

VNFD Information Model Specification for Open-O

1.	Scope.....	3
2.	Terms, Definitions and Abbreviations.....	3
3.	VNF Information Mode.....	3
4.	Functional Requirements.....	4
4.1	VNFD.....	4
4.2	VLD.....	6
5.	Information Model Definition.....	6
5.1	VNFD.....	6
5.2	VDU.....	7
5.3	VduCpd.....	8
5.4	VnfVirtualLinkDesc.....	9
5.5	VirtualComputeDesc.....	10
5.6	VirtualStorageDesc.....	11
5.7	SwImageDesc.....	11
5.8	ConnectivityType.....	12
5.9	VnfdElementGroup(To Be Implemented).....	13
5.10	VnfcConfigurableProperties.....	13
5.11	AddressData.....	13
5.12	L3AddressData.....	14
5.13	RequestedAdditionalCapabilityData.....	15
5.14	VirtualMemoryData.....	15
5.15	VirtualCpuData.....	16
5.16	VirtualCpuPinningData.....	17

1. Scope

The scope of the present document is to describe the VNFD Information Model Specification.

2. Terms, Definitions and Abbreviations

For the purposes of the present document, the following abbreviations apply:

Abbreviation	Definition
CM	Conditional Mandatory
CO	Conditional Optional
CP	Connection Point
CPD	Connection Point Descriptor
DSL	Domain Specific Language
MANO	Management and Orchestration
MPLS	Multi-Protocol Label Switching
VL	Virtual Link
VLD	Virtual Link Descriptor
XOR	eXclusive OR

Table 2-1 Abbreviations

3. VNF Information Mode

The present document develops specifications for packaging of VNFs to be delivered to service providers, focusing on the holistic end-to-end view of the VNF Package lifecycle, from design to runtime, capturing development as well as operational views. The present document provides an analysis of end-to-end VNF Package lifecycle management operations based on use-cases and NFV Architectural Framework functional blocks.

A VNF Package contains all of the required files and meta-data descriptors required to validate and instantiate a VNF.

Standardized meta-data descriptors are required to:

- describe the NFV infrastructure resource requirements for a VNF in a service provider environment;
- describe design constraints and other dependencies in order for the VNF to successfully install, instantiate and terminate; and

describe VNF operational behaviour including VNF lifecycle events (e.g. scaling, upgrading). Standardized packaging and validation of VNFs is required to:

- provide a consistent, documented method for VNF providers to package VNFs;
- harmonize the service provider on-boarding process for VNFs coming from different VNF providers;
- ensure integrity, trust and auditability of a VNF Package;
- allow for a flexible and extensible VNF packaging structure that accommodates a wide variety of NFV infrastructure scenarios; and
- allow the packaged VNF-related meta-data to be interpreted and the packaged VNF to be instantiated in a wide variety of orchestration systems irrespective of technology choice or infrastructure environment.

A VNFD is a deployment template which describes a VNF in terms of deployment and operational behavior requirements. It also contains connectivity, interface and virtualized resource requirements.

4. Functional Requirements

4.1 VNFD

Table 4-3 specifies requirements to the templates for VNFD instances.

Numbering	Requirements Description
VNF_VNFD001	The VNFD shall support a description of deployment policies.
VNF_VNFD002	The VNFD shall support a description of required virtualisation containers in terms of e.g. amount, characteristics and capabilities for virtual CPUs and virtual RAM and virtual disks..
VNF_VNFD003	The description of a virtualisation container in the VNFD shall support a description of attached additional virtual devices and their characteristics and capabilities.
VNF_VNFD004	The description of a virtualisation container in the VNFD shall support a description of acceleration capabilities and characteristics.
VNF_VNFD005	The VNFD shall support a description of the minimum and maximum number of instances of each particular virtualisation container that conform to the VNF.
VNF_VNFD006	The VNFD shall support a description of the VNF internal connectivity, including the connectivity between virtualization containers, and associated connectivity resource requirements.
VNF_VNFD007	The VNFD shall support a description of one or more DFs to choose a particular variant of the VNF to be instantiated.

