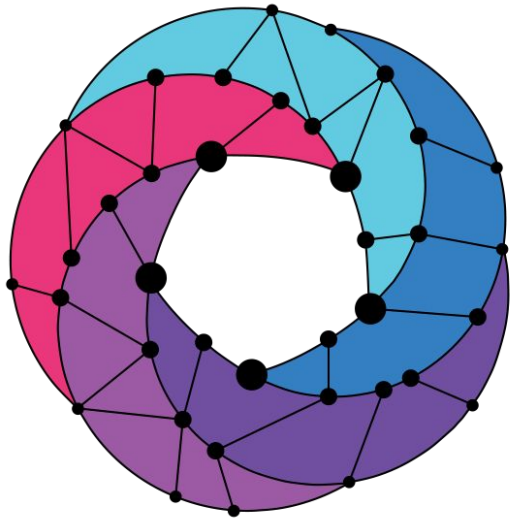


Open Source
MANO

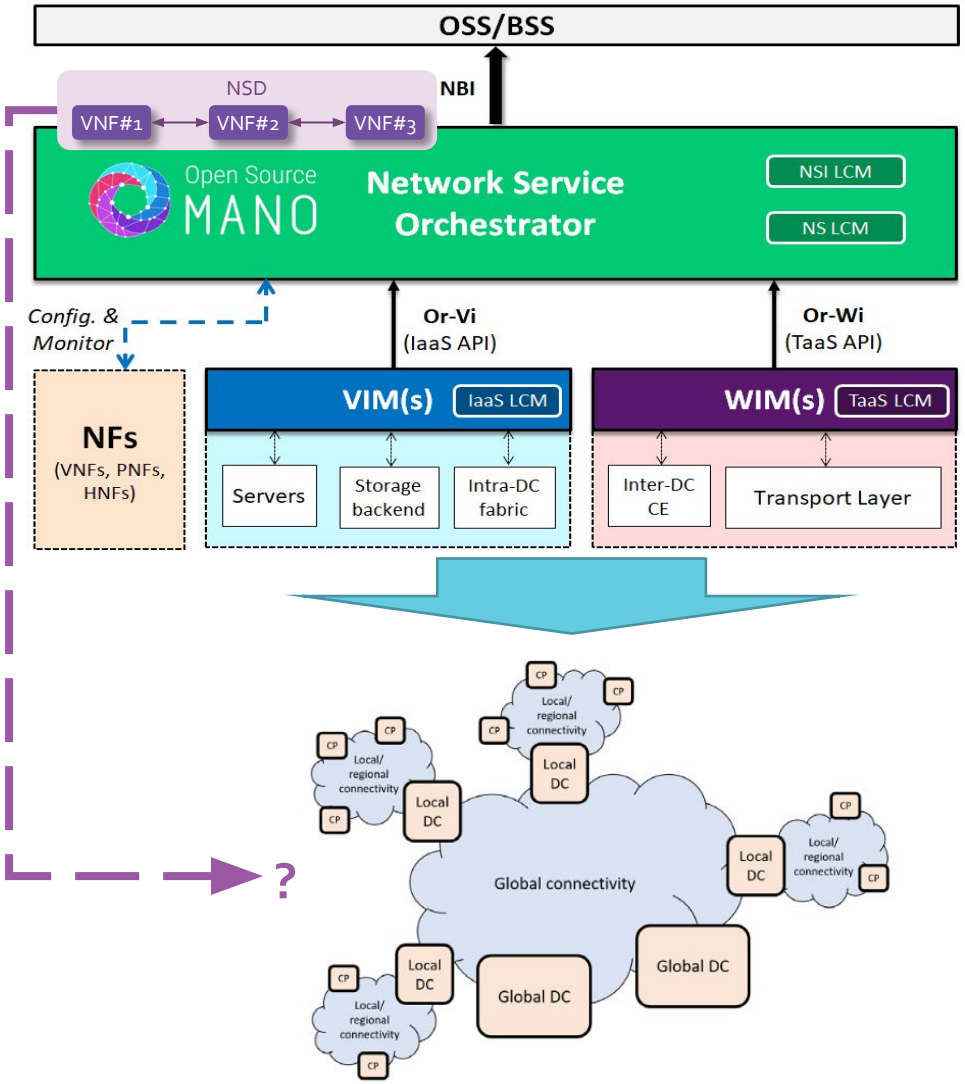
OSM-MR#10 Hackfest

Placement of virtual desktop

Lars-Göran Magnusson (Arctos Labs)

