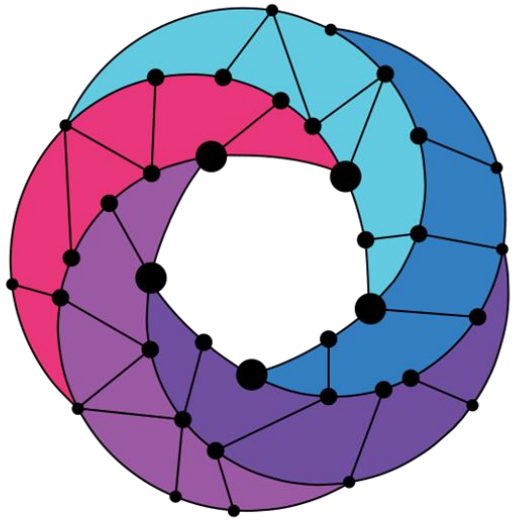


Open Source
MANO

OSM#10 Hackfest
Placement Optimization

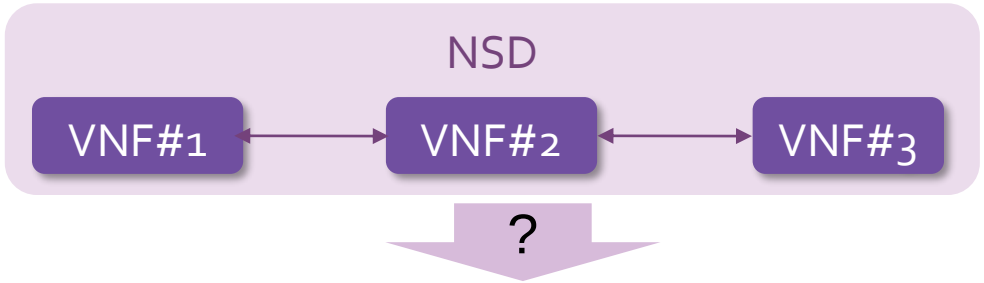
Lars-Göran Magnusson (Arctos Labs)