Numbering	Requirements Description
VNF_VNFD008	The VNFD shall support a description of parameters which can be monitored for the VNF after instantiation.
VNF_VNFD009	The VNFD shall support a description of parameters which can be configured for the VNF and whether the parameters can be configured after VNF instantiation.
VNF_VNFD010	The VNFD shall support a description of lifecycle events and related actions which can be performed for the VNF.
VNF_VNFD011	The VNFD shall support a description of metadata about the VNF product.
VNF_VNFD012	The VNFD shall support a description of metadata about placement of virtualisation containers relative to each other.
VNF_VNFD013	The VNFD shall support a description of the supported VNF instance scaling.
VNF_VNFD014	The VNFD shall support a description of rules for auto-scaling describing which actions shall be executed if a threshold for a monitoring parameter is crossed.
VNF_VNFD015	The VNFD shall support a description of metadata to determine if an EM is used for the VNF and parameters describing how to connect to the EM.
VNF_VNFD016	The VNFD shall support a description of metadata about dependencies between virtualisation containers.
VNF_VNFD017	The VNFD shall support a description of Service Availability Level (SAL) requirements for virtual resources on the underlying NFVI.
VNF_VNFD018	The VNFD shall support a description of parameters whose values have to be specified as input to the instantiation process.
VNF_VNFD019	The VNFD shall support metadata related to network addresses to be assigned to Connection Point(s) (CP).
VNF_VNFD020	The VNFD shall support the description of VNF indicators. See note.
VNF_VNFD021	The VNFD shall support a description of external CP supported by the VNF enabling connectivity with one or more external entities.
VNF_VNFD022	The description of a virtualisation container in a VNFD shall support a description of meta data about software image(s).
VNF_VNFD023	The VNFD shall provide the possibility to reference information elements via URLs e.g. to external files provided by the VNF provider.

Numbering	Requirements Description
VNF_VNFD024	The VNFD shall provide a reference to the VNFM(s) compatible with the VNF described in the VNFD.
<p>NOTE: VNF Indicators are information supplied by the VNF or the EM to provide some indication on the VNF behaviour. VNFM can use these indicators in conjunction with virtualised resource data to perform auto-scaling decisions or to trigger a VNF LCM script.</p>	

Table 4-3 specifies requirements to the templates for VNFD instances

4.2 VLD

Table 4-4 specifies requirements to the templates for VLD instances.

Numbering	Requirements Description
VNF_VLD001	A VLD shall provide the information about an internal VL.

Table 4-4 requirements of VLD instance

5. Information Model Definition

5.1 VNFD

Attribute	Qualifier	Cardinality	Content	Description
id	M	1	Identifier	Identifies this vnfd information element within a NSD.
vendor	M	1	String	Identifies the vendor of VNFD.
version	M	1	String	Identifies the version of the VNFD
vnfmInfo	M	1...N	String	Identifies VNFM(s) compatible with the VNF described in this version of the VNFD.
vdu	M	1...N	VDU	Virtualisation Deployment Unit.

Attribute	Qualifier	Cardinality	Content	Description
intVirtualLinkDesc	M	0...N	VnfVirtualLinkDesc	Represents the type of network connectivity mandated by the VNF provider between two or more CPs which includes at least one internal CP.
vnfExtCpd	M	1...N	VnfExtCpd	Describes network connectivity between VNF instances.

Table 5-1 Attributes of VNFD information element

5.2 VDU

Attribute	Qualifier	Cardinality	Content	Description
id	M	1	Identifier	Identifies the id of this vdu in VNFD.
name	M	1	String	Human readable name of the Vdu.
description	M	1	String	Human readable description of the Vdu.
intCpd	M	1...N	VduCpd	Describes network connectivity between a VNFC instance (based on this Vdu) and an internal Virtual Link (VL).
virtualComputeDesc	M	1	VirtualComputeDesc	Describes CPU, Memory and acceleration requirements of the Virtualisation Container realizing this Vdu.
virtualStorageDesc	M	0...N	VirtualStorageDesc	Describes storage requirements for a VirtualStorage instance attached to the virtualisation container created from virtualComputeDesc defined for this Vdu.
bootOrder	M	0...N	KeyValuePair	The key indicates the boot index (lowest index defines highest boot priority). The Value references a descriptor from which a valid boot