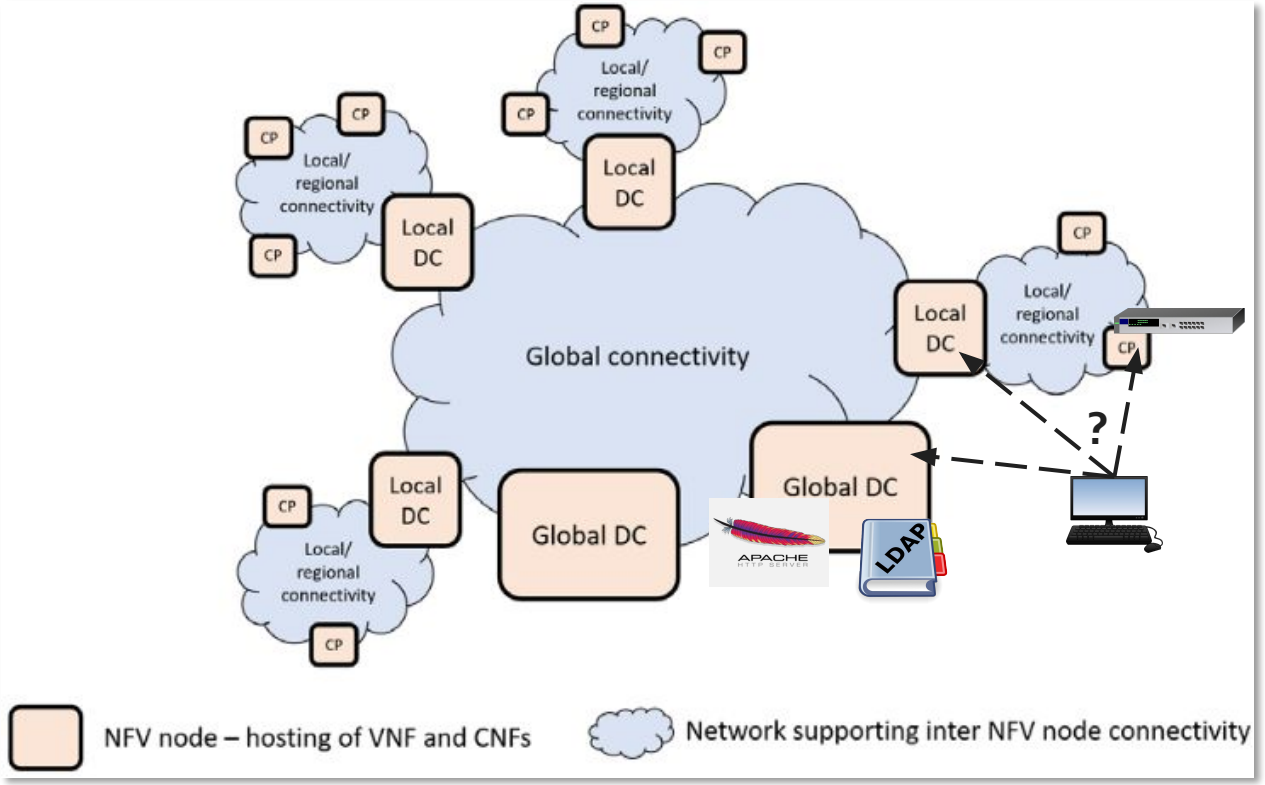
ETSI

In this session



- In which VIM(s) do we deploy the VNF(s)?
- OSM has an optional module, PLA, that support the life cycle manager (LCM) module to make these decisions automatic and optional, in case the NBI client (e.g., OSS) request automatic placement

