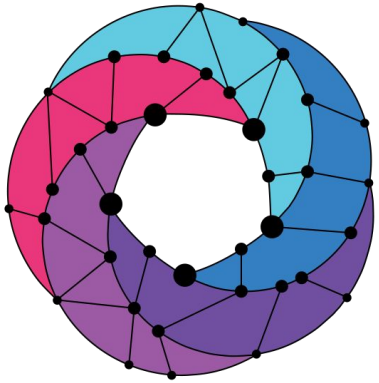


Open Source
MANO

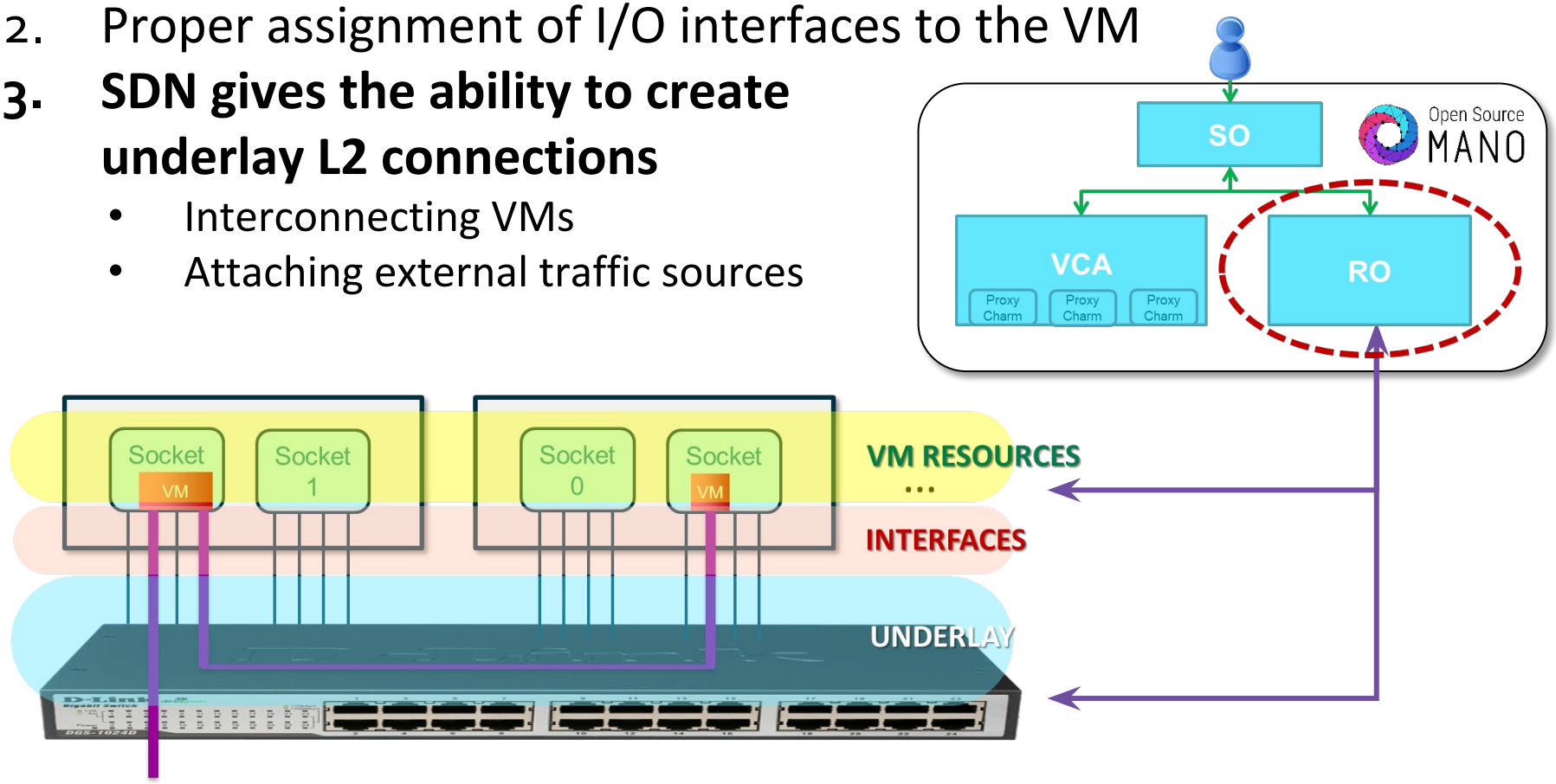
OSM Hackfest – Session 5
Modeling EPA capabilities in VNF

Eduardo Sousa (Canonical)
Guillermo Calviño (Altran)

ETSI

EPA support combined with **SDN Assist** enables chaining of **high performance** VNFs

1. Accurate assignment of resources at VM level
2. Proper assignment of I/O interfaces to the VM
3. **SDN gives the ability to create underlay L2 connections**
   - Interconnecting VMs
   - Attaching external traffic sources



EPA

- **EPA features** like use of large hugepages memory, dedicated CPUs, strict NUMA node placement, and the use of passthrough and SR-IOV interfaces, **can be used in OSM's VNF descriptors since Rel Zero**.

- If your VIM supports EPA, then you don't need to do anything extra to use it from OSM. VIM connectors in OSM take advantage of EPA capabilities if the VIM supports it. All you need to do is build your descriptors and deploy.