Attribute	Qualifier	Cardinality	Content	Description
				device is created e.g. VirtualStorageDesc from which a VirtualStorage instance is created. See note 1.
swImageDesc	M	0...1	SwImageDesc	Describes the software image which is directly loaded on the virtualization container realizing this Vdu.
nfviConstraint (To be implemented)	M	0...N	String	Describes constraints on the NFVI for the VNFC instance(s) created from this Vdu. For example, aspects of a secure hosting environment for the VNFC instance that involve additional entities or processes. See note 3.
monitoringParameter (To be implemented)	M	0...N	MonitoringParameter	Describes monitoring parameters of this VDU.
configurableProperties	M	0...N	VnfcConfigurableProperties	Describes the configurable properties of all VNFC instances based on this VDU.
<p>NOTE 1: If no boot order is defined the default boot order defined in the VIM or NFVI shall be used.</p> <p>NOTE 2: More software images can be attached to the virtualization container using VirtualStorage resources.</p> <p>NOTE 3: These are constraints other than stipulating that a VNFC instance has access to a certain resource, as a prerequisite to instantiation. The attributes virtualComputeDesc and virtualStorageDesc define the resources required for instantiation of the VNFC instance.</p>				

Table 5-2 Attributes of VDU information element

5.3 VduCpd

Attribute	Qualifier	Cardinality	Content	Description
id	M	1	Identifier	Identifier of this VduCpd information element.

Attribute	Qualifier	Cardinality	Content	Description
layer_protocol	M	1	String	Identifies which protocol the CP uses for connectivity purposes (Ethernet, MPLS, ODU2, IPV4, IPV6, Pseudo-Wire, etc.). See note.
cpRole	M	0..1	String	Identifies the role of the port in the context of the traffic flow patterns in the VNF or parent NS. For example a VNF with a tree flow pattern within the VNF will have legal cpRoles of ROOT and LEAF.
description	M	0..1	String	Human readable description of the VduCpd.
addressData	M	1	AddressData	Provides information on the addresses to be assigned to the CP(s) instantiated from this VduCpd.
vduld	M	0..1	Identifier (Reference to Vdu)	Reference to the internal VDU which is used to instantiate internal CPs. These internal CPs are, in turn, exposed as external CPs defined by this vdu.
virtualLinkDesc	M	0..1	Identifier (Reference to VnfVirtualLinkDesc)	Provides human-readable information on the purpose of the VL (e.g. control plane traffic).
virtualNetworkInterfaceRequirements	M	0..N	VirtualNetworkInterfaceRequirements	Specifies requirements on a virtual network interface realising the CPs instantiated from this VduCpd.
bitrateRequirement	M	0..1	Number	Bitrate requirement on this VduCpd P.
NOTE: This information determines, amongst other things, which type of address to assign to the access point at instantiation time.				

Table 5-3 Attributes of VduCpd information element

5.4 VnfVirtualLinkDesc

Attribute	Qualifier	Cardinality	Content	Description
id	M	1	Identifier	Identifies this CPD information element within a NSD.
description	M	0...1	String	Specifies human-readable information on the purpose of the connection point (e.g. connection point for control plane traffic).
virtualLinkDescFlavour	M	1..N	VirtualLinkDesc Flavour	Describes a specific flavour of the VL with specific bitrate requirements.
connectivityType	M	0...1	ConnectivityType	Describes the connectivity type of this VL.
testAccess	M	0..N	String	Specifies test access facilities expected on the VL (e.g. none, passive monitoring, or active (intrusive) loopbacks at endpoints.

Table 5-4 Attributes of VnfVirtualLinkDesc information element

5.5 VirtualComputeDesc

Attribute	Qualifier	Cardinality	Content	Description
id	M	1	Identifier	Identifies the id of this VLD in VNFD.
requestAdditionalCapabilities	M	0...N	RequestedAdditionalCapabilityData	Specifies requirements for additional capabilities. These may be for a range of purposes. One example is acceleration related capabilities.
virtualMemory	M	1	VirtualMemoryData	The virtual memory of the virtualized compute.
virtualCpu	M	1	VirtualCpuData	The virtual CPU(s) of the virtualised compute..