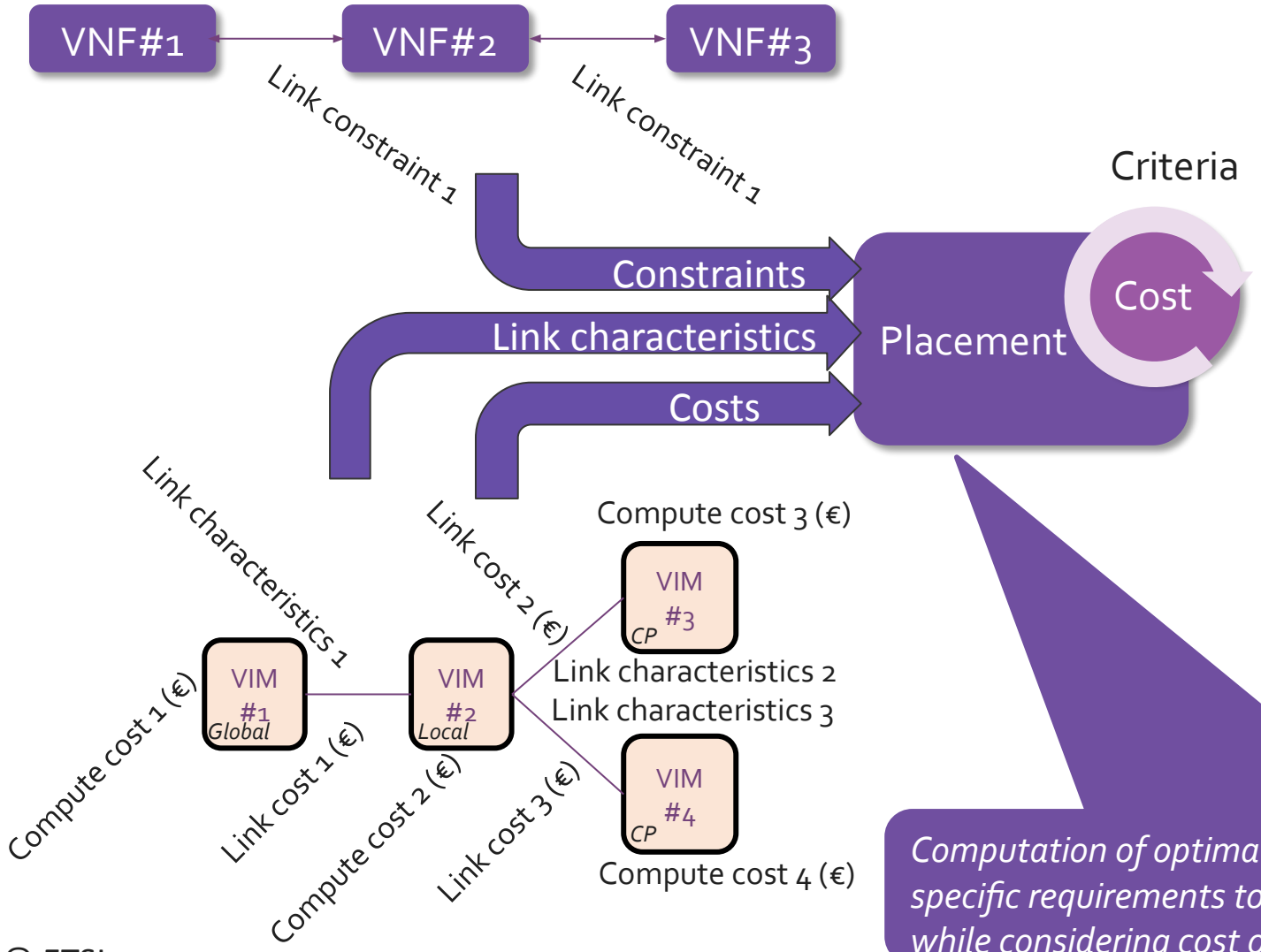
Automating Multi-site deployments



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

- There are many VIMs that can hosts the NFs
- OSM support multi-site deployments - You can map a vnf to a vim
  --config '{vnf: [{member-vnf-index: "1", vim_account: vim1}, {member-vnf-index: "2", vim_account: vim2}]}
- How to make best use of the available NFV infrastructure for a service instance?
  - Deploy a VNF as close to a consumer as it must be
  - Deploy a VNF as far away that it can be
  - Deploy a VNF to reduce transport load
- Placement feature makes this process <u>Automatic & Optional</u>
  with considerations taken to
  - Cost of compute in VIMs
  - Cost of links for NS interworking
  - Constraints in NS interworking (Latency, Jitter) – if there are any