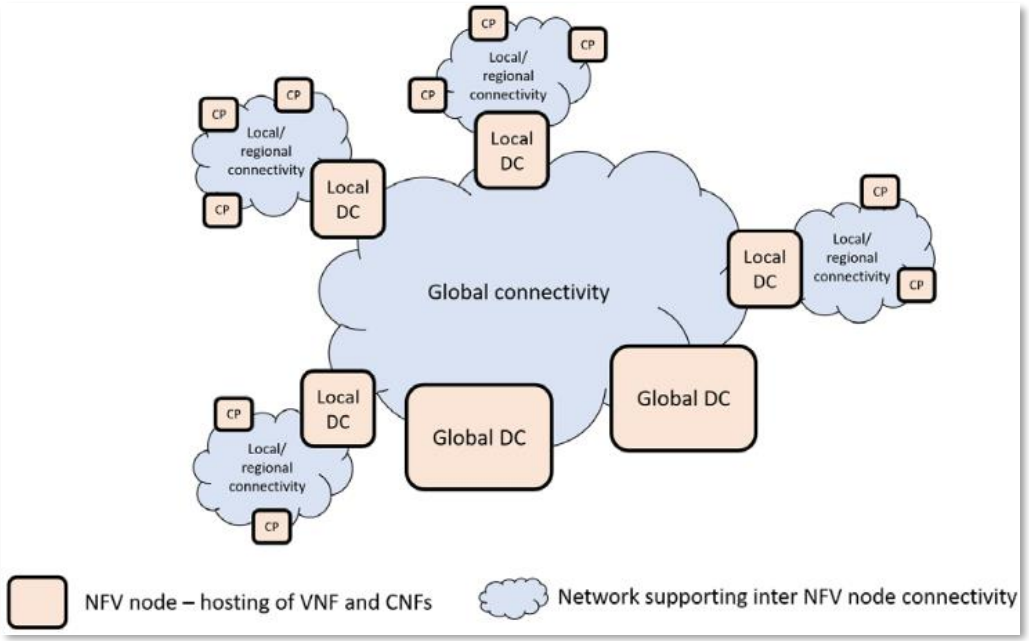
ETSI

Open Source MANO

Introduction to Placement Optimization

The purpose with Placement Optimization

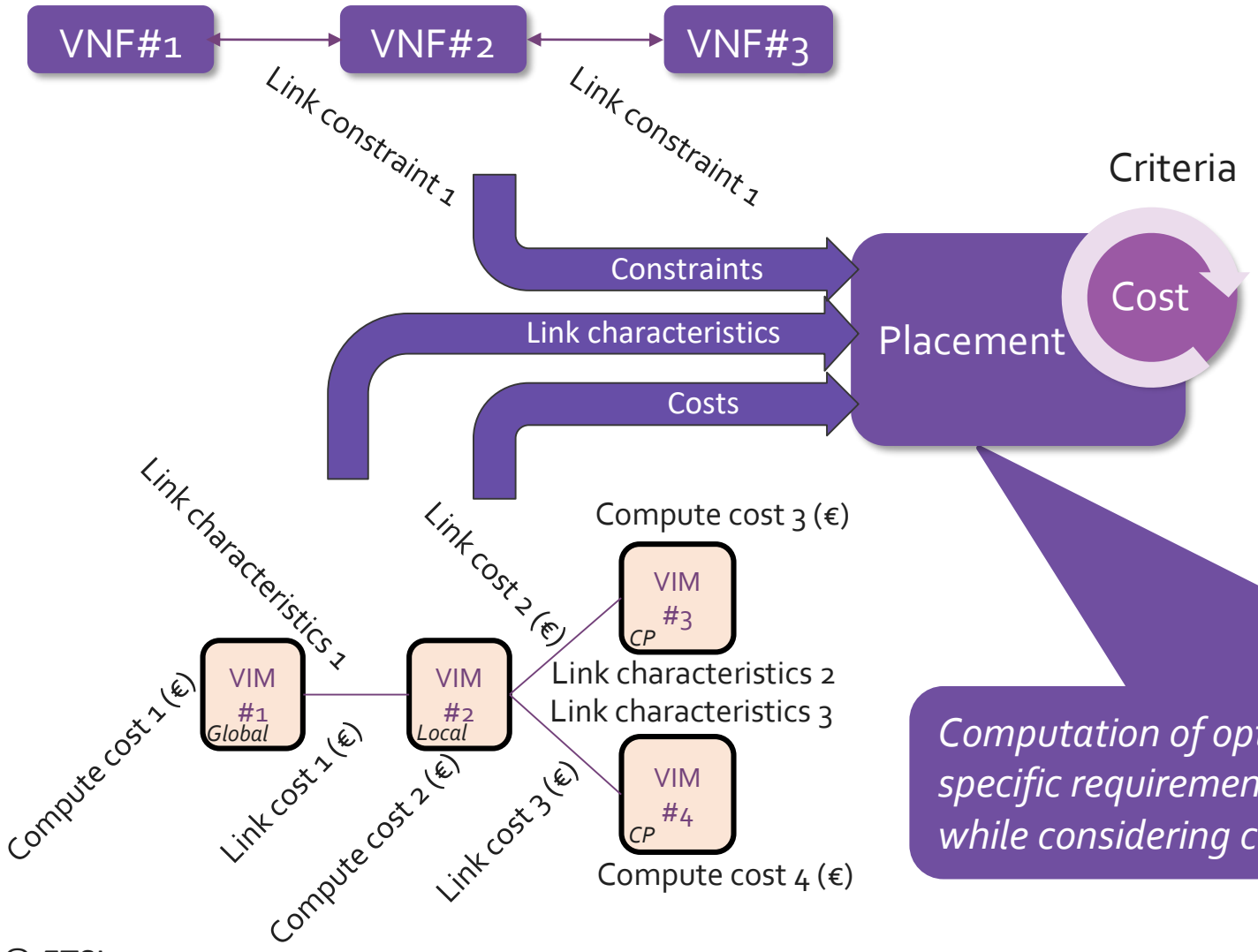




Business Service Basic Architecture,
from *OSM Deployment and Integration WP, Feb 2020*