Table 5-5 Attributes of VirtualComputeDesc information element

5.6 VirtualStorageDesc

Attribute	Qualifier	Cardinality	Content	Description
id	M	1	Identifier	Identifies the id of this VLD in VNFD.
typeOfStorage	M	1	String	Type of virtualised storage resource (e.g. volume, object).
sizeOfStorage	M	1	Number	Size of virtualised storage resource (e.g. size of volume, in GB).
rdmaEnabled	M	0...1	String	Indicate if the storage support RDMA.
SwImageDesc	M	0...N	Identifier (Reference to a SwImageDesc)	Software image to be loaded on the VirtualStorage Resource created based on this VirtualStorageDesc.

Table 5-6 Attributes of VirtualStorageDesc information element

5.7 SwImageDesc

Attribute	Qualifier	Cardinality	Content	Description
id	M	1	Identifier	Identifies the id of this software image.
name	M	1	String	Name of this software image.
version	M	1	String	The version of this software image.
checksum	M	1	String	The checksum of this software image.
containerFormat	M	1	String	The container format describes the container file format in which software image is provided..
diskFormat	M	1	String	The disk format of a software image is the format of the underlying disk image.

Attribute	Qualifier	Cardinality	Content	Description
minDisk	M	1	Number	The minimal disk size requirement for this software image. The value of the "size of storage" attribute of the VirtualStorageDesc referencing this SwImageDesc shall not be smaller than the value of minDisk.
minRam	M	0..1	Number	The minimal RAM requirement for this software image. The value of the "size" attribute of VirtualMemoryData of the Vdu referencing this SwImageDesc shall not be smaller than the value of minRam.
size	M	1	Number	The size of this software image.
swImage	M	1	String	This is a reference to the actual software image. The reference can be relative to the root of the VNF Package or can be a URL.
operatingSystem	M	0..1	String	Identifies the operating system used in the software image. This attribute may also identify if a 32 bit or 64 bit software image is used.
supportedVirtualisationEnvironment	M	0..N	String	Identifies the virtualisation environments (e.g. hypervisor) compatible with this software image.

Table 5-7 Attributes of SwImageDesc information element

5.8 ConnectivityType

Attribute	Qualifier	Cardinality	Content	Description
layerProtocol	M	1	String	Identifies the protocol that the VL supports (Ethernet, MPLS, ODU2, IPV4, IPV6, Pseudo-Wire, etc.).

Attribute	Qualifier	Cardinality	Content	Description
flowPattern	M	0..1	String	Identifies the flow pattern of the connectivity (Line, Tree, Mesh, etc.).

Table 5-8 Attributes of ConnectivityType information element

5.9 VnfdElementGroup(To Be Implemented)

Attribute	Qualifier	Cardinality	Content	Description
id	M	1	Identifier	Identifies the id of this group in VNFD.
description	M	1	String	Human readable description of the group.
vdu	M	0...N	ConnectivityType	References to Vdus that are part of this group.
virtualLinkDesc	M	0...N	String	Provides human-readable information on the purpose of the VL (e.g. control plane traffic).

Table 5-9 Attributes of VnfdElementGroup information element

5.10 VnfcConfigurableProperties

Attribute	Qualifier	Cardinality	Content	Description
additionalVnfcConfigurableProperty	M	0...N	KeyValuePair	It provides VNFC configurable properties.

Table 5-10 Attributes of VnfcConfigurableProperties information element

5.11 AddressData

Attribute	Qualifier	Cardinality	Content	Description
addressType	M	0...N	Enum	Describes the type of the address to be assigned to the CP instantiated from the parent CPD.

Attribute	Qualifier	Cardinality	Content	Description
				Value: <ul style="list-style-type: none"> • MAC address. • IP address. • ... The content type shall be aligned with the address type supported by the layerProtocol attribute of the parent CPD.
l3AddressData	M	0...1	L3AddressData	Provides the information on the IP addresses to be assigned to the CP instantiated from the parent CPD. Shall be present when the addressType is IP address.

