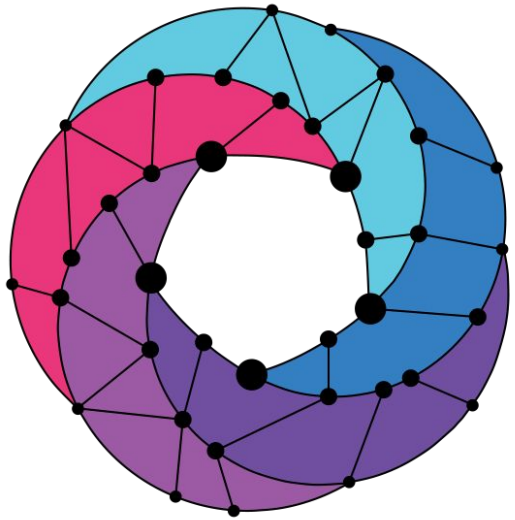


Open Source
MANO

8th OSM Hackfest – Session 2.1
Creating a basic VNF and NS

Abubakr Magzoub (Lancaster University)

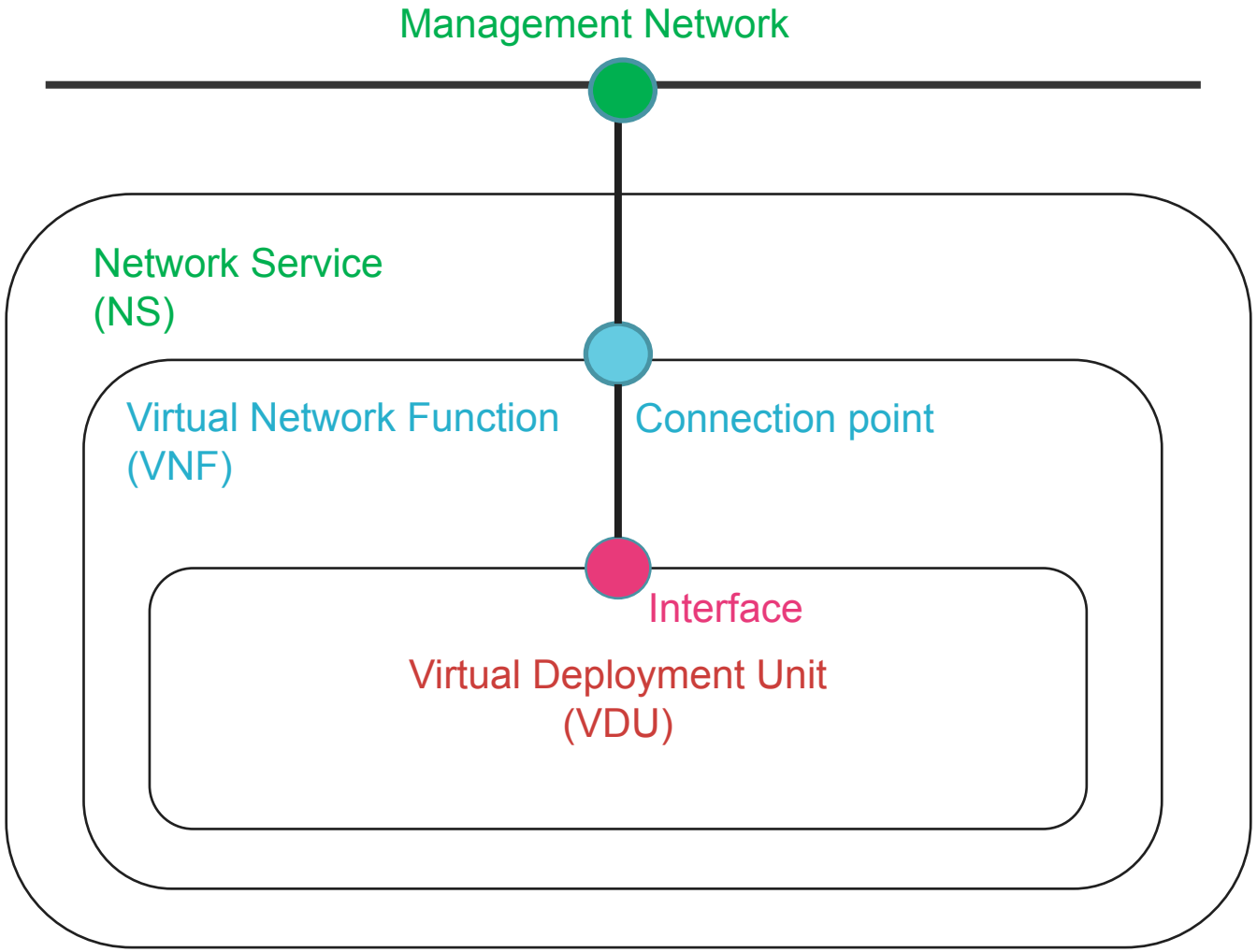
ETSI

Steps for Network Service (NS) Deployment

1. Decide what network function to deploy
2. Define the required specifications
3. Preparing Descriptor:
4. Onboard packages
5. Instantiate the NSD