The optimization process



- Placement function
  - Model of the interconnected VIMs
    - connectivity, cost & characteristics
  - Compute cost model
    - cost per vnfd per vim (per tenant)
  - 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 fulfils constraints at the lowest possible cost
  - Modelled as a constraints optimization problem

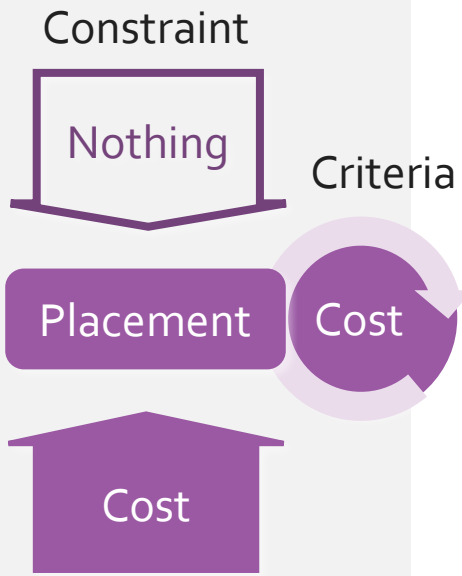
Computation of optimal placement of VNFs over VIMs by matching NS specific requirements to infrastructure availability and run-time metrics, while considering cost of compute/network.

Optimization criteria

Open Source MANO

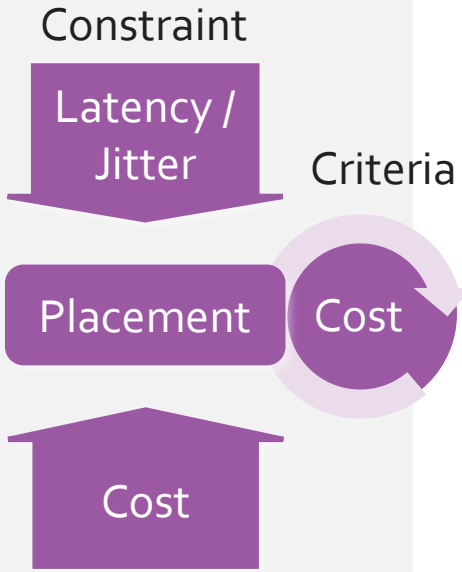
1

Cost optimization **only**



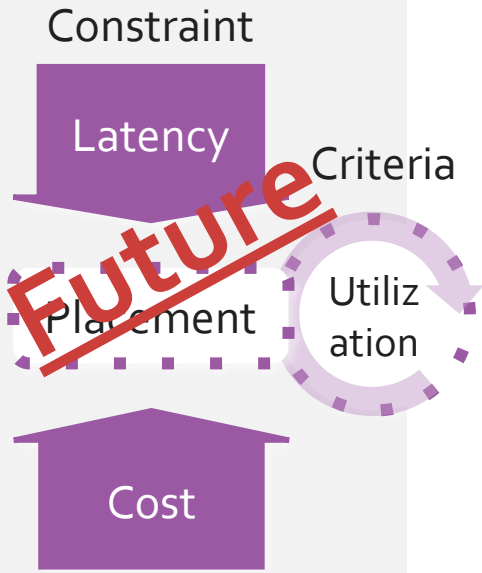
2

Cost optimization **with Latency** constraint



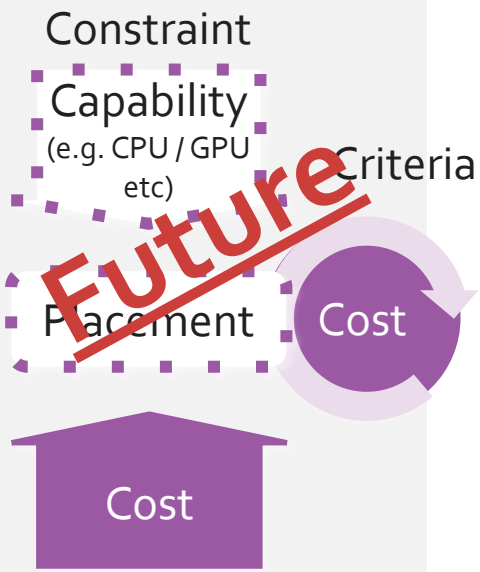
3

Utilization optimization **with Latency** constraint



4

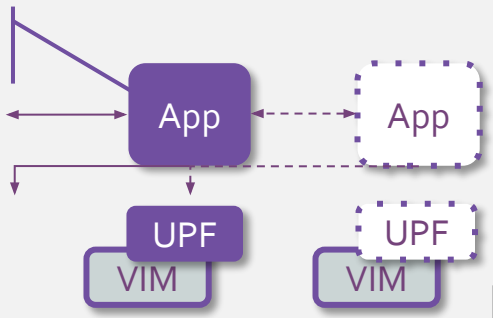
Cost optimization **with Capability** constraint