Table 5-11 Attributes of AddressData information element

5.12 L3AddressData

Attribute	Qualifier	Cardinality	Content	Description
iPAddressAssignment	M	1	Boolean	Specify if the address assignment is the responsibility of management and orchestration function or not. If it is set to True, it is the management and orchestration function responsibility.
iPAddressType	M	0...1	Enum	Define address type. Value: <ul style="list-style-type: none"> • IPv4 address. • IPv6 address. See note.
floatingIpActivated	M	1	Boolean	Specify if the floating IP scheme is activated on the CP or not.
numberOfIpAddresses	M	0...1	Integer	Minimum number of IP addresses to be assigned based on this L3AddressData information element.
NOTE: The address type should be aligned with the address type supported by the layerProtocol attribute of the parent VnfExtCpd.				

Table 5-12 Attributes of L3AddressData information element

5.13 RequestedAdditionalCapabilityData

Attribute	Qualifier	Cardinality	Content	Description
requestedAdditionalCapabilityName	M	1	String	Identifies a requested additional capability for the VDU. ETSI GS NFV-IFA 002 [i.1] describes acceleration capabilities.
supportMandatory	M	1	Boolean	Indicates whether the requested additional capability is mandatory for successful operation.
numaEnabled	M	0...1	String	Identifies the minimum version of the requested additional capability.
preferredRequestedAdditionalCapabilityVersion	M	0...1	String	Identifies the preferred version of the requested additional capability.
targetPerformanceParameters	M	1...N	KeyValuePair	Identifies specific attributes, dependent on the requested additional capability type.

Table 5-13 Attributes of RequestedAdditionalCapabilityData information element

5.14 VirtualMemoryData

Attribute	Qualifier	Cardinality	Content	Description
virtualMemSize	M	1	Number	Amount of virtual Memory (e.g. in MB).
virtualMemOversubscriptionPolicy	M	1	Integer	The memory core oversubscription policy in terms of virtual memory to physical memory on the platform. The cardinality can be 0 during the allocation request, if no particular value is requested.
numaEnabled	M	0...1	Boolean	It specifies the memory allocation to be cognisant of the relevant process/core allocation. The cardinality can be 0 during

Attribute	Qualifier	Cardinality	Content	Description
				the allocation request, if no particular value is requested.

Table 5-14 Attributes of VirtualMemoryData information element

5.15 VirtualCpuData

Attribute	Qualifier	Cardinality	Content	Description
cpuArchitecture	M	0...1	Enum	Describes the type of the address to be assigned to the CP instantiated from the parent CPD. Value: <ul style="list-style-type: none"> • MAC address. • IP address. • ... The content type shall be aligned with the address type supported by the layerProtocol attribute of the parent CPD.
numVirtualCpu	M	1	Integer	Provides the information on the IP addresses to be assigned to the CP instantiated from the parent CPD. Shall be present when the addressType is IP address.
virtualCpuClock	M	0...1	Number	Minimum virtual CPU clock rate (e.g. in MHz). The cardinality can be 0 during the allocation request, if no particular value is requested.
virtualCpuOversubscriptionPolicy	M	0...1	String	The CPU core oversubscription policy e.g. the relation of virtual CPU cores to physical CPU cores/threads. The cardinality can be 0 during the allocation request, if no particular value is requested.
virtualCpuPinning	M	0...1	VirtualCpuPinningData	The virtual CPU pinning configuration for the virtualised compute resource.

Table 5-15 Attributes of VirtualCpuData information element

5.16 VirtualCpuPinningData

Attribute	Qualifier	Cardinality	Content	Description
cpuPinningPolicy	M	1	Enum	Specify if the address assignment is the responsibility of management and orchestration function or not. If it is set to True, it is the management and orchestration function responsibility.
cpuPinningMap	M	0...1	KeyValuePair	If cpuPinningPolicy is defined as "static", the cpuPinningMap provides the map of pinning virtual CPU cores to physical CPU cores/threads. Cardinality is 0 if cpuPinningPolicy has a different value than "static".

Table 5-16 Attributes of VirtualCpuPinningData information element