- There are many VIMs (NFV nodes) that can hosts the NFs
- How to make best use of the available NFV infrastructure for a service instance?
  - Deploy a VNF as close to consumer as it has to be
  - Deploy a VNF as far away that it can be
  - Deploy a VNF to reduce transport load
- Placement in context of OSM is the process of deciding **which VNF goes into which VIM**
- Optimization is subject to:
  - Cost of compute in VIMs
  - Cost of links for NS interworking
  - Constraints in NS interworking (Latency, Jitter) – if there are any
- Placement feature makes this process **Automatic & Optimal**

The optimization process

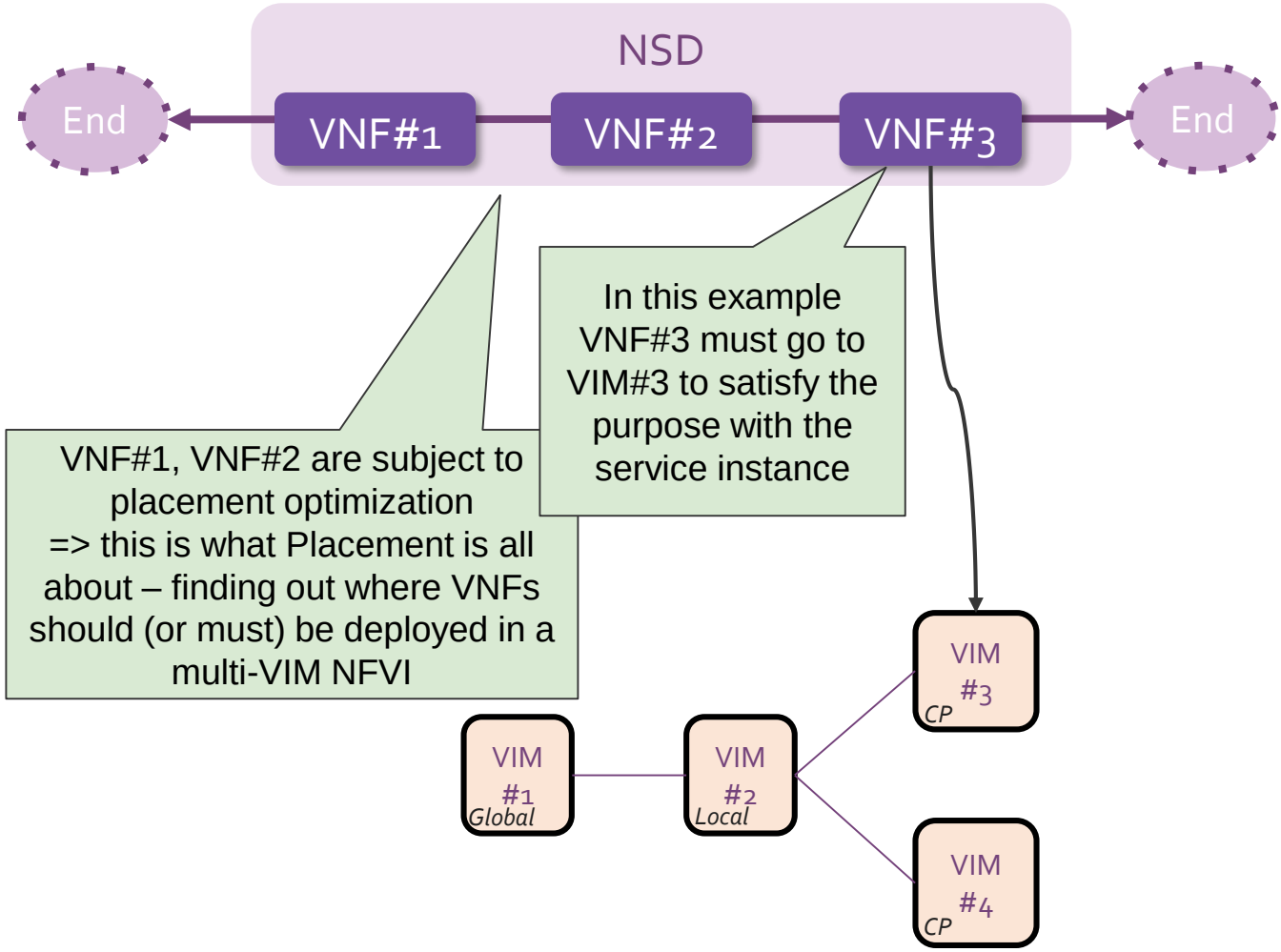


- Placement function
  - Will consider all VIM's available to the user
  - Will make sure constraints are met – if there are any
  - Will optimize Cost (the Criteria)
- I.e. select the distribution of VNFs that fulfills constraints at the lowest possible cost
  - Modeled as a constraints optimization problem

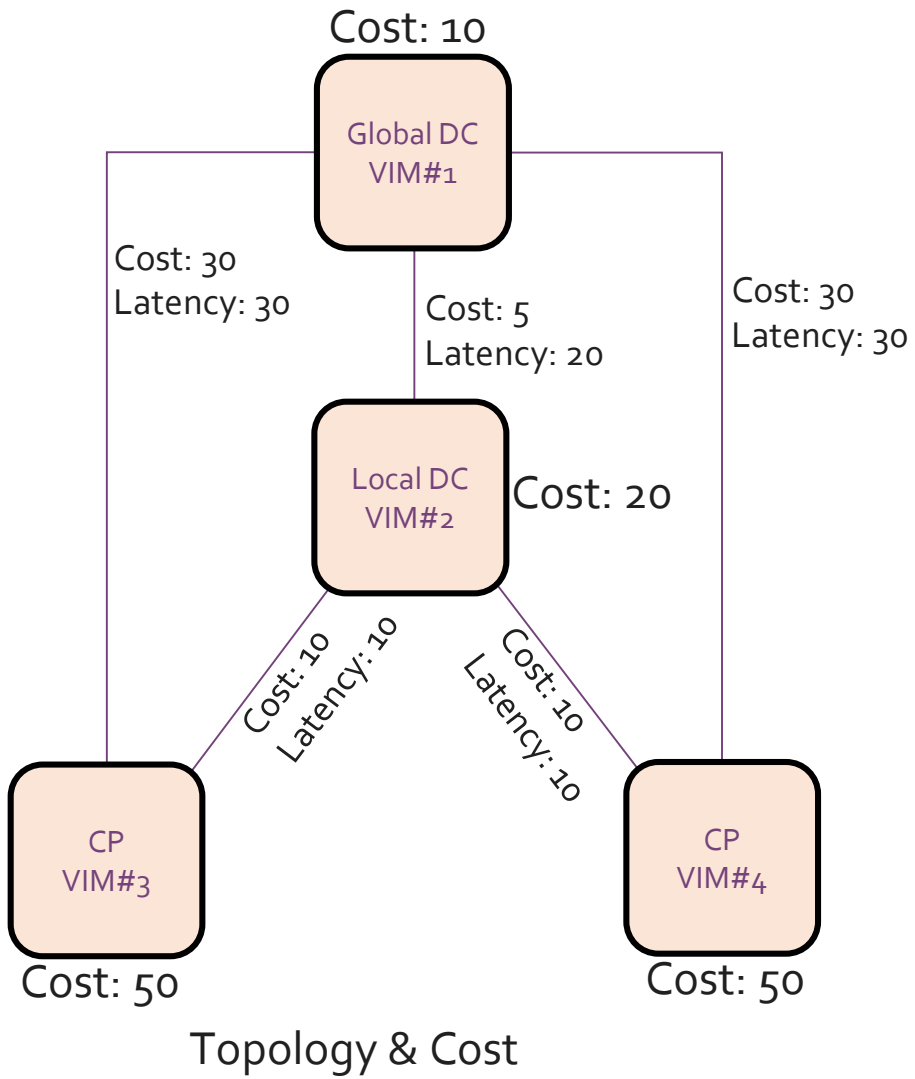
Computation of optimal placement of VNFs over VIMs by matching NS specific requirements to infrastructure availability and run-time metrics, while considering cost of compute/network.

