



AutoDeploy Configuration File Constructs

This chapter provides information on the following topics:

- [Overview](#), page 1
- [Service Deployment \(service-deployment\)](#), page 2
- [Site Descriptor \(sited\)](#), page 2
- [AutoVNF Descriptor \(autovnfd\)](#), page 3
- [VNF Rack Descriptor \(vnf-rackd\)](#), page 5
- [Secure-token](#), page 6
- [VNF-package](#), page 6
- [VNF Descriptor \(vnfd\)](#), page 6
- [VNFM Descriptor \(vnfmd\)](#), page 11
- [NFVI POP Descriptor \(nfvi-popd\)](#), page 11
- [VIM Descriptor \(vimd\)](#), page 12
- [VIM Orchestrator Descriptor \(vim-orchd\)](#), page 12

Overview

Constructs within the AutoDeploy configuration file are based on the ETSI MANO architecture. These same constructs are used in the UWS service deployment configuration file as well.

The information in this section describes the relationships between the constructs used in the context of the configuration file (or UWS XML file) used by AutoDeploy to instantiate AutoVNF, the VNFM, and the USP-based VNF.

In general terms, each construct is comprised of a configuration parameters. These parameters are dictated by, and defined within the construct's respective YANG model. YANG models are available in the following directory on each of the UAS VMs:

```
ll /opt/cisco/usp/uas/models/
total 84
drwxr-xr-x 3 root root 4096 May 25 19:39 ./
drwxr-xr-x 9 root root 4096 May 25 19:39 ../
```

```

drwxr-xr-x 2 root root 4096 May 25 19:39 hash/
-rw-rw-r-- 1 root root 23338 May 25 19:33 usp-autodeploy.fxs
-rw-rw-r-- 1 root root 3779 May 25 19:33 usp-autodeploy-oper.fxs
-rw-rw-r-- 1 root root 14919 May 25 19:33 usp-autovnf.fxs
-rw-rw-r-- 1 root root 5097 May 25 19:33 usp-autovnf-oper.fxs
-rw-rw-r-- 1 root root 6631 May 25 19:33 usp-autovnf-vnfm.fxs
-rw-rw-r-- 1 root root 2028 May 25 19:33 usp-uas-oper.fxs
-rw-rw-r-- 1 root root 4524 May 25 19:33 usp-yang-types.fxs

```

Depending on your configuration, there may be multiple instances of some constructs. Additionally, some constructs are referenced by, or within, other constructs to form relationships that define the overall configuration for your deployment scenario.

Service Deployment (service-deployment)

Service-deployment is a list of sites deployed and orchestrated through UAS. Each site is referenced by its site descriptor.

Figure 1: service-deployment Construct



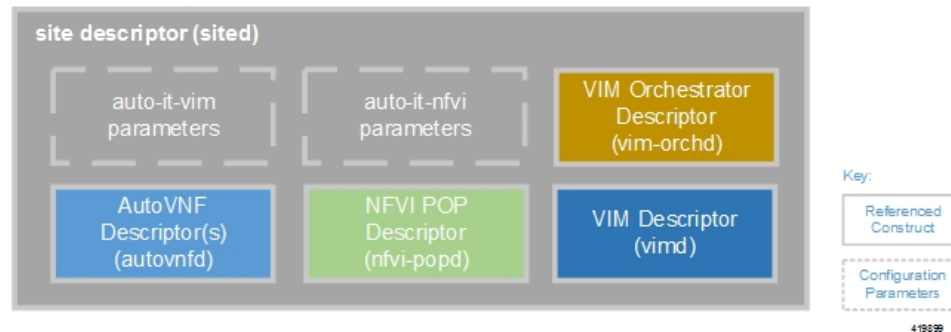
Site Descriptor (sited)

Sited is a collection of VIM parameters and a list of:

- AutoVNF descriptors (autovnfds): Each AutoVNF descriptor pertains to an individual VNF within the Site
- VIM Orchestrator descriptor (vim-orchd): Undercloud node
- NFVI Point-of-Presence descriptor (nfvi-popd): VIM node list

- VIM descriptor (vimd): VIM node configuration parameters

Figure 2: sited Construct



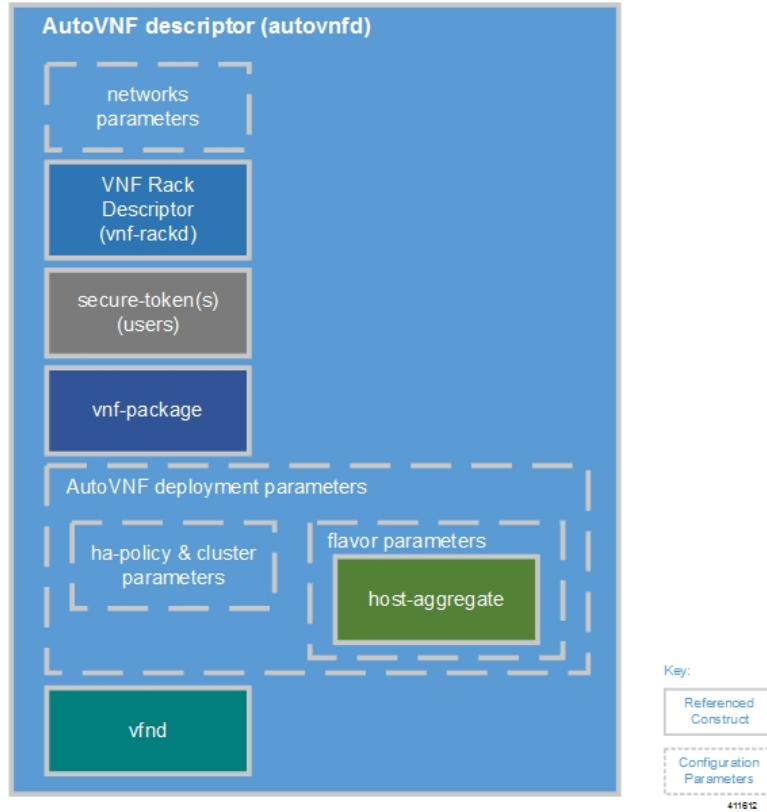
AutoVNF Descriptor (autovnf)

AutoVNF is the descriptor file that identifies the resources required for the VMs hosting the AutoVNF software function. It is a collection of:

- Network parameters
- VNF Rack Descriptor (vnf-rackd) listings
- Secure-token listings
- VNF-package listings
- AutoVNF deployment parameters including HA policy, cluster parameters, and flavor parameters