- Openstack configuration for EPA (reference guide):
  - https://osm.etsi.org/wikipub/index.php/Openstack_configuration_(Release_THREE)#Configure_Openstack_for_full_EPA_support_in_OSM

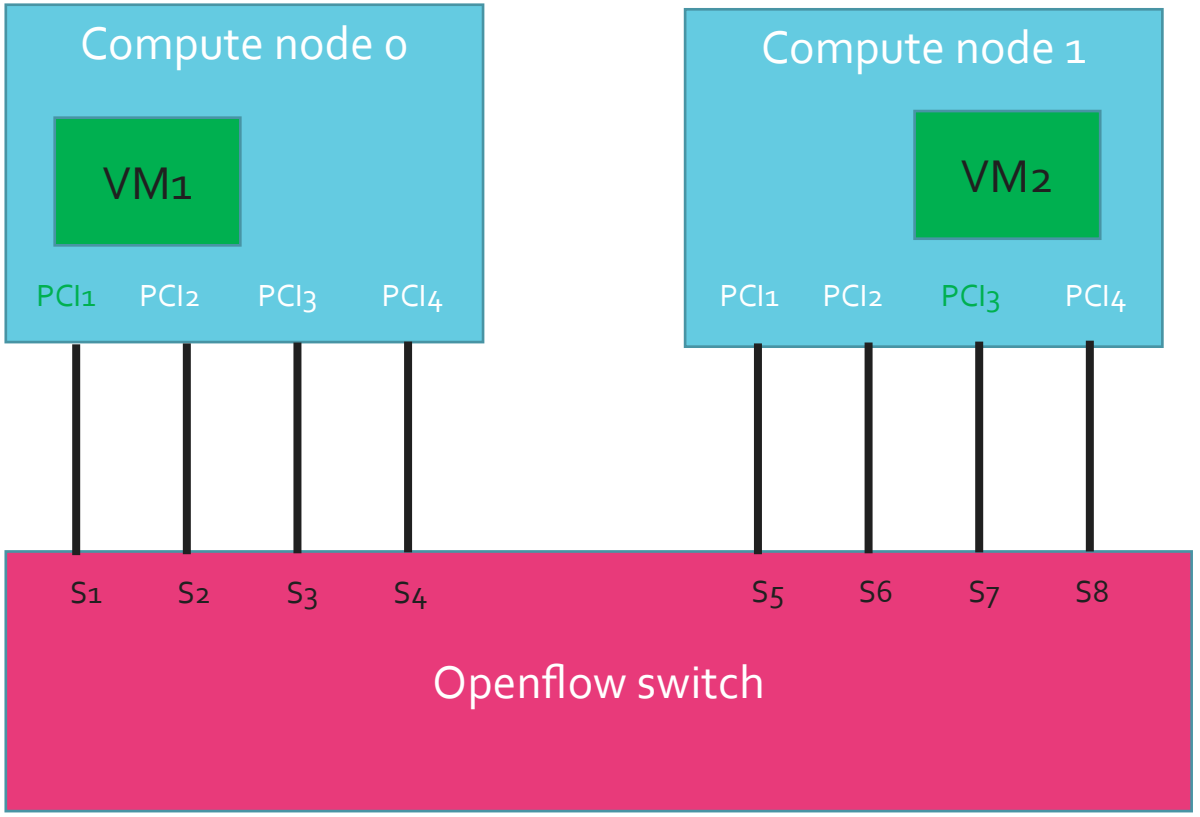
SDN assist

- This feature allows to use an external controller to create the underlying connectivity
- Wiki page:
  - https://osm.etsi.org/wikipub/index.php/EPA_and_SDN_assist
- Requirements:
  - A dataplane switch with Openflow capabilities that will connect the physical interfaces of the VIM compute nodes.
  - An external SDN controller controlling the previous dataplane switch.
  - The mapping between the switch ports (identified by name) and the compute node interfaces (identified by host-id and PCI address)
  - Some VIMs as Openstack requires admin credentials in order to be able to get the physical place of the SRIOV/passthrough VM interfaces

SDN assist. The way it works

- OSM will deploy the VMs of a NS with Passthrough and/or SRIOV interfaces
- OSM will get from the VIM (in your case, Openstack) the compute node where the VM was deployed and the physical interface assigned to the VM (identified by its PCI address).
- OSM will map those interfaces to Openflow ports in the switch making use of the mapping that you should have introduced in the system
- OSM will create the dataplane networks by talking to the SDN controller and connecting the appropriate Openflow ports to the same network.