Pin a VNF to a VIM

- Sometimes we have absolute constraints for which VIM a VNF must be hosted on
  - the VIM with a specific VNF (e.g. P-GW)
  - the VIM with connectivity to a PNF
  - a CPE (customer location)
- It is therefore possible to pin the VNFs to a specific VIM

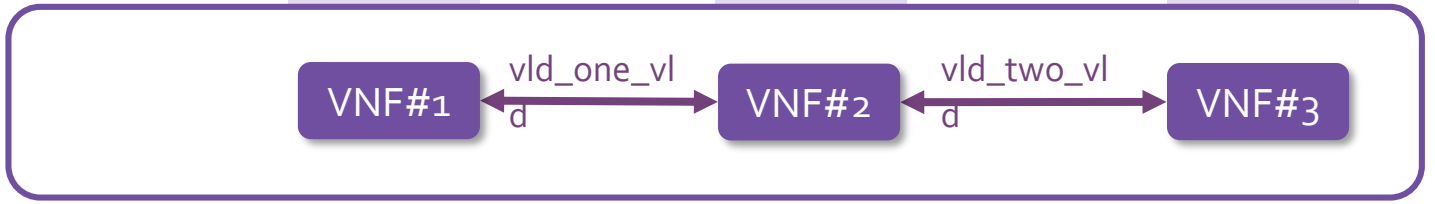


Some different scenarios



Topology & Cost

	Pinning	Constraints	Pinning	Constraints	Pinning
Scenario a:	-		-		-
Scenario b:	-		-		VIM#3
Scenario c:	-	latency=15	-		VIM#4