Examples of use cases

App supporting Low-latency

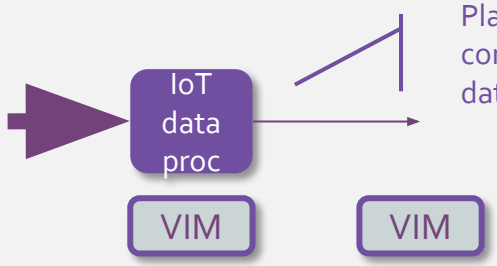
Placement of an App close to customers' UPF to achieve latency constraint



Deploy as close as it must be

Transport optimization (cost) for Application components

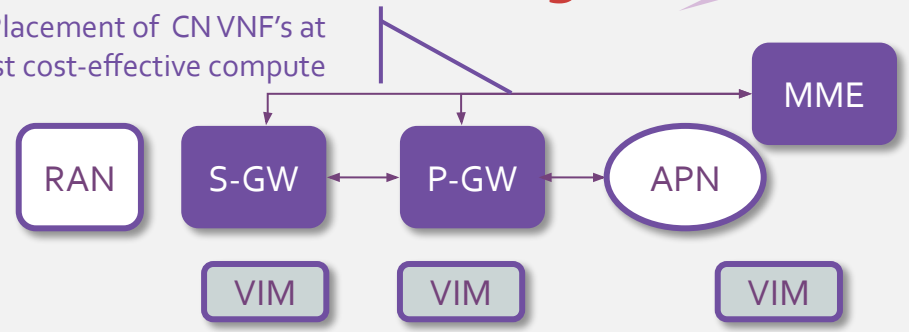
Placement of Application components close to the source of data to reduce transport cost / load

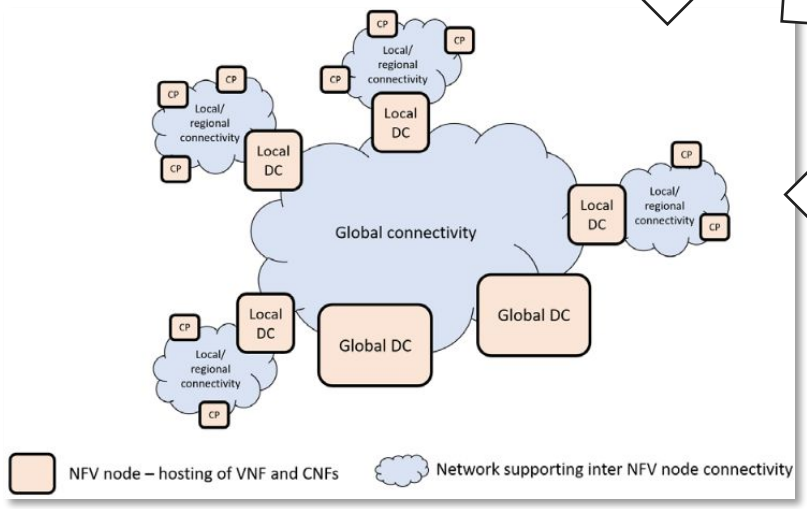


Compute cost optimization for slicing

Deploy as far away as it can be

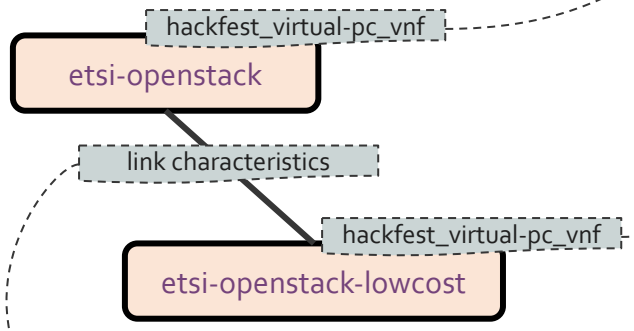
Placement of CN VNF's at most cost-effective compute