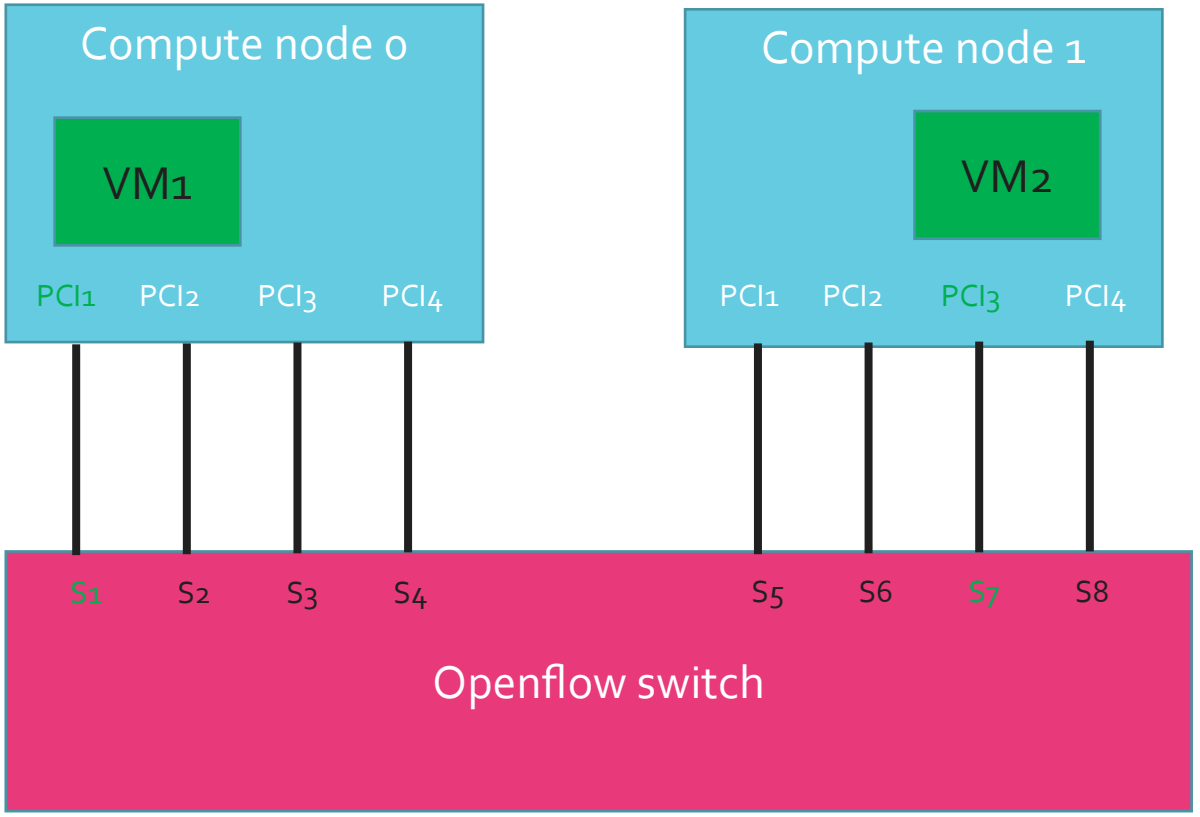
SDN assist. The way it works
1) VMs are deployed



VM1 will be e.g. deployed in compute node 0 and will consume interface identified by PCI address PCI1 and MAC MACX

VM2 will be e.g. deployed in compute node 1 and will consume interface identified by PCI address PCI3 and MAC MACY

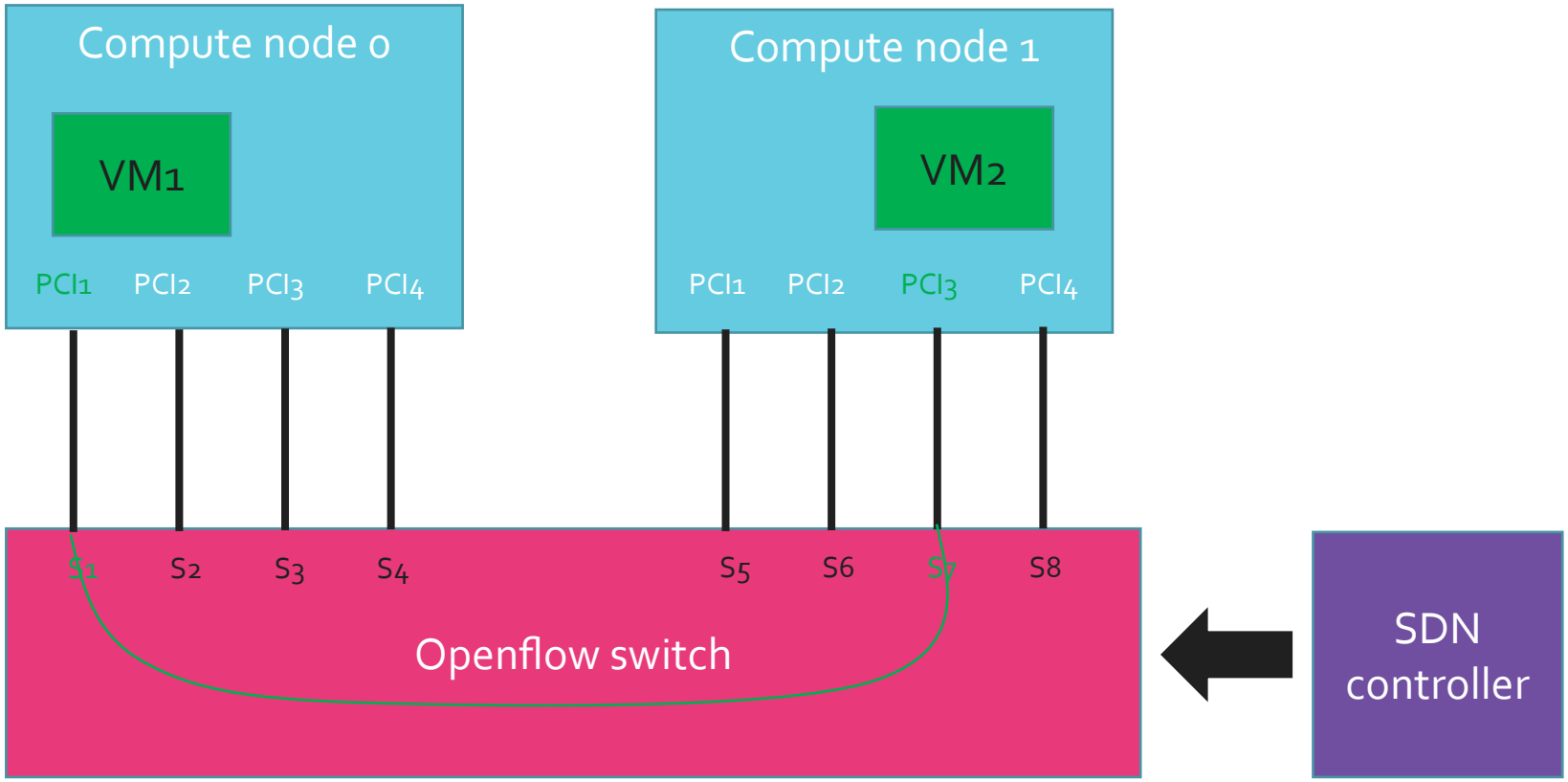
SDN assist. The way it works
2) OSM uses port mapping to identify the ports in the switch



The interface at PCI address PCI1 of compute node 0 is connected to port S1 in the switch

The interface at PCI address PCI3 of compute node 1 is connected to port S7 in the switch

SDN assist. The way it works
3) OSM talks to the SDN controller and connect the ports to the same network.