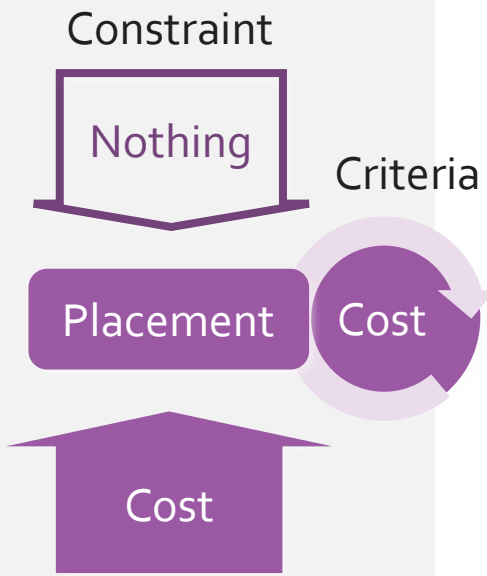


Optimization criteria

Open Source MANO

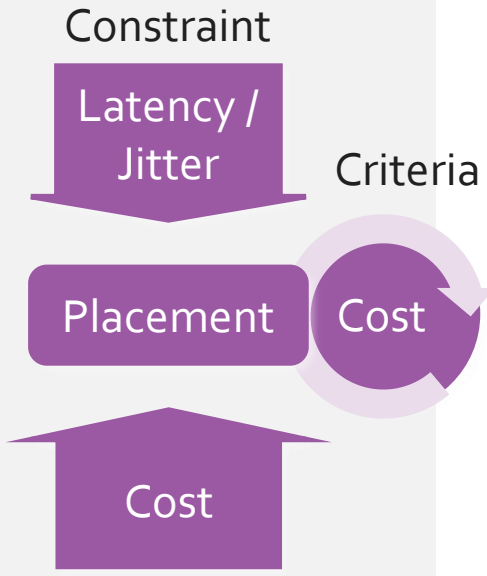
1

Cost optimization **only**



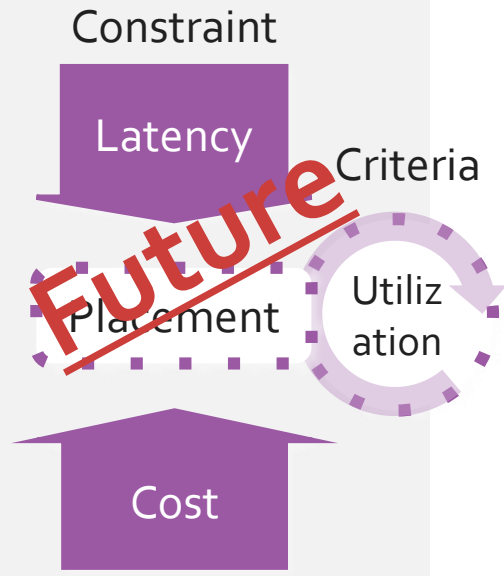
2

Cost optimization **with Latency** constraint



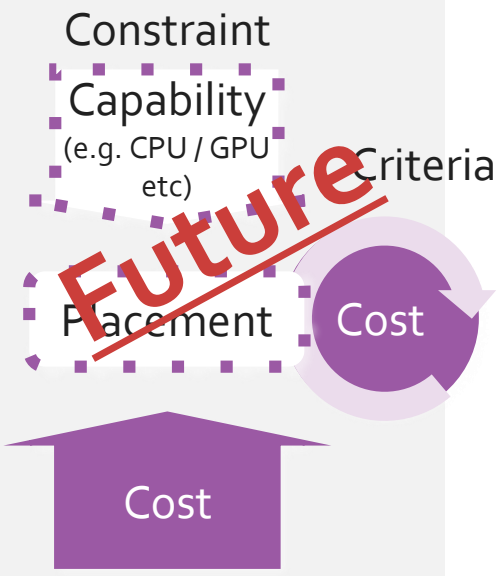
3

Utilization optimization **with Latency** constraint



4

Cost optimization **with Capability** constraint



Open Source MANO

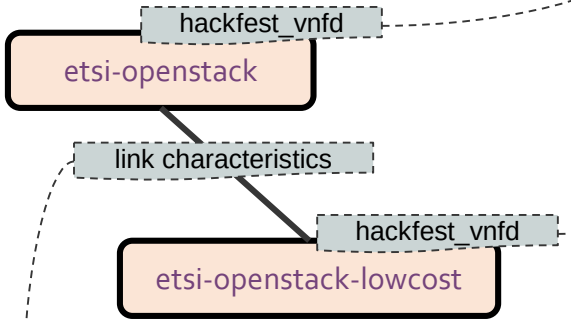
Install, configure and invoke PLA in OSM

Install and Configure PLA

- PLA is optional
  - Install with --pla
  - New in rel EIGHT, basic functionality initially
- You need two configuration files
  - vnf_price_list.yaml
  - pil_price_list.yaml
- The configuration files are copied to the PLA container using the following commands:

```
$ docker cp vnf_price_list.yaml $(docker ps -
qf name=osm_pla):/placement/.

$ docker cp pil_price_list.yaml $(docker ps -
qf name=osm_pla):/placement/.
```




```
- vnfd: hackfest_vnfd
  hackfest:
    prices:
      - vim_url: http://172.21.247.1:5000/v3
        vim_name: etsi-openstack
        price: 5
      - vim_url: http://172.21.7.5:5000/v3
        vim_name: etsi-openstack-lowcost
        price: 1
  admin:
    prices:
      - vim_url: http://172.21.247.1:5000/v3
        vim_name: etsi-openstack
        price: 5
      - vim_url: http://172.21.7.5:5000/v3
        vim_name: etsi-openstack-lowcost
        price: 1
```

```
pil:
  - pil_description: Link between vim1 and vim2
    pil_price: 5
    pil_latency: 10
    pil_jitter: 2
    pil_endpoints:
      - etsi-openstack
      - etsi-openstack-lowcost
```

Note: In current OSM release the link characteristics are hard coded into this file, in future releases this data should be retrieved from the infrastructure by monitoring mechanisms.

Note: Don't copy the files as hackfest participant, it's already done!