- VNF Descriptor (vnfd) listings

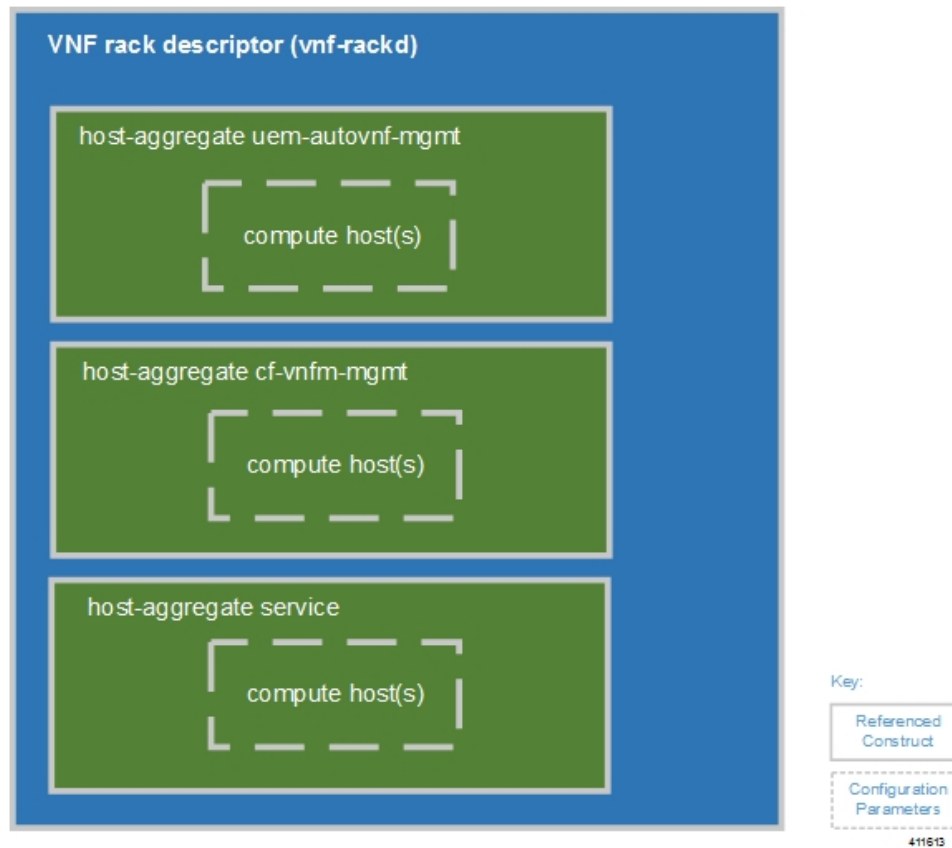
Figure 3: autovnf Construct



VNF Rack Descriptor (vnf-rackd)

A listing of the host aggregates which are used to identify the specific Compute Nodes on which the various VM functions (e.g. UEM, AutoVNF, CF, VNFM, and SFs) are to be deployed.

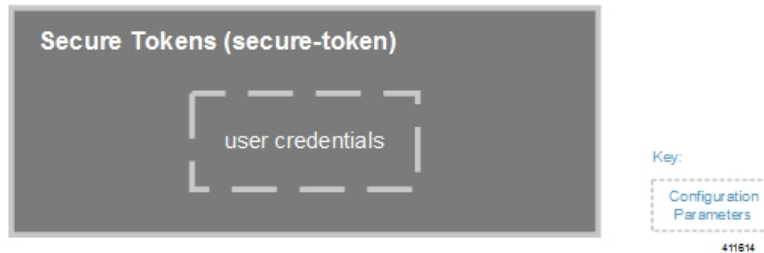
Figure 4: vnf-rackd Construct



Secure-token

Secure-token is used to create username/password pair and data is stored in encrypted form in CDB. These secure tokens can be referred using secure-id under VDU, deployment to pass authentication as needed.

Figure 5: secure-token Construct



VNF-package

VNF-package is used to identify the location of the specific ISO package and Day-0 configuration information to use when bringing up the VNF.