Example of rules for E-LINE and PCI passthrough interfaces:
IN: Port S1 → OUT: Port S7
IN: Port S7 → OUT: Port S1

Device role tagging

- Whenever a new NS is deployed, and if the VIM allows it, VM interfaces will be tagged with the interface name specified in the descriptor

- This allows proper identification of interfaces in the VM

- A service and a script have to be pre-installed in the VM:
  - RedHat-based VMs: https://github.com/oglok/udev_data
  - Ubuntu-based VMs: https://github.com/gcalvino/udev_data

Adding new VIM account: openstack-epa

- VIM:
  - openstack-epa: 172.21.2.22
- Test VIM:
  - ping
  - curl http://:5000/v2.0
  - Load Openstack credentials:
    - export OS_AUTH_URL=http://172.21.2.22:5000/v2.0
    - export OS_USERNAME=osm
    - export OS_TENANT_NAME=osm
    - export OS_PASSWORD=osm
  - Run some commands:
    - openstack image list
    - openstack network list
    - openstack flavor list
    - openstack server list

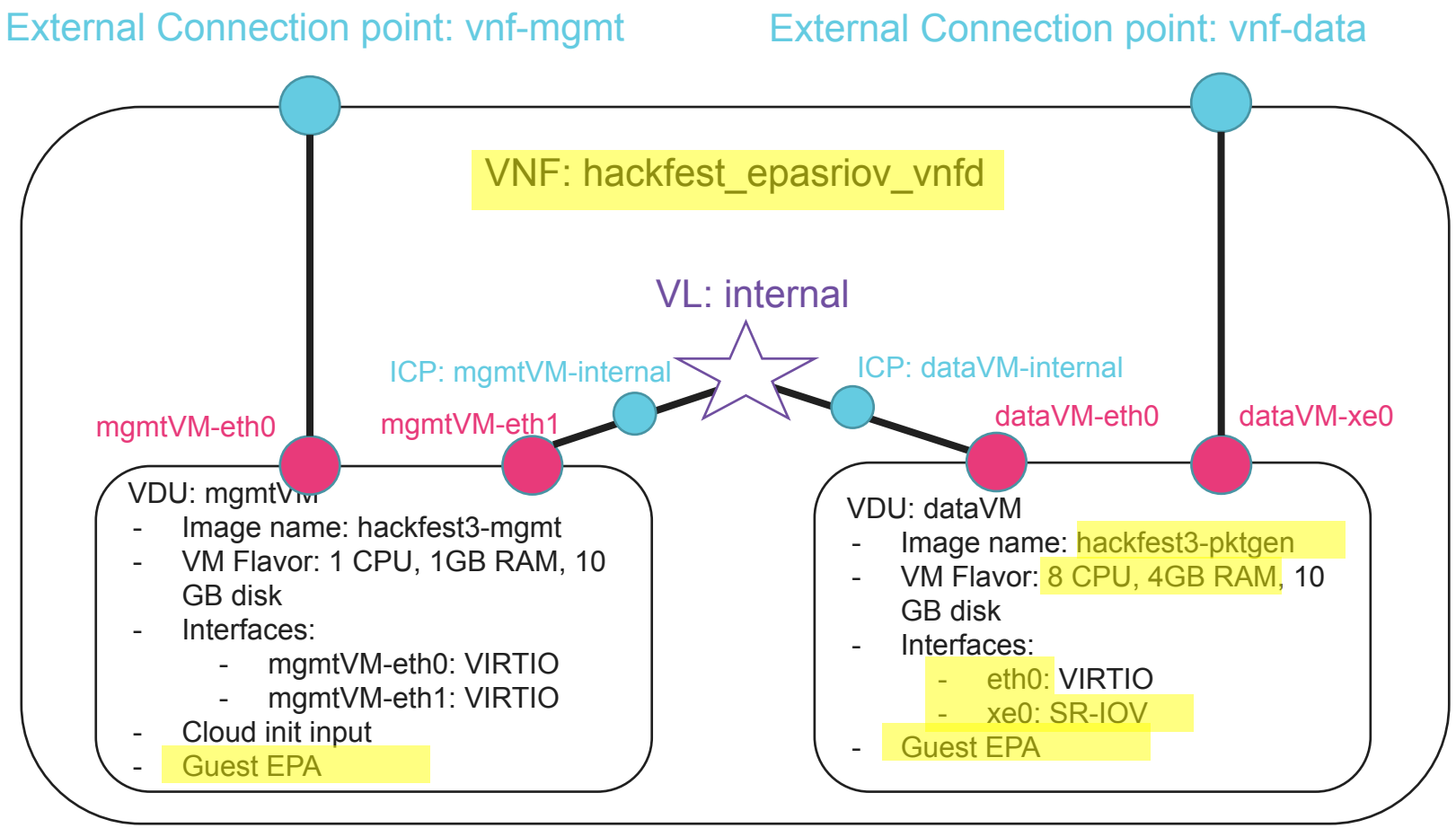
Adding new VIM account: openstack-epa

- Add your second VIM ‘openstack-epa’ with the OSM client:
  - osm vim-create --name openstack-epa --account_type openstack \ --auth_url http://172.21.2.22:5000/v2.0 \ --user xxx --password xxx --tenant xxx \ --description "ETSI openstack site 2, with EPA, with tenant xxx“ \ --config '{dataplane_physical_net: physnet_sriov, microversion: 2.32}'
  - osm vim-list
  - osm vim-show openstack-epa
- Config options:
  - dataplane_physical_net:
    - Used to instantiate VMs with SR-IOV and Passthrough interfaces
    - Value: The physical network label used in Openstack both to identify SRIOV and passthrough interfaces (nova configuration) and also to specify the VLAN ranges used by SR-IOV interfaces (neutron configuration).
  - microversion:
    - Used for device role tagging
    - Value: 2.32