Invoke PLA

- Automatic placement is optional, invoked by the user at instantiate of Network Service

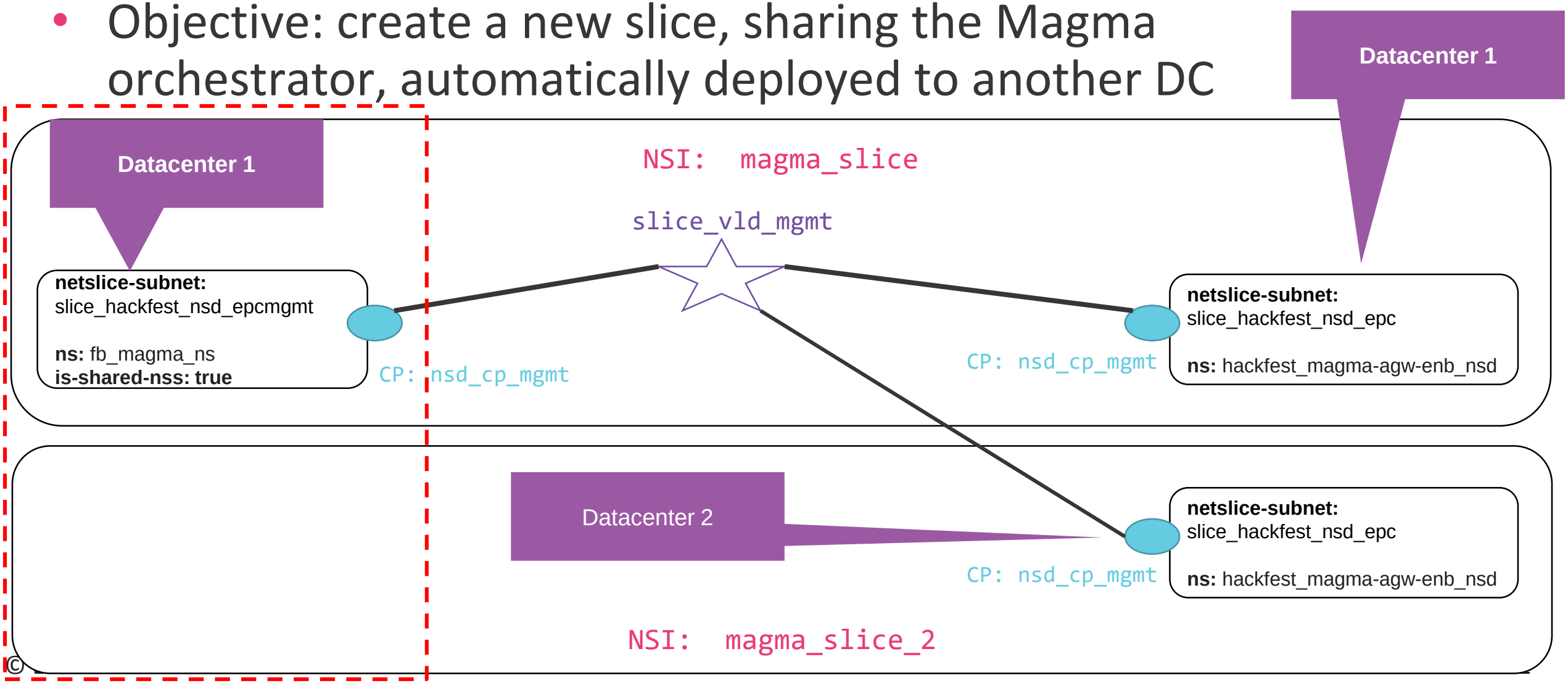
Request Placement Cost Optimization	--config '{ placement-engine: PLA }'
Request Placement Cost Optimization with pinning of specified VNF	--config '{placement-engine: PLA, vnf: [{member-vnf-index: "1", vim_account: OpenStack3}]}’
Request Placement Cost Optimization with VLD Constraints	--config '{placement-engine: PLA, placement-constraints: {vld-constraints: [{id: vld_1, link-constraints: {latency: 120, jitter: 20}}, {id: vld_2, link-constraints: {jitter: 20 }}]}}'
Request Placement Cost Optimization with pinning of specified VNF and with VLD constraints	--config '{placement-engine: PLA, vnf: [{member-vnf-index: "1", vim_account: OpenStack4}], placement-constraints: {vld-constraints: [{id: vld_1, link-constraints: {latency: 15}}]}}'