Figure 6: vnf-package Construct



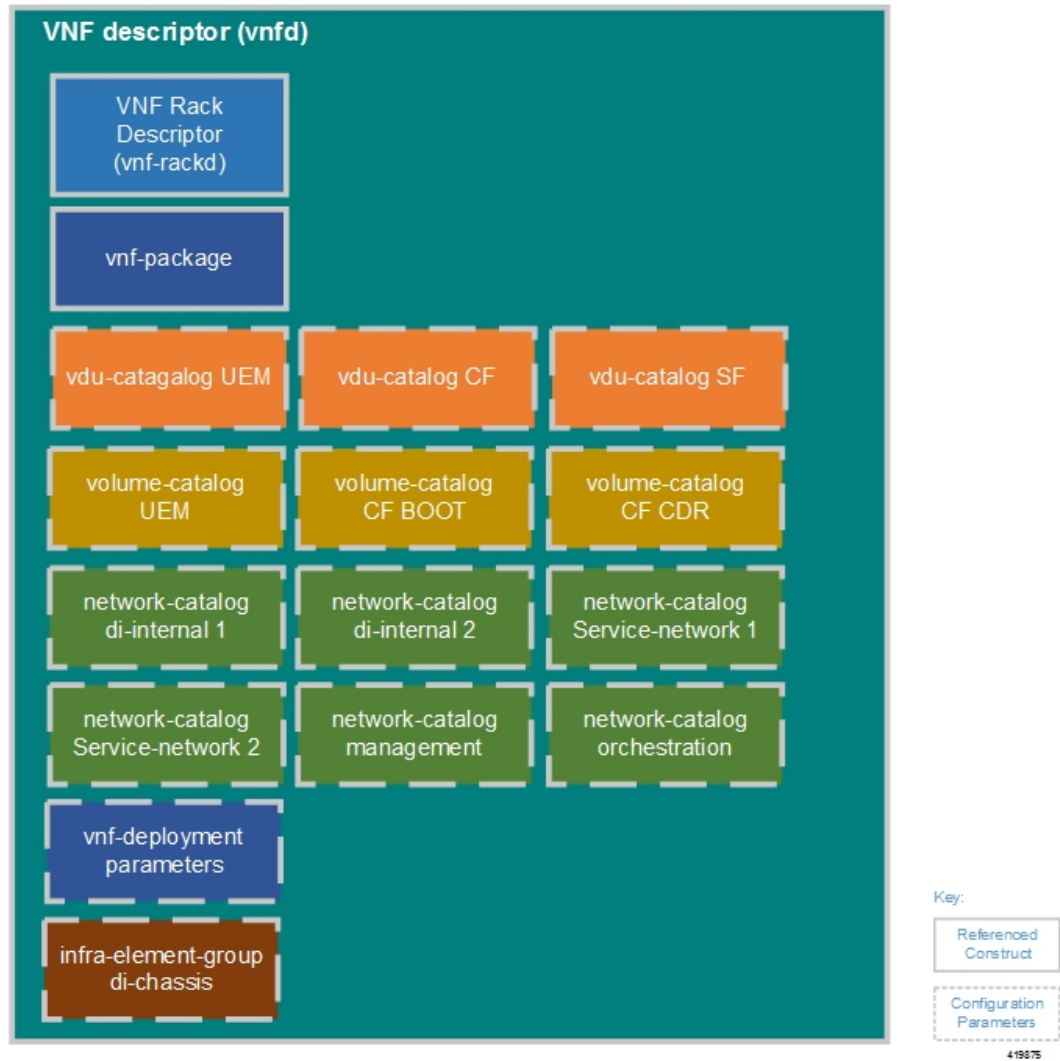
VNF Descriptor (vnfd)

VNFD is the descriptor file that identifies the resources required for the VMs hosting the components that comprise the USP-based VNF. It is a collection of:

- VNF Rack Descriptor (vnf-rackd) listings
- VNF-package listings
- Virtual descriptor unit (VDU) catalogs
- Volume catalogs
- Network catalogs
- VNF deployment parameters

- Infra-element groups

Figure 7: vnfd Construct



VDU Catalogs

Virtual descriptor unit (VDU) catalogs identify the physical resources and settings required for deploying the VMs that pertain to each of the specific USP-based VNF components.