1. Generate default skeletons
2. Modify default descriptors
3. Validate descriptors
4. Create package for each descriptor

Basic Virtual Network Function Diagram



Open Source MANO

Virtual Network Function Descriptor (VNFD)

Generate VNF Descriptor

- Clone (download) the **devops** repository
  - `git clone -b v6.0 https://osm.etsi.org/gerrit/osm/devops.git`
- Generate skeleton directory for Virtual-Network-Function-Descriptor:
  - `cd /devops/descriptor-packages/tools/`
  - `Ls -l`
  - `./generate_descriptor_pkg.sh -t vnfd --image ubuntu1604 -c hackfest-basic`

Editing VNF Descriptor

- Edit the VNF-descriptor:
  - `ls /devops/descriptor-packages/tools/`
  - `Edit: hackfest-basic_vnfd/hackfest-basic_vnfd.yaml`
- Descriptor language is YAML :
  - Indentation is part of the mark-up
  - Always use the same indentation characters
    - Recommendation: 2 spaces is the preferred indentation
  - Use the Information-Model (IM) tree representation of VNFD as a reference:
    - http://osm-download.etsi.org/ftp/osm-doc/vnfd.html

```
vnfd:
-   id: hackfest-basic_vnfd
    name: hackfest-basic_vnfd
    ...
    mgmt-interface:
        cp: vnf-cp0
    vdu:
    -   id: hackfest-basic_vnfd-VM
        name: hackfest-basic_vnfd-VM
        vm-flavor:
            vcpu-count: 1
            memory-mb: 1024
            storage-gb: 10
        image: ubuntu1604
        interface:
        -   name: eth0
            virtual-interface:
                type: VIRTIO
                ...
            external-connection-point-ref: vnf-cp0
    connection-point:
    -   name: vnf-cp0
        …
```

Information-Model (IM)

Open Source MANO

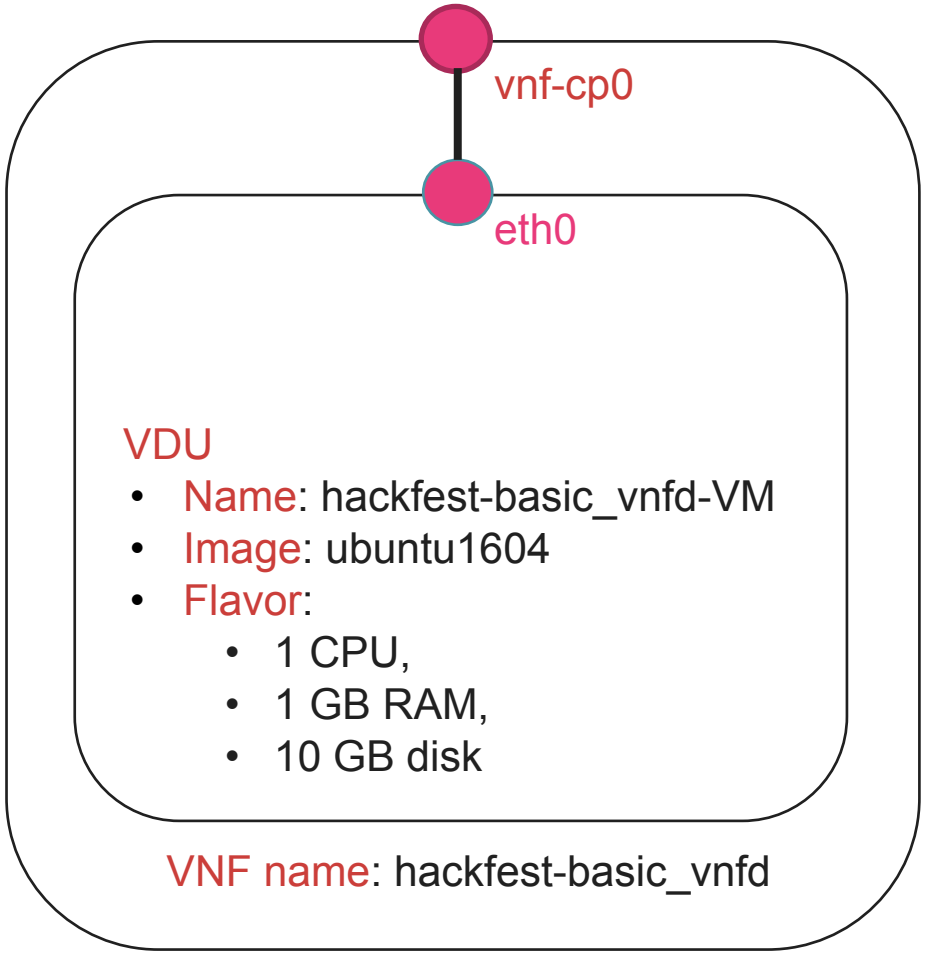
Module: vnfd, Namespace: urn:etsi:osm:yang:vnfd, Prefix: vnfd