Install and Configure PLA

- PLA is optional
  - Install with --pla
  - New in rel EIGHT, basic functionality initially
  - Part next rel NINE point release
- You need two configuration files
  - vnf_price_list.yaml
  - pil_price_list.yaml
- The configuration files are copied to the PLA container



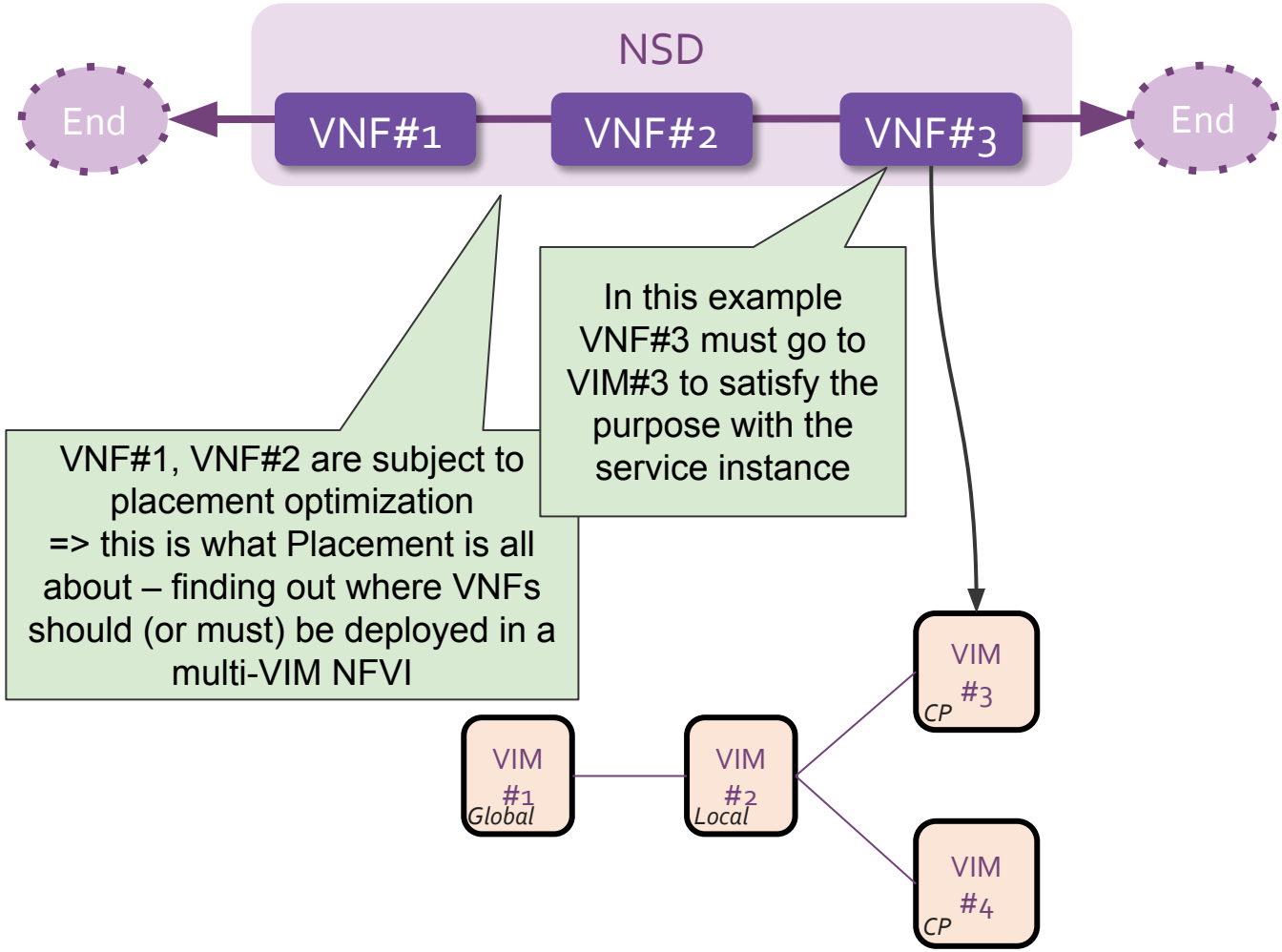

```
- vnfd: hackfest_virtual-pc_vnf
  hackfest:
    prices:
      - vim_url: http://172.21.247.1:5000/v3
        vim_name: etsi-openstack
        price: 5
      - vim_url: http://172.21.7.5:5000/v3
        vim_name: etsi-openstack-lowcost