Adding an SDN controller

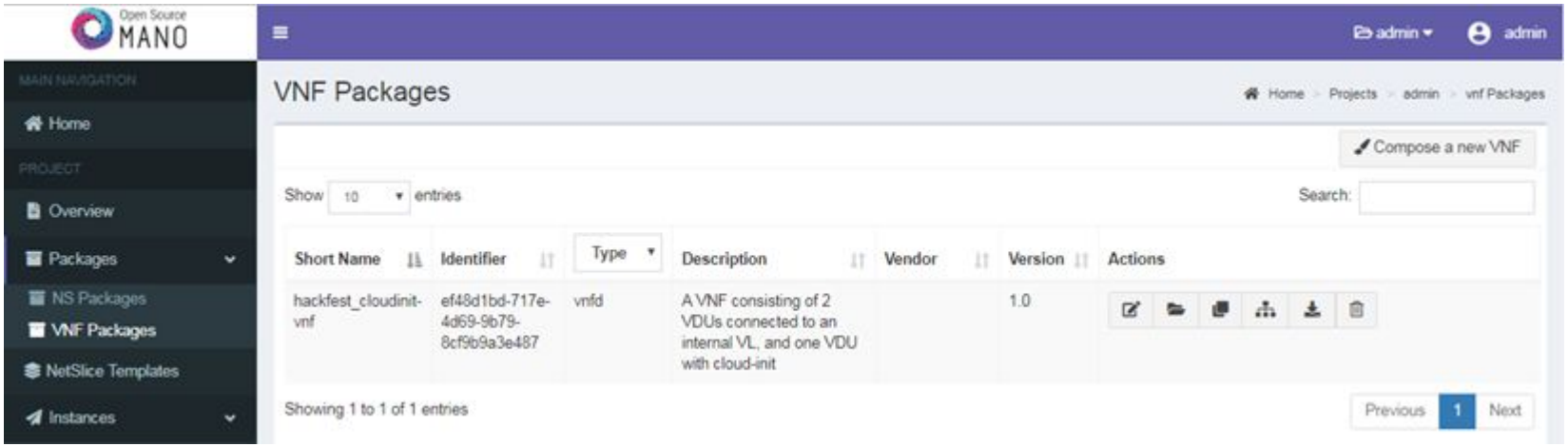
- Add the SDN controller through OSM client:
  - osm sdnc-create --name etsi_DSS9000_fl --type floodlight --ip_address 172.21.2.23 --port 8080 --switch_dpid 00:01:64:00:6a:e6:b3:14
- Port mapping file:
  - https://osm-download.etsi.org/ftp/osm-5.0-five/5th-hackfest/other/port-mapping-openstack2.yaml
  - From the OSM host, download port-mapping:
    - wget https://osm-download.etsi.org/ftp/osm-5.0-five/5th-hackfest/other/port-mapping-openstack2.yaml
  - osm vim-update etsi-openstack-epa --sdn_controller etsi_DSS9000_fl --sdn_port_mapping port-mapping-etsi-openstack2.yaml

VNF diagram
Changes highlighted in yellow

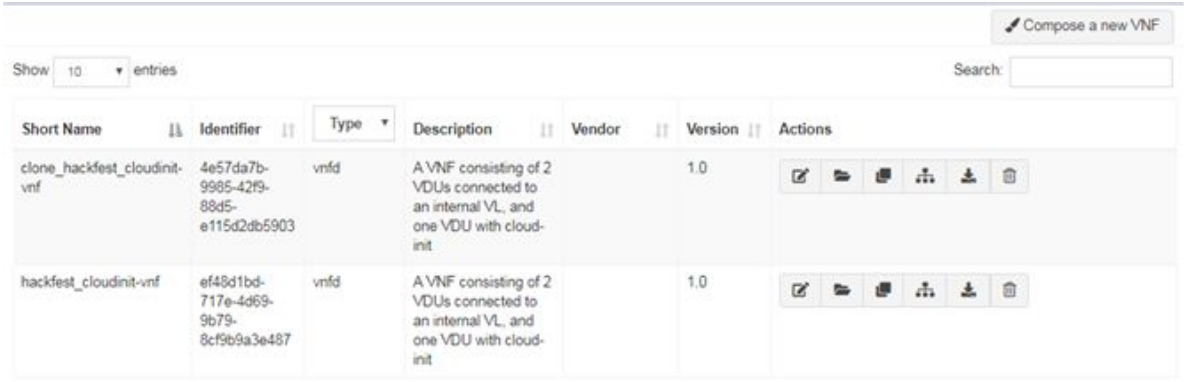


User Interface

- Clone hackfest_cloudInit_vnf in the user interface



- A new hackfest_cloudinit_vnf appears:



Creating the VNFD (1/3)