Element [+]Expand all [-]Collapse all	Schema Type		Flags	Opts	Status	Path
vnfd	module					
vnfd-catalog	container		config		current	/vnfd:vnfd-catalog
schema-version	leaf	string	config	?	current	/vnfd:vnfd-catalog/vnfd:schema-version
vnfd[id]	list		config		current	/vnfd:vnfd-catalog/vnfd:vnfd
id	leaf	string	config		current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:id
name	leaf	string	config		current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:name
short-name	leaf	string	config	?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:short-name
vendor	leaf	string	config	?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vendor
logo	leaf	string	config	?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:logo
description	leaf	string	config	?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:description
version	leaf	string	config	?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:version
vnf-configuration	container		config		current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vnf-configuration
operational-status	leaf	vnf-operational-status	config	?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:operational-status
mgmt-interface	container		config		current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:mgmt-interface
internal-vld[id]	list		config		current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:internal-vld
ip-profiles[name]	list		config		current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:ip-profiles
connection-point[name]	list		config		current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:connection-point
vdu[id]	list		config		current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vdu
vdu-dependency[vdu-source-ref]	list		config		current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vdu-dependency
service-function-chain	leaf	enumeration	config	?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:service-function-chain
service-function-type	leaf	string	config	?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:service-function-type
http-endpoint[path]	list		config		current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:http-endpoint
scaling-group-descriptor[name]	list		config		current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:scaling-group-descriptor
monitoring-param[id]	list		config		current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:monitoring-param
placement-groups[name]	list		config		current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:placement-groups

http://osm-download.etsi.org/ftp/osm-doc/vnfd.html

VNF Descriptor and Diagram

```
vnfd:
-   id: hackfest-basic_vnfd
    name: hackfest-basic_vnfd
    ...
    mgmt-interface:
        cp: vnf-cp0
    vdu:
    -   id: hackfest-basic_vnfd-VM
        name: hackfest-basic_vnfd-VM
        vm-flavor:
            vcpu-count: 1
            memory-mb: 1024
            storage-gb: 10
        image: ubuntu16.04
        interface:
        -   name: eth0
            virtual-interface:
                type: VIRTIO
                ...
            external-connection-point-ref: vnf-cp0
    connection-point:
    -   name: vnf-cp0
        …
```



Descriptors Validation Prerequisite :

- https://osm.etsi.org/wikipub/index.php/Creating_your_own_VNF_package#Validate_descriptors
- Install the python OSM IM package:
  - `curl "https://osm-download.etsi.org/repository/osm/debian/ReleaseSIX/OSM%20ETSI%20Release%20Key.gpg" | sudo apt-key add -`
  - `sudo apt-get update`
  - `apt-get update && add-apt-repository -y "deb [arch=amd64] https://osm-download.etsi.org/repository/osm/debian/ReleaseSIX stable IM osmclient devops"`
- update python-osm-im and its dependencies:
  - `sudo apt-get update`
  - `sudo apt-get install python-osm-im`
  - `sudo -H pip install pyangbind`

Validating and Generating VNFD Packages:

- Validate VNF Descriptor:
  - `cd /devops/descriptor-packages/tools/`
  - `./validate_descriptor.py hackfest-basic_vnfd/hackfest-basic_vnfd.yaml`
- Generate VNFD Package:
  - `cd /devops/descriptor-packages/tools/`
  - `./generate_descriptor_pkg.sh -t vnfd -N hackfest-basic_vnfd/`