        price: 1
  admin:
    prices:
      - vim_url: http://172.21.247.1:5000/v3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

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



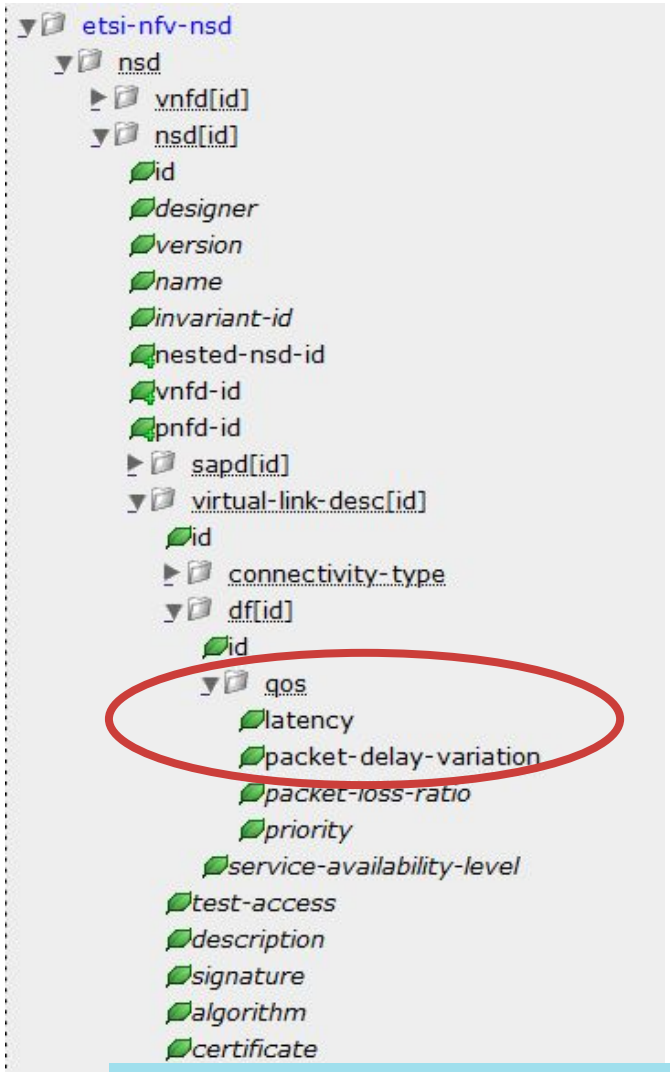
Invoke PLA

- 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}}]}}'

Note: By design in rel NINE, still the old and not the new SOL006 identifiers!

The NSD may define the constraints



- Part of the IM SOL006 adaptation in rel NINE
- Will be considered by PLA if present in the nsd when invoking ns-create with