- Edit the new descriptor

- Modify the name and id: hackfest_epasriov_vnfd

- Modify VDU mgmtVM:
- guest-epa:
  - mempage-size: LARGE
  - cpu-pinning policy: DEDICATED
  - cpu-thread-pinning-policy: PREFER (HW threads) or ISOLATE (cores)
  - numa-node-policy:
    - node-cnt: 1
    - mem-policy: STRICT
    - node:
      - - id: 1

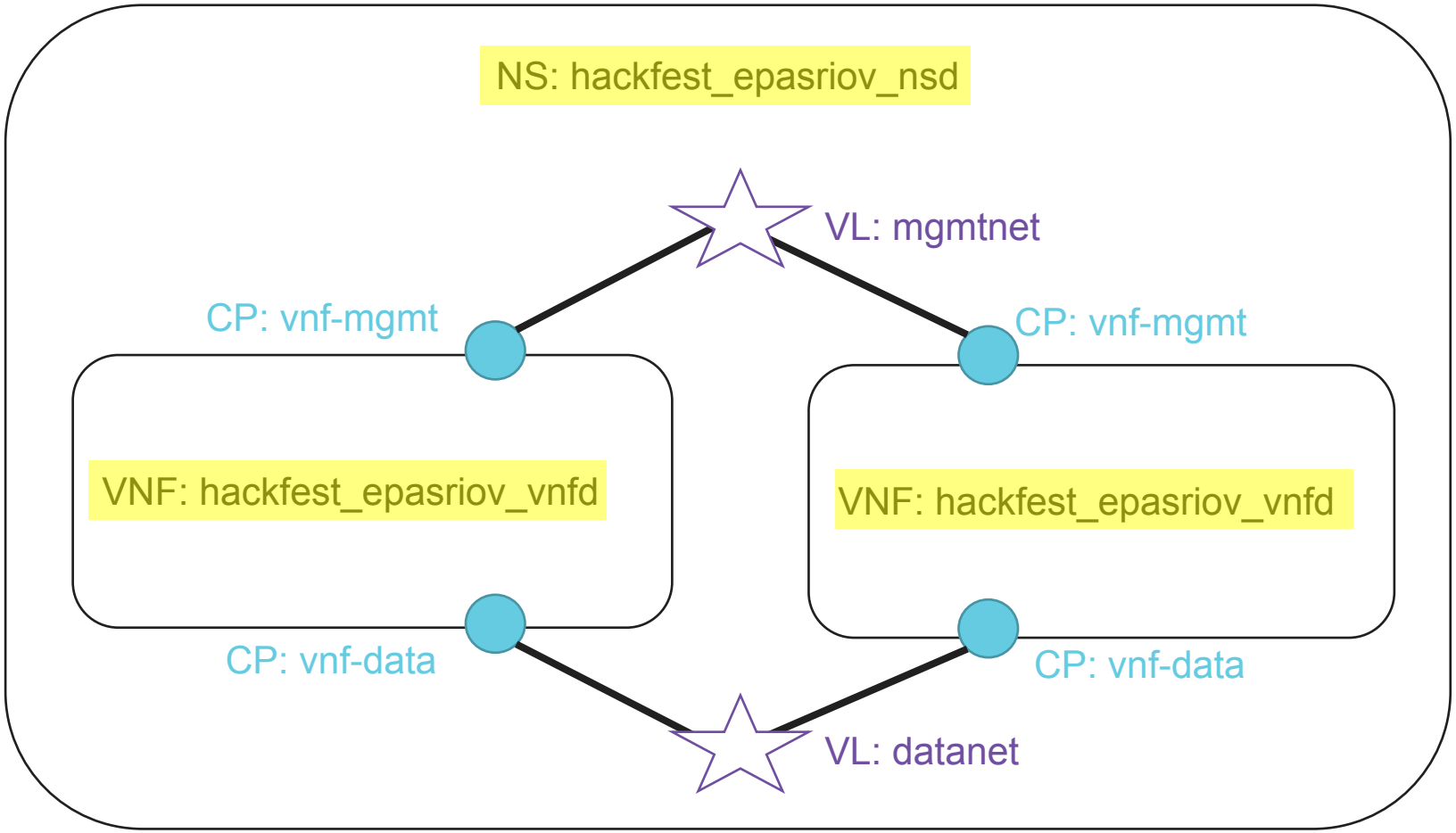
Creating the VNFD (2/3)

- Modify VDU dataVM:
  - Image name: hackfest-pktgen
  - Flavor:
    - 8 CPUs
    - 4096 MB RAM
    - 10 GB disk
  - guest-epa:
    - mempage-size: LARGE
    - cpu-pinning policy: DEDICATED
    - cpu-thread-pinning-policy: PREFER (HW threads) or ISOLATE (cores)
    - numa-node-policy:
      - node-cnt: 1
      - mem-policy: STRICT
      - node:
        - id: 1
  - Modify interfaces of the VDU
    - Interface 1:
      - Name: eth0
    - Interface 2:
      - Name: xe0
    - Virtual-interface:
      - Type: SR-IOV

Creating the VNFD (3/3)