Onboarding VNF Descriptor

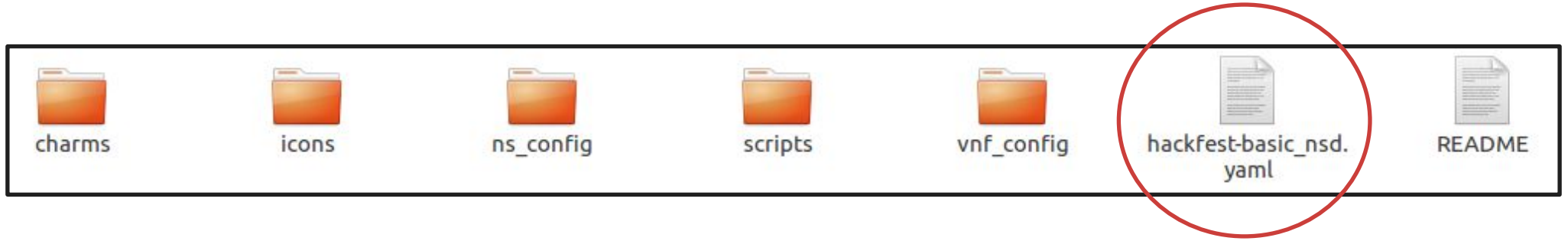
- Onboard VNFD Package using:
  - OSM Web-interface:
    - Log-in to OSM web-interface (admin/admin)
    - Click on “VNF Package”
    - Go the directory where the VNFD is located.
    - Select, drag and drop the VNFD in the “Just drag and drop files here” box
  - OSM Client:
    - `osm vnfd-create hackfest-basic_vnfd.tar.gz`
    - `osm vnfd-list`
    - `osm vnfd-show hackfest-basic_vnfd`

Open Source MANO

Network Service Descriptor (NSD)

Generate NS Descriptor

- Generate skeleton directory for Virtual-Network-Function-Descriptor:
  - `cd /devops/descriptor-packages/tools/`
  - `./generate_descriptor_pkg.sh -t nsd -c hackfest-basic`