Figure 8: vdu-catalog Construct

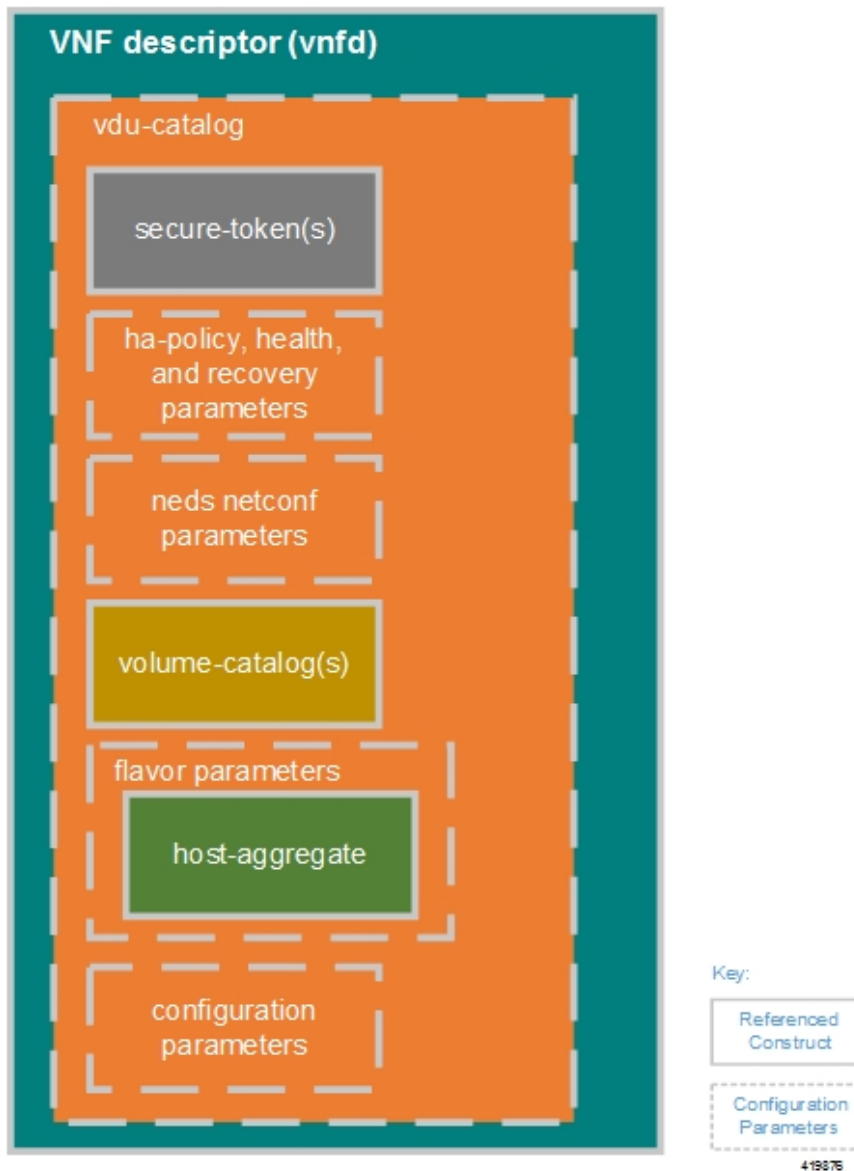


Figure 7: [vnfd Construct](#), on page 7 illustrates some of the common VDU catalogs required for USP-based VNFs:

- Ultra Element Manager (UEM)
- Control Function (CF)

- Service Function (SF)

Volume Catalogs

Volume catalogs identify the volume storage requirements for the USP-based VNF.

Figure 9: volume-catalog Construct

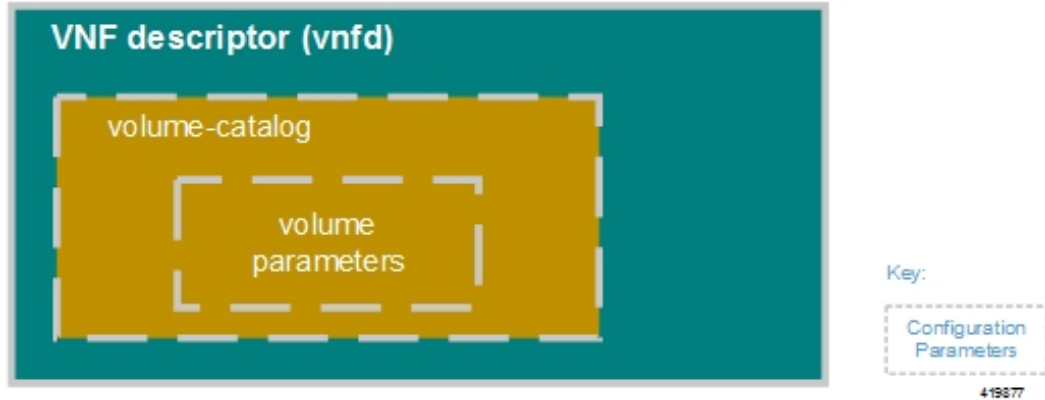


Figure 7: [vnfd Construct](#), on page 7 illustrates some of the common Volume catalogs required for USP-based VNFs:

- **Ultra Element Manager (UEM)**: Used to store operational data, KPI data, and logs.
- **Control Function (CF)**: Used to store VNF configuration files, the VNF boot images, KPI data, and logs.
- **CF Call Detail Record (CDR)**: Used to store customer billing records.

Network Catalogs

Network catalogs consist of the parameters that define the various networks required by the USP-based VNF.

Figure 10: network-catalog Construct

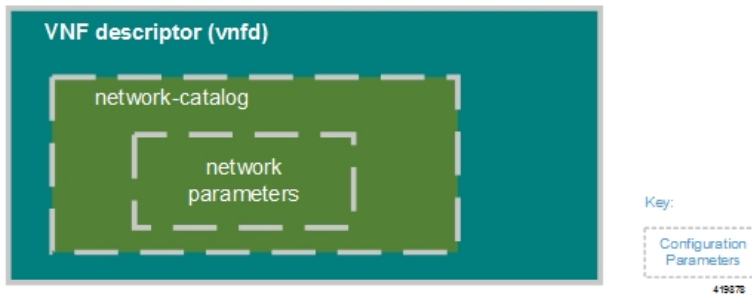


Figure 7: [vnfd Construct, on page 7](#) illustrates some of the common Network catalogs required for USP-based VNFs:

- **Di-internal:** This is the DI-internal network which serves as a "backplane" for CF-SF and CF-CF communications. Since this network is internal to the USP-VNF, it does not have a gateway interface to the router in the OpenStack network topology. A unique DI-internal network must be created for each instance of the USP-based VNF.
- **Service:** The service networks used to carry subscriber data traffic into and out of the SFs. In addition, service networks also connect SFs to other network elements required for subscriber data processing (e.g. AAA, billing, policy, etc.).
- **Management:** This is the local management network between the CFs and other management elements like the UEM and VNFM. This network is also used by the Ultra Manager to deploy the VNFM and AutoVNF. A common management network can be used for multiple instances of the USP-based VNF.
- **Orchestration:** This is the network used for VNF deployment and monitoring. It is used by the VNFM to onboard the USP-based VNF.

Infra Element Group(s)

Infra element groups define the specific parameters for the VDUs that comprise the element group and associates those VDUs with defined networks and interfaces.

Figure 11: *infra-element-groups Construct*

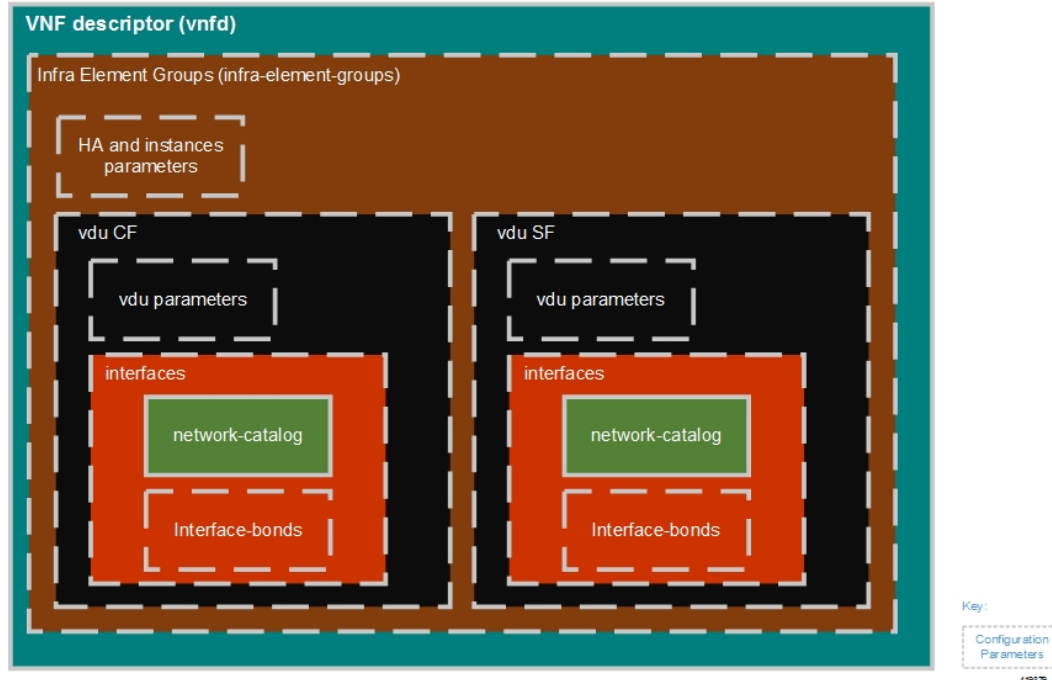


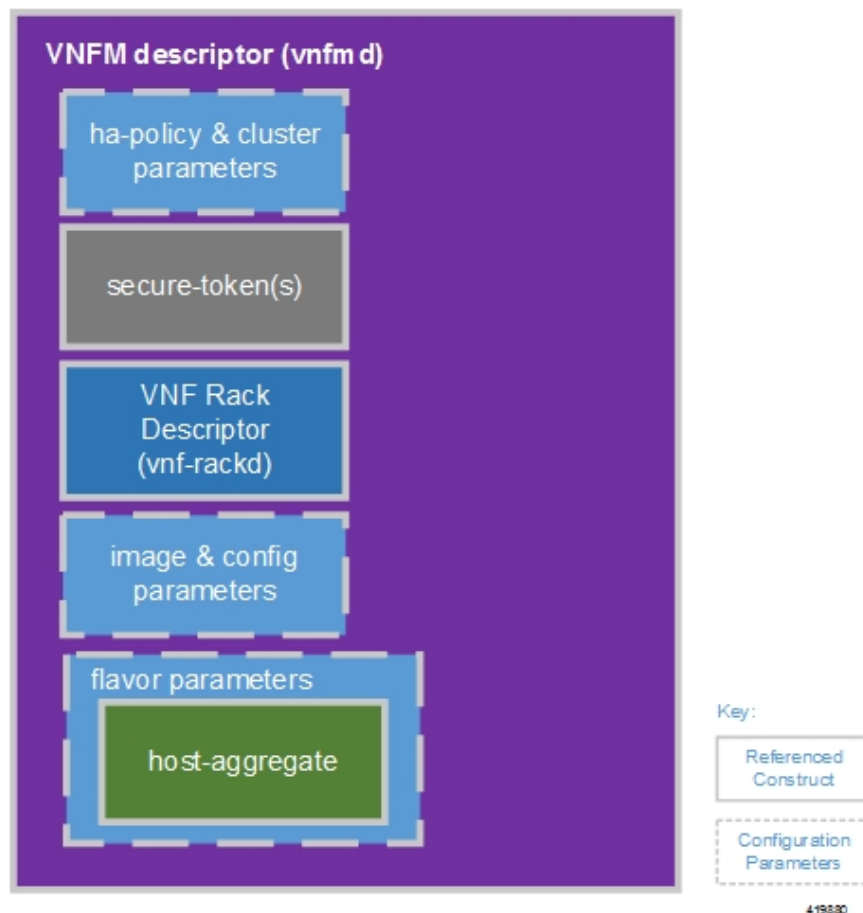
Figure 7: [vnfd Construct, on page 7](#) depicts a single infra-element-group called "di-chassis". VDUs are configured for VNFs such as the CF and SF as shown in [Figure 11: infra-element-groups Construct, on page 10](#).

VNFM Descriptor (vnfmd)

VNFM is the descriptor file that identifies the resources required for the VMs hosting VNFM. It is a collection of:

- HA and cluster parameters
- Secure-tokens
- VNF Rack Descriptor (vnf-rackd) listings
- VNFM deployment parameters

Figure 12: vnfmd Construct

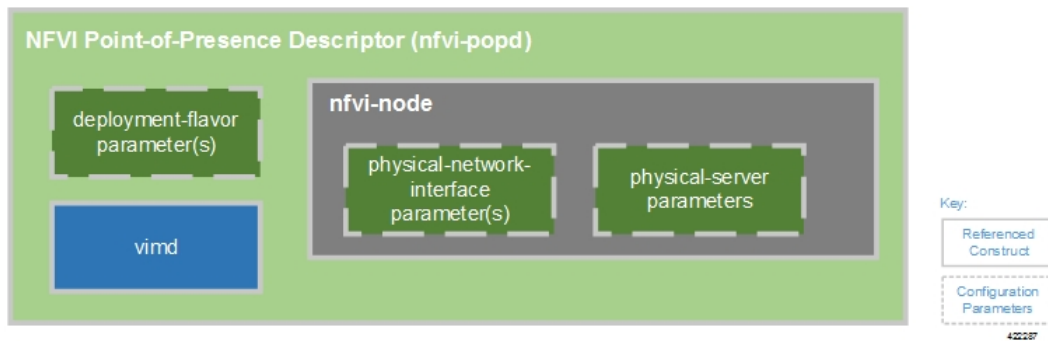


NFVI POP Descriptor (nfvi-popd)

The descriptor file that identifies network characteristics for all of the hardware nodes that comprise the NFVI. It is a collection of parameters on a per-node basis:

- Deployment flavor parameter(s)
- Virtual Infrastructure Manager (VIM) descriptor (vimd)
- Physical network interface parameter(s)
- Physical server parameters

Figure 13: nfvi-popd Construct

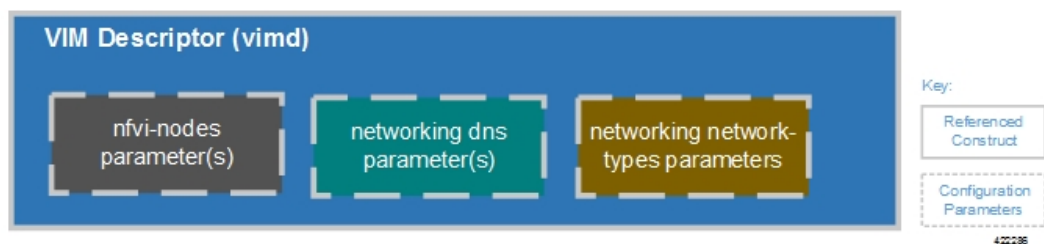


VIM Descriptor (vimd)

The descriptor file that identifies properties for the VIM (Overcloud) nodes. It is a collection of:

- NFVI node parameters
- DNS network parameter(s)
- Network type parameters

Figure 14: vimd Construct



VIM Orchestrator Descriptor (vim-orchd)

The descriptor file that identifies properties for the VIM Orchestrator (Undercloud). It is a collection of:

- Network and login parameters
- Satellite server parameters
- CDN parameters

- Subscription-manager parameters
- External network parameters
- Provisioning network parameters
- Software image and flavor parameters

Figure 15: vim-orchd Construct