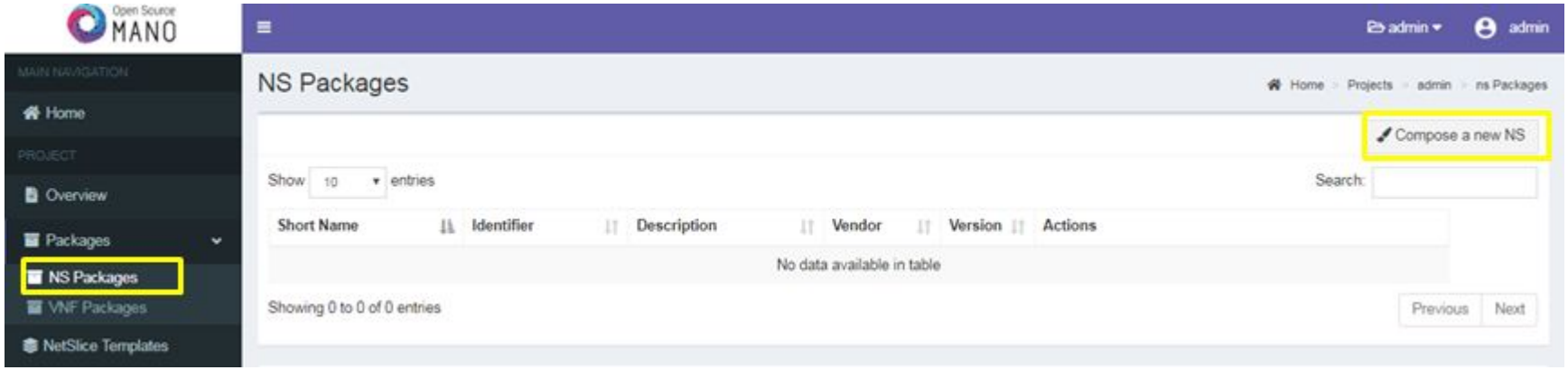
- And finally, this is the sample file:
  Hackfest EPA SRIOV VNF Descriptor -
https://osm-download.etsi.org/ftp/osm-5.0-five/5th-hackfest/packages/hackfest_epasriov_vnf.tar.gz

NS diagram
Changes highlighted in yellow



User Interface

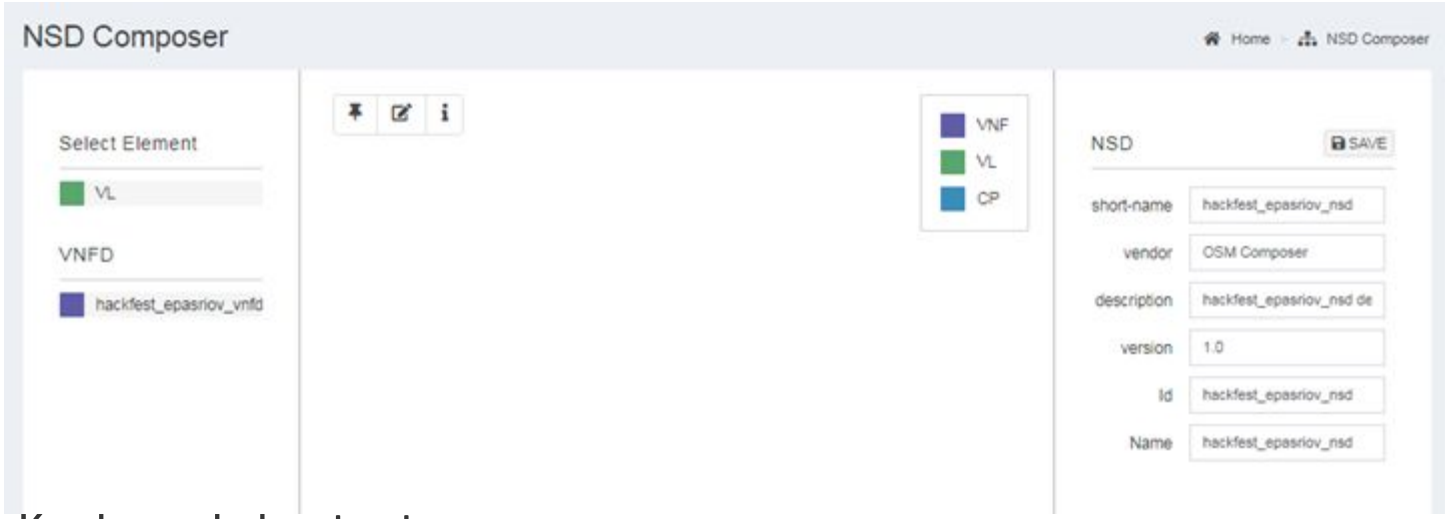
- Steps:
  - Compose a new NS



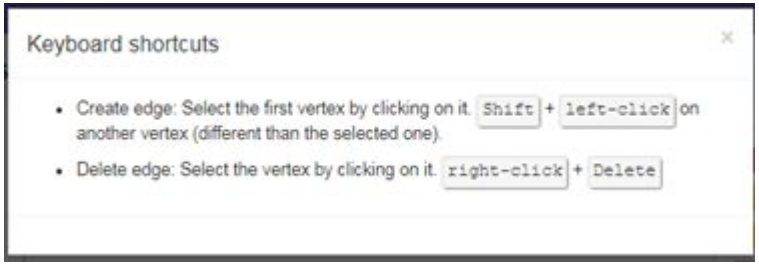
  - Create new Package

NSD Composer

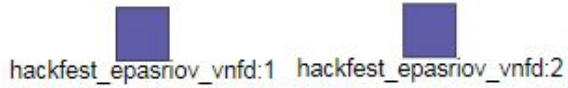
- Steps
  - NSD Composer



  - Keyboard shortcuts



Creating the NSD (1/3)

- Steps
  - Select VNFs: 

(Drag and drop)

VNF

member-vnf-index 1

vnfd-id-ref hackfest_epasriov_vnfd

VNF

member-vnf-index 2

vnfd-id-ref hackfest_epasriov_vnfd

  - Create VLs: 

(Drag and drop)