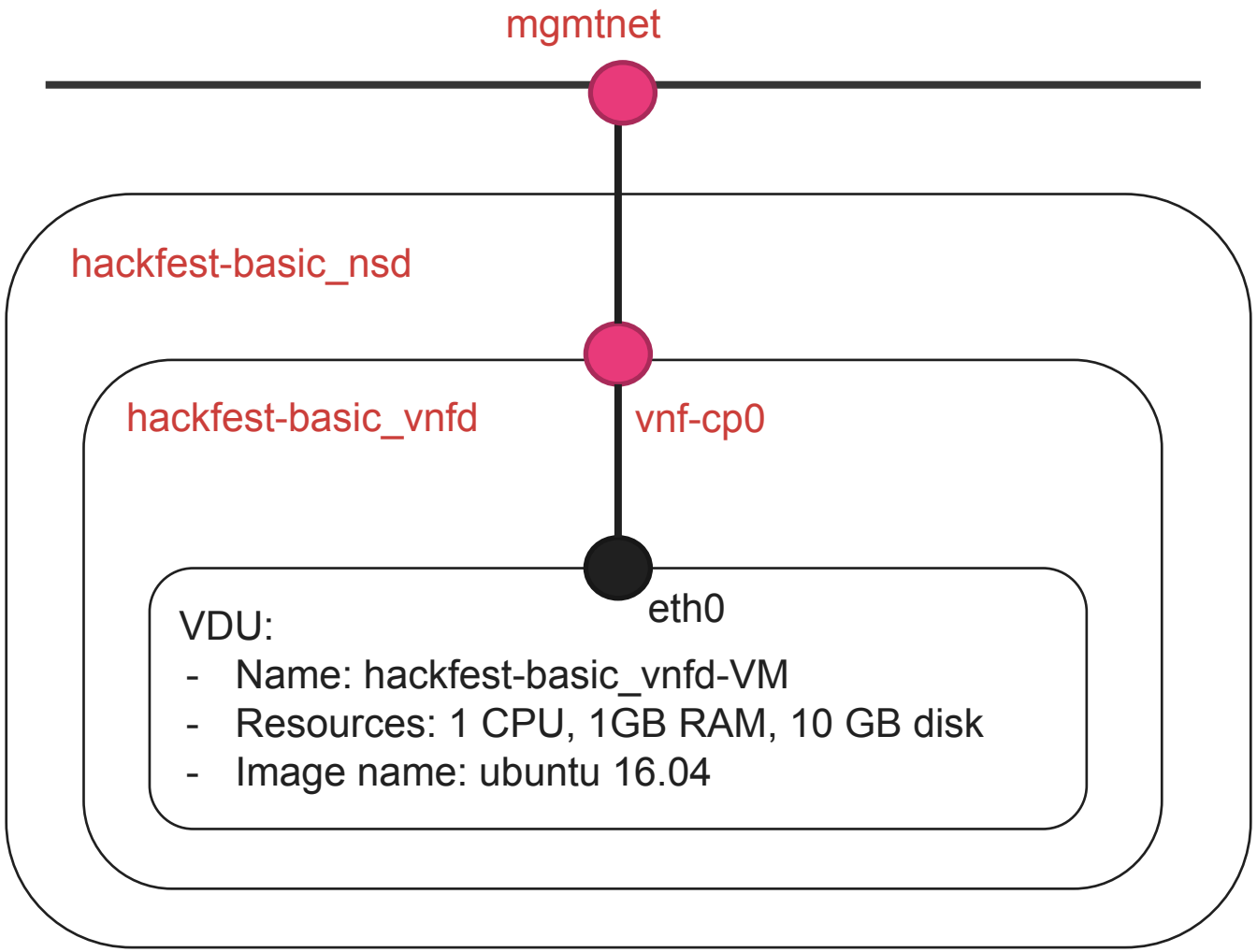


Editing NS Descriptor

- Edit the NS-descriptor:
  - `cd /devops/descriptor-packages/tools/`
  - `Edit: hackfest-basic_nsd/hackfest-basic_nsd.yaml`
- Descriptor language is YAML :
  - Indentation is part of the mark-up
  - Always use the same indentation characters
    - Recommendation: 2 spaces is the preferred indentation
  - Use the Information-Model (IM) tree representation of VNFD as a reference:
    - http://osm-download.etsi.org/ftp/osm-doc/nsd.html

```
nsd:
-   id: hackfest-basic_nsd
    name: hackfest-basic_nsd
    ...
    constituent-vnfd:
    -   member-vnf-index: 1
        vnfd-id-ref: hackfest-basic_vnfd
    vld:
    -   id: mgmtnet
        name: mgmtnet
        type: ELAN
        mgmt-network: true
        vnfd-connection-point-ref:
        -   member-vnf-index-ref: 1
            vnfd-connection-point-ref: vnf-cp0
            vnfd-id-ref: hackfest-basic_vnfd
```

NS Diagram



Validating NS Descriptor and Generating Package

Open Source MANO

- Validate NSD Descriptor:
  - `cd /devops/descriptor-packages/tools/`
  - `./validate_descriptor.py hackfest-basic_nsd/hackfest-basic_nsd.yaml`
- Generate NSD Package:
  - `cd /devops/descriptor-packages/tools/`
  - `./generate_descriptor_pkg.sh -t nsd -N hackfest-basic_nsd/`

Onboarding NS Descriptor

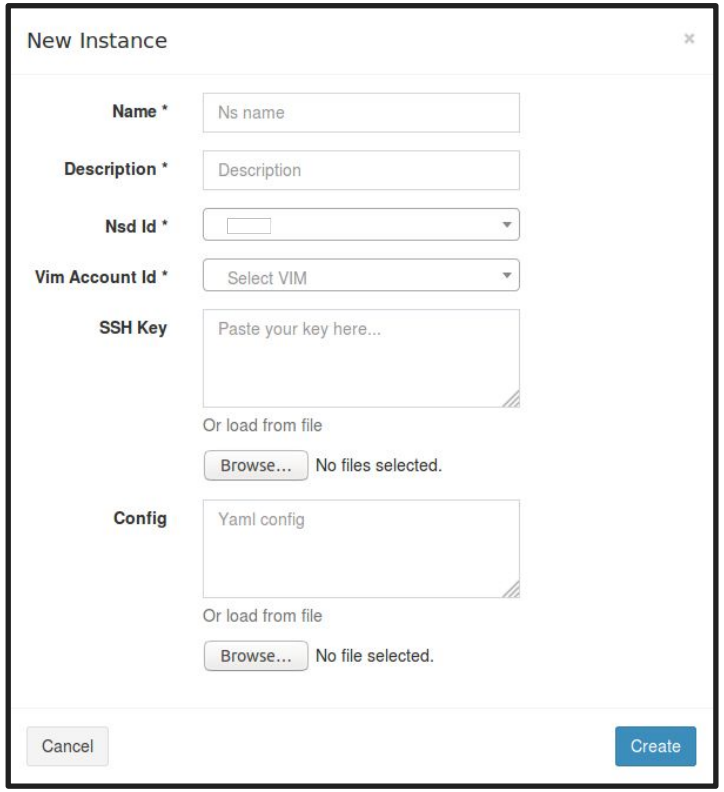
- Onboard VNFD Package using:
  - OSM Web-interface:
    - Log-in to OSM web-interface (admin/admin)
    - Click on “NS Package”
    - Go the directory where the NSD is located.
    - Select, drag and drop the VNFD in the “Just drag and drop files here” box
  - OSM Client:
    - osm nsd-create hackfest-basic_nsd.tar.gz
    - osm nsd-list
    - osm nsd-show hackfest-basic_nsd

