameters in the 'update NS' operation to support conveying selected values for VDU attributes related to resource capacity on a per instance basis of the constituent VNFs when the NS flavour is changed as part of this operation.	See clause 7.3.3.3.
Nslcm.ConfAttr.008	It is recommended to add a new update type to the 'update NS' operation, or to enhance existing parameters in the 'scale NS' operation to support the modification of selected values for VDU attributes related to resource capacity on a per instance basis of the constituent VNFs or the constituent VNFs of a nested NS.	See clause 7.3.3.3. In case a new update type to the 'update NS' operation is added, whether or not to use the same update type as for requirement Nslcm.DepMod.008 is left for the information model design phase.

Table 8.3-3: Recommendations on the VNF lifecycle operation granting interface

Identifier	Recommendation	Comment
VnfGrant.DepMod.001	It is recommended to add new parameters in the 'grant VNF lifecycle operation' operation to convey the deployableModules selected in the VNF lifecycle operation.	See clause 7.2.3.4.
VnfGrant.ConfAttr.001	It is recommended to add new parameters in the 'grant VNF lifecycle operation' operation to support conveying the values for VDU attributes related to resource capacity selected in the VNF lifecycle operation.	See clause 7.3.3.4.

9 Conclusions

The present document has illustrated use cases showcasing the introduction of higher flexibility in the VNF deployment by means of allowing parameterization of aspects of a VNF that today are hardcoded in the VNFD for a particular deployment flavour.

Cloud native descriptors provide this flexibility as they offer the corresponding parameterization capabilities. However, when these descriptors are used as part of the NFV framework, the flexibility has not been exploited so far. The introduction of the enhancements recommended in clause 8 will enable the use of these capabilities in the NFV framework.

Annex A:
Change History

Date	Version	Information about changes
10/2021	0.0.1	NFVIFA(21)000761r1, NFVIFA(21)000856, NFVIFA(21)000879r2
01/2022	0.0.2	NFVIFA(21)000979r3
01/2022	0.0.3	NFVIFA(21)001090r1
03/2022	0.0.4	NFVIFA(22)000181r2
07/2022	0.0.5	NFVIFA(22)000338r4
08/2022	0.0.6	NFVIFA(22)000104r1, NFV(22)000447r1, NFV(22)000448r1
09/2022	0.0.7	NFVIFA(22)000513r1, NFVIFA(22)000514, NFVIFA(22)000628r1, NFVIFA(22)000652r2, NFVIFA(22)000653r2, NFVIFA(22)000667r1, NFVIFA(22)000677r1,
11/2022	0.0.8	NFVIFA(22)000741, NFVIFA(22)000757r1, NFVIFA(22)000758r2, NFVIFA(22)000763r1, NFVIFA(22)000773, NFVIFA(22)000774r2
12/2022	0.0.9	NFVIFA(22)000843r3, NFVIFA(22)000844r1, NFVIFA(22)000850r1, NFVIFA(22)000882r1, NFVIFA(22)000895, NFVIFA(22)000896r1

History

Document history		
V5.1.1	February 2023	Publication