Virtual Link SAVE

Vim network name PUBLIC

Name mgmtnet

Mgmt network true

Type ELAN

Id mgmtnet

Virtual Link SAVE

Vim network name

Name datanet

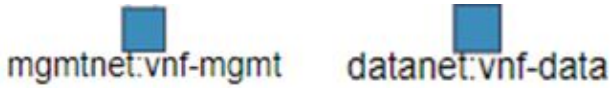
Mgmt network false

Type ELAN

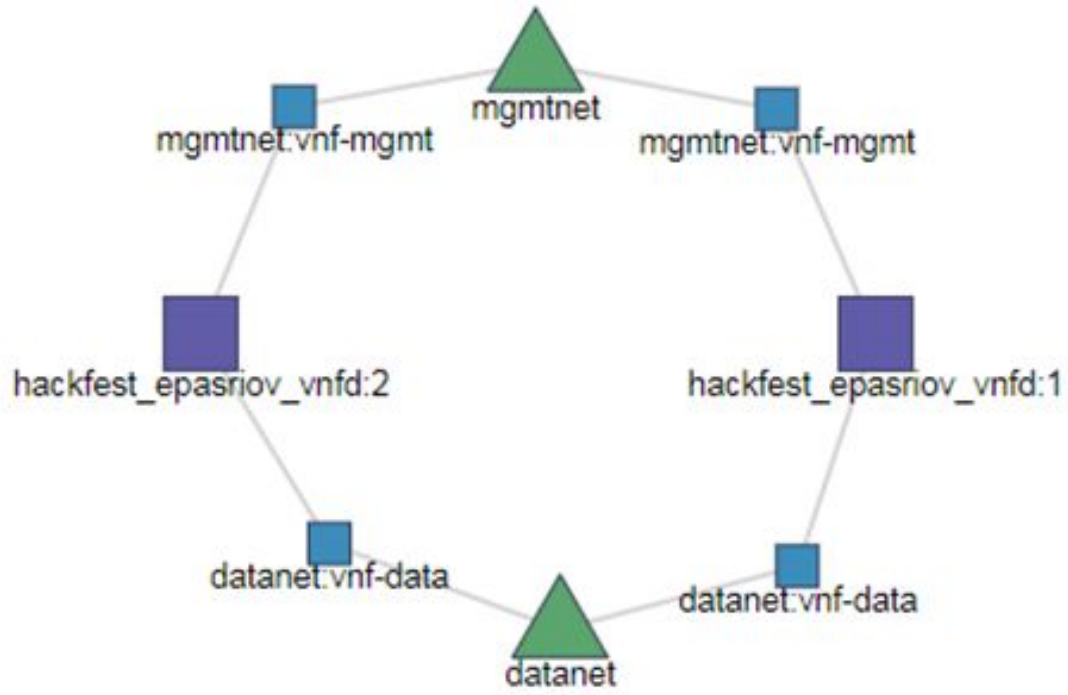
Id datanet

Creating the NSD (2/3)

- Steps
  - Link VLs with VNFs (Shift + Left Click)
    - Select the name for the CPs (vnf-data and vnf-mgmt)



- Final Scenario



Creating the NSD (3/3)

- And finally, against the sample file:
  Hackfest EPA SRIOV NS Descriptor -
https://osm-download.etsi.org/ftp/osm-5.0-five/5th-hackfest/packages/hackfest_epasriov_ns.tar.gz

Deploying NS in the UI (1/3)

- Onboard VNFD and NSD to catalog using the UI
- Launch the NS from the UI
  - Depending on the VIM, specify a VIM network name to map MGMTNET
  - If you need to change the VIM, change the network name using config:
    {vld: [{name: mgmtnet, vim-network-name: PUBLIC}]}
- Click the info button to see the mgmt IP address of each VNF
- Connect to each VNF:
  - ssh ubuntu@<IP> (pwd: osm4u)

Deploying NS in the UI (2/3)

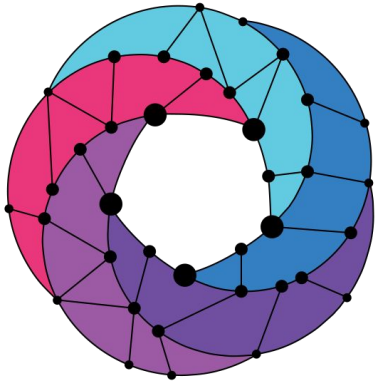
- SSH to the mgmtVM of one of the VNFs
- Check that cloud-init worked in mgmtVM
- Check interface names in mgmtVM
  - They must be different than the ones in the descriptor (no udev metadata service)
- Jump to dataVM:
  - ssh -i test4.pem ubuntu@IPAddrInternalNetDataVM
- Check interface names in dataVM
  - They must be the same names as in the descriptor
- Run ‘pktgen’

Deploying NS in the UI (3/3)

- SSH to the mgmtVM of the second VNF
- Check that cloud-init worked in mgmtVM
- Check interface names in mgmtVM
  - They must be different than the ones in the descriptor (no udev metadata service)
- Jump to dataVM:
  - ssh -i test4.pem
- Check interface names in dataVM
  - They must be the same names as in the descriptor
- Run ‘pktgen’

Instructions to run ‘pktgen’

- Configure interfaces to use DPDK:
  - cd /opt/dpdk-17.11
  - sudo -E ./usertools/dpdk-devbind.py --status
  - sudo -E ./usertools/dpdk-devbind.py --bind=igb_uio xe0
- Run pktgen:
  - cd /opt/pktgen-3.4.5/
  - sudo -E ./app/x86_64-native-linuxapp-gcc/pktgen -n 3 -l 2-7 -- -P -m "[4-7].0"
- From pktgen console, set the destination MAC address to the one used by xe0 interface in the other VNF, and send traffic:
  - set all dst mac xx:xx:xx:xx:xx:xx <- destination MAC where to send traffic
  - start all <- to start traffic
  - stop all <- to stop traffic



Open Source
MANO

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

ETSI