

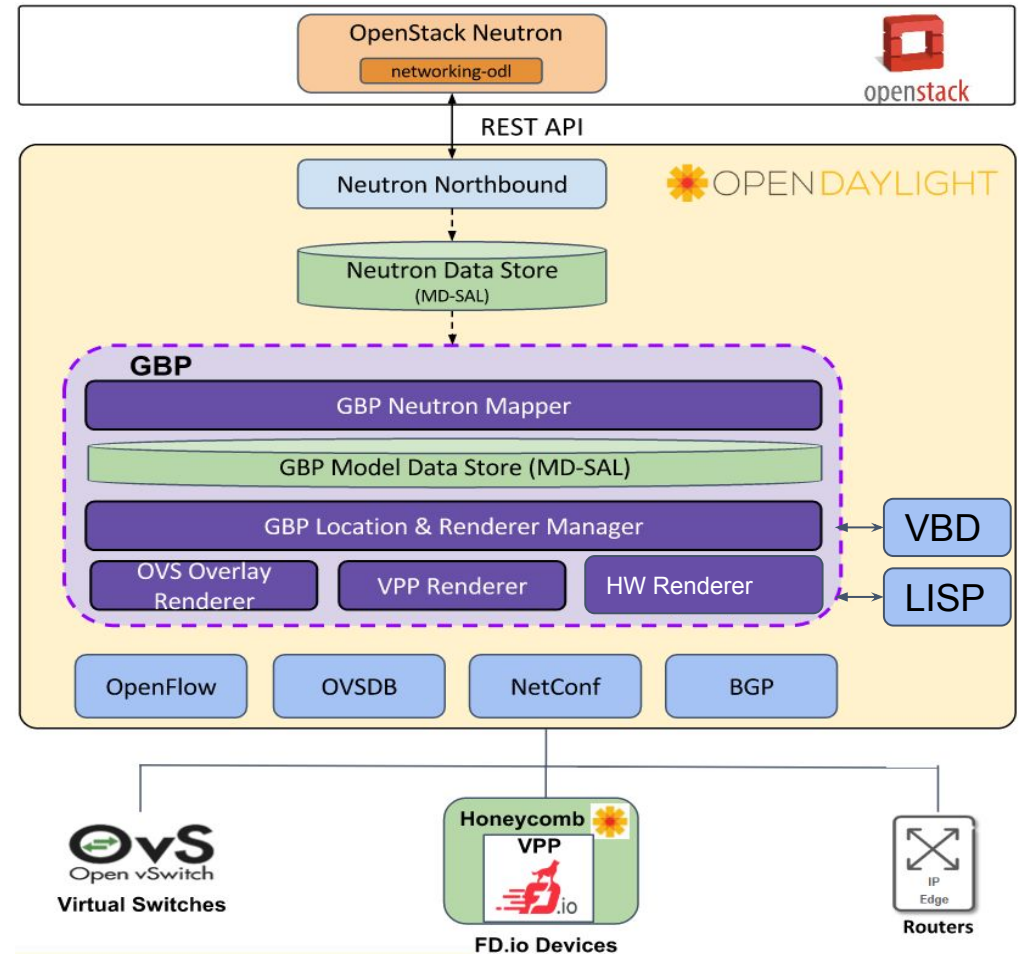
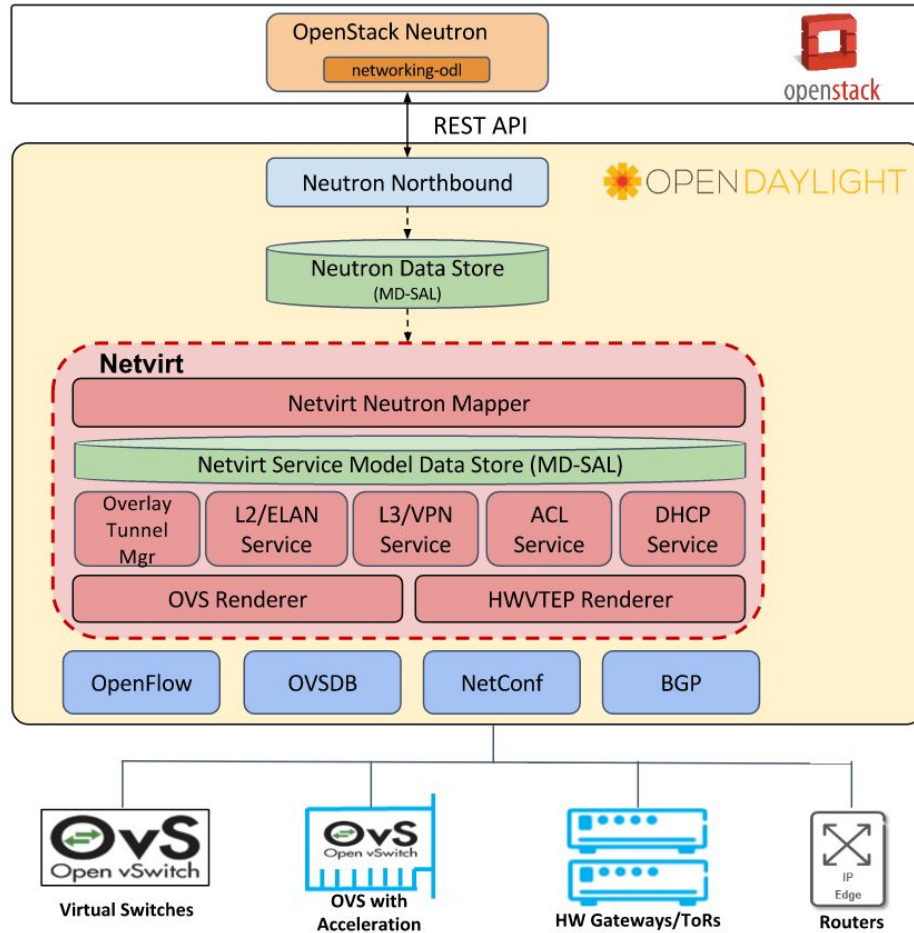


Realization of Nirvana Stack: Integration of OpenDaylight and FD.io

Srikanth Vavilapalli, Ericsson; Frank Brockners, Cisco
OpenStack Summit 2017- Boston

Recap - what we have