Open Source MANO

Hands-on: Placement of the Magma AGW + emulator VNF

Shared network slices

- Objective: create a new slice, sharing the Magma orchestrator, automatically deployed to another DC



Launch a 2nd slice

- Create another VIM

The vim name is important, it must match content of the vnf_price_list.yaml file

--user, --password and –tenant follows your personal settings for the hackfest

```
osm vim-create --name osm_hackfest_1_lowcost --user osm_hackfest_x --password osm_hackfest_x --auth_url http://172.21.247.4:5000/v3 --tenant osm_hackfest_x --account_type openstack --config '{management_network_name: osm-ext}'
```

Another VIM url

Don’t forget the additional configuration

- List the vims, and collect the new vim uuid, we need it in the next step

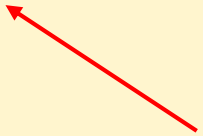
```
osm vim-list
```

Launch a 2nd slice

- Register the PDU to the 2nd vim account

1)Edit pdu.yaml

```
name:          router01
description:    router
type:          gateway
vim_accounts:   [ 94c1218a-e9c7-42d8-b0ae-6de0d0a635ae ]
shared:        false
interfaces:
 -  name:      eth0
    ip-address: 172.21.250.200
    vim-network-name: osm-ext
    mgmt:      true
 -  name:      eth1
    ip-address: 192.168.239.7
    mgmt:      false
```



Enter the uuid for the new vim to the vim_accounts list

2) Launch the pdu-create command

```
osm pdu-create --descriptor_file pdu.yaml
```

- Note: You may also use the GUI (Instances → PDU Instances) to register the PDU

Launch a 2nd slice

- Prepare for PLA support – modify the configuration file
  - make a copy of params_slices.yaml

Uncomment placement-engine: PLA
Uncomment wimAccountId: False

Need another agw_id, agw_name e.g. 101

```
netslice-subnet:
- id: slice_hackfest_nsd_epc
  placement-engine: PLA
  wimAccountId: False
  additionalParamsForVnf:
  - member-vnf-index: '1'
    additionalParams:
      agw_id: 'agw_101'
      agw_name: 'AGW101'
      orch_ip: '172.21.251.XXX'  ## change this to the MetalLB IP address of your orc8r_proxy service.
      orch_net: 'osmnet'

- id: slice_hackfest_nsd_epcmgmt
  additionalParamsForVnf:
  - member-vnf-index: 'orc8r'
    additionalParamsForKdu:
    - kdu_name: orc8r
      additionalParams:
        proxyserviceloadBalancerIP: '172.21.251.XXX' # MetalLB IP Address
```

- Create the slice

```
osm nsi-create --nsi_name magma_slice_2 --nst_name magma_slice_hackfest_nst \
--config_file params_slices2.yaml --ssh_keys ~/.ssh/id_rsa.pub --vim_account hackfest
```

Use your ordinary vim_account

Launch a 2nd slice

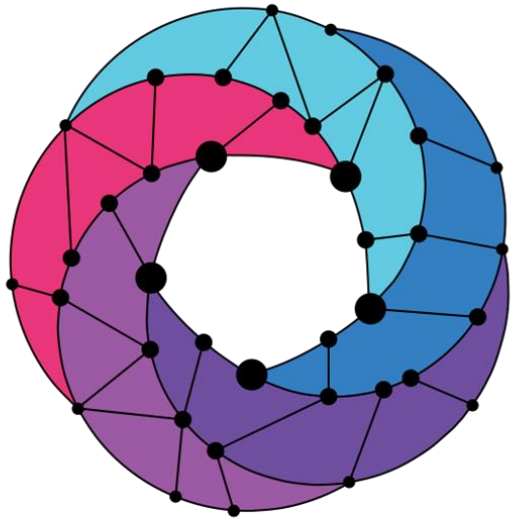
- Check where the vnf ended up

```
osm vnf-list
```

- vim_account_id should correspond to hackfest-lowcost for the new slice
- same Magma orc8r as before
- You may configure and send traffic over the new slice

- Clean up: delete the slice

```
osm nsi-delete <nsi_name> or <nsi_id>
```



Open Source MANO

Find us at:

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