```
--config '{ placement-engine: PLA }'
```

- Override with

```
--config '{placement-engine: PLA,
placement-constraints: {vld-constraints:
[{id: vld_1, link-constraints: {latency: 120,
jitter: 20}},
{id: vld_2, link-constraints: {jitter: 20
}}]}}'
```

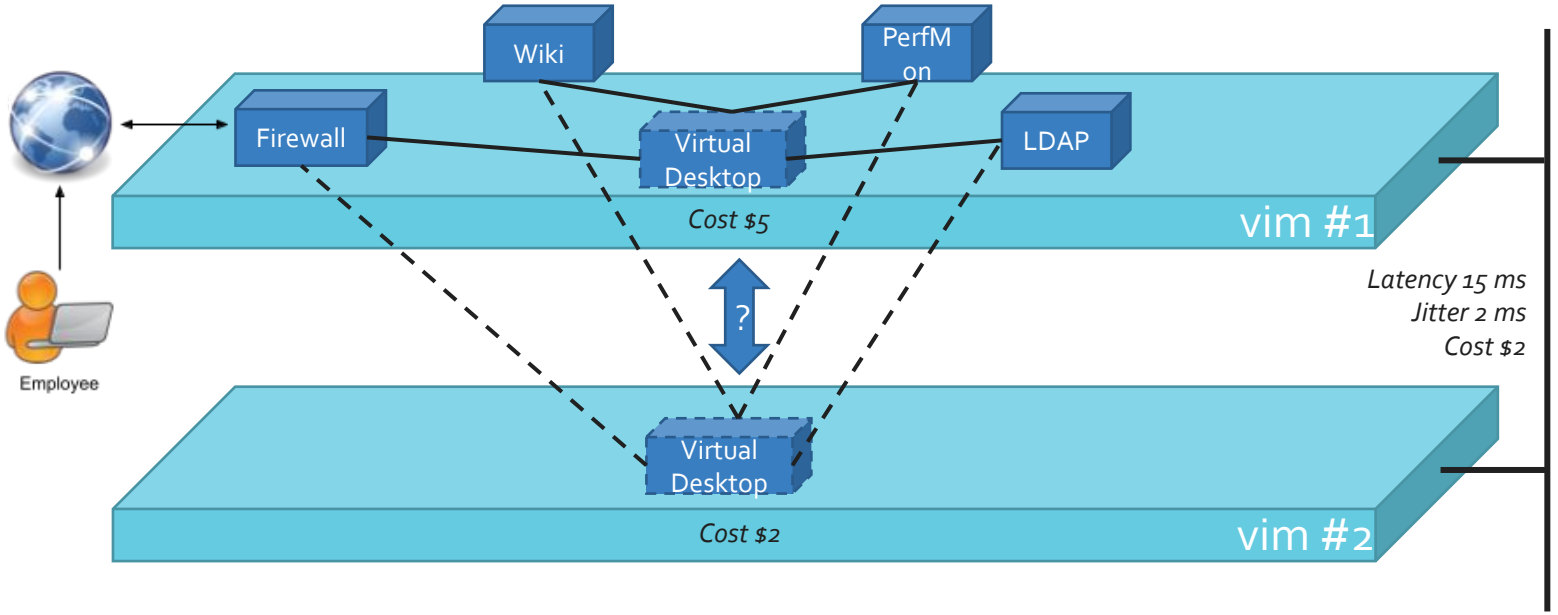
http://osm-download.etsi.org/repository/osm/debian/ReleaseNINE/docs/osm-im/osm_im_trees/etsi-nfv-nsd.html#

Placement of a Virtual Desktop slice

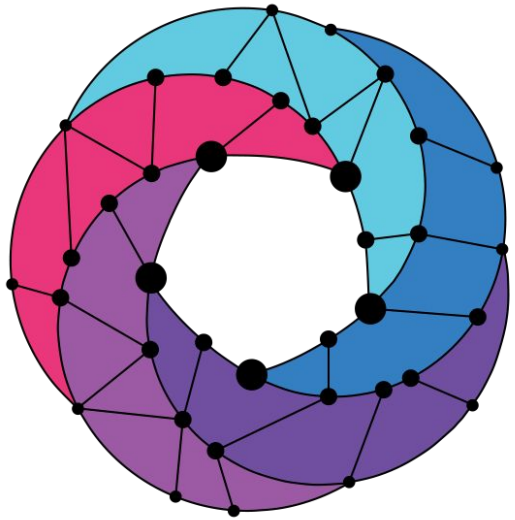
Open Source MANO

Example: Adding a Virtual Desktop slice for a new Employee

- Multi-site setup with two vims, vim#2 is low-cost
- Firewall, Wiki and other components are shared



ns-create command	Virtual desktop placement
--config '{ placement-engine: PLA }'	vim #2, because it gives the lowest total cost (compute + transport)
--config '{placement-engine: PLA, placement-constraints: {vld-constraints: [{id: vld_desktop_firewall, link-constraints: {latency: 10}]}}'	vim #1, even though it is more expensive, because we need to satisfy the latency constraint
--config '{placement-engine: PLA, placement-constraints: {vld-constraints: [{id: vld_desktop_firewall, link-constraints: {latency: 20}]}}'	vim#2, because it gives lowest total cost, and we still satisfy the latency constraint



Open Source MANO

Find us at:
osm.etsi.org
osm.etsi.org/wikipub