Open Source MANO

NS Instantiation

Web-interface NS instantiatation

- Log-in to OSM Web-interface (admin/admin)
- Click on “NS packages”
- Click on “Instantiate NS” icon
- Complete the form:
  - Add a name to the NS instance
  - Select the VIM where the NS will be deployed
  - In the config section, specify a default VIM network name:
    - {vld: [{name: mgmtnet, vim-network-name: osm-ext}] }
  - Paste your SSH key
- Click “Create”
- Click on “VNF Instances”
- Click on the specific VNF to see the instance IP address



Deploying NS with the client

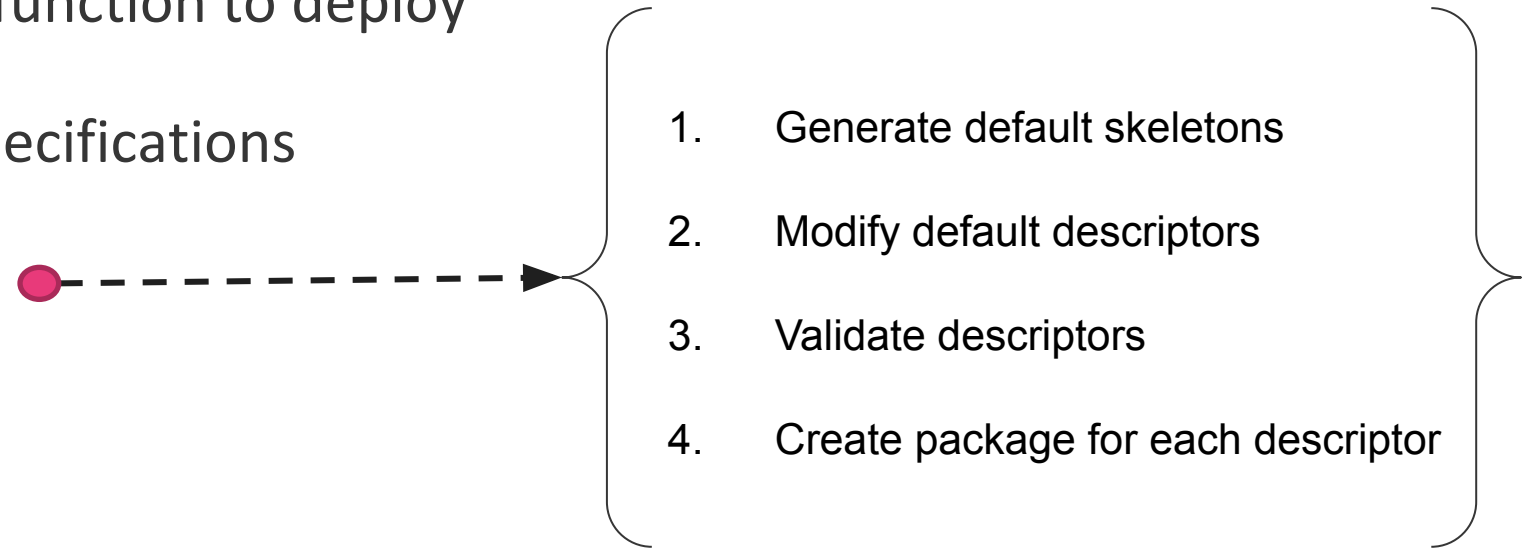
- Create an SSH key
  - `ssh-keygen`
- List NSD Packages:
  - `osm nsd-list`
- Create the NS:
  -
```
osm ns-create \
 --ns_name NS-NAME \
 --nsd_name NSD-PACKAGE-NAME \
 --vim_account VIM-ACCOUNT-NAME  \
 --ssh_keys SSH-PUBLIC-KEY\
 --config '{vld: [ {name: MANAGEMENT-NETWORK-NAME, vim-network-name: DATA-NETWORK} ] }'
```
- For example:
  -
```
osm ns-create \
 --ns_name hf-basic \
 --nsd_name hackfest-basic_nsd \
 --vim_account vim_name  \
 --ssh_keys ~/.ssh/id_rsa.pub \
 --config '{vld: [ {name: mgmtnet, vim-network-name: osm-ext} ] }'
```

Deploying NS with the client

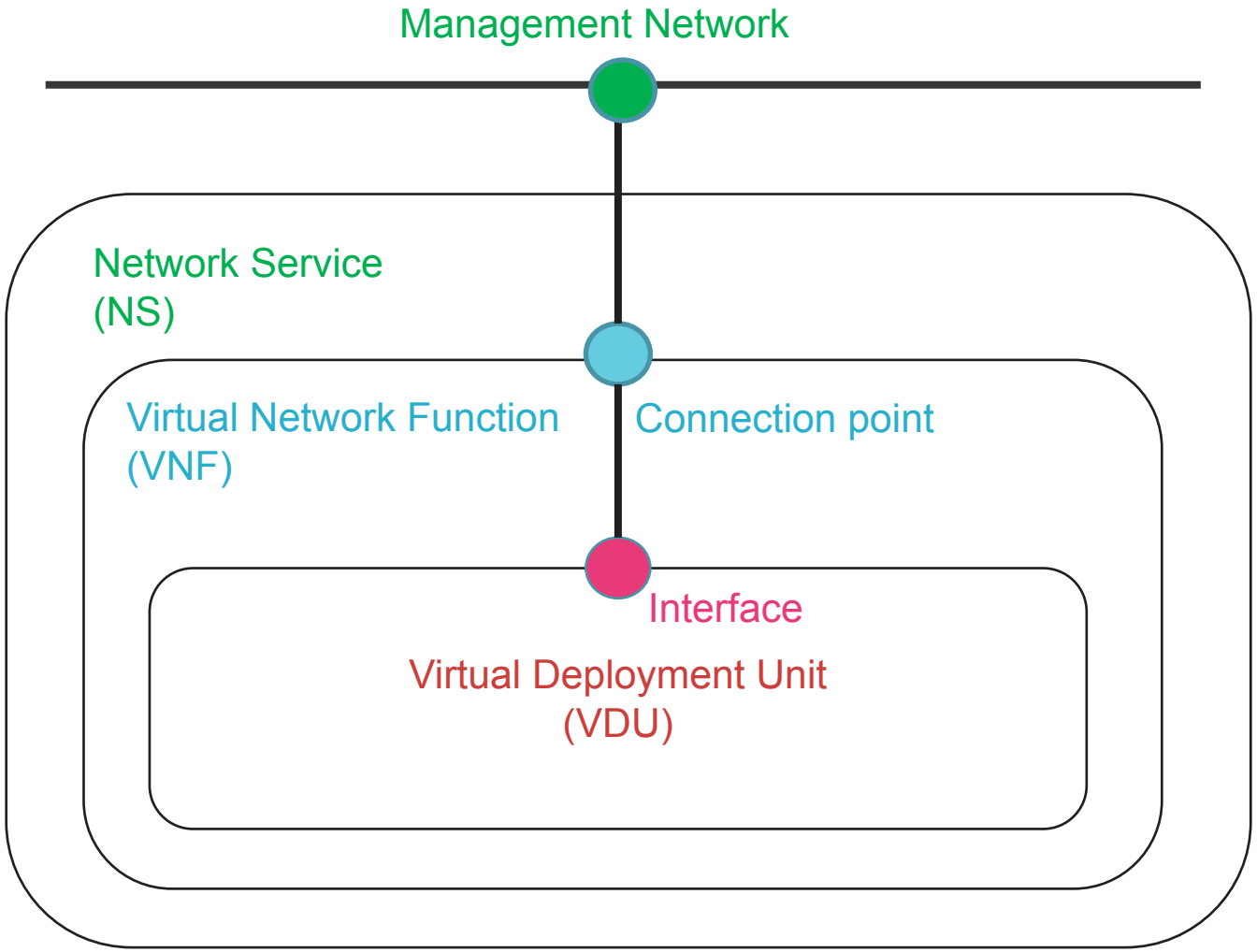
- List instantiated NS
  - `osm ns-list`
- Show instantiated NS details
  - `osm ns-show hf-basic`
- List instantiated VNF instances :
  - `osm vnf-list`
- Show the instantiated VNF instance details:
  - `osm vnf-show VNF-NAM`
- Remote login to VNF instance :
  - `ssh username@instance-ip-address`

Basic VNF and NS Deployment Review

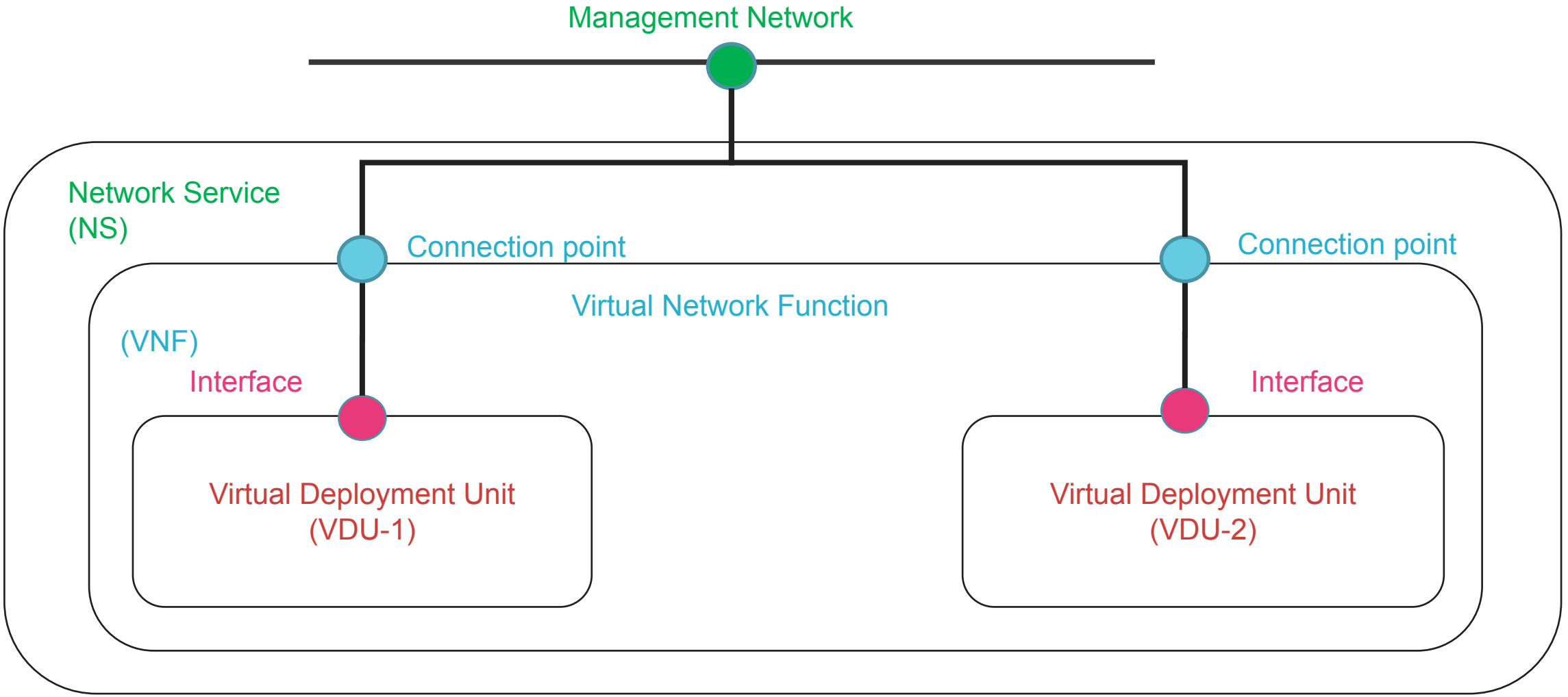
1. Decide what network function to deploy
2. Define the required specifications
3. Preparing Descriptor:
4. Onboard packages
5. Instantiate the NSD

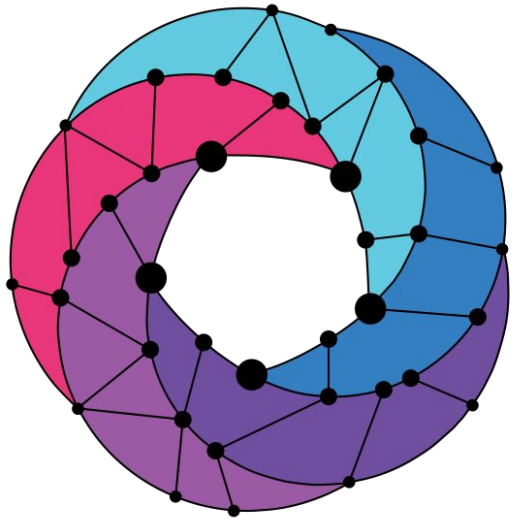


Basic Virtual Network Function Diagram



2 VDUs Virtual Network Function Diagram





Open Source MANO

Find us at:

osm.etsi.org
osm.etsi.org/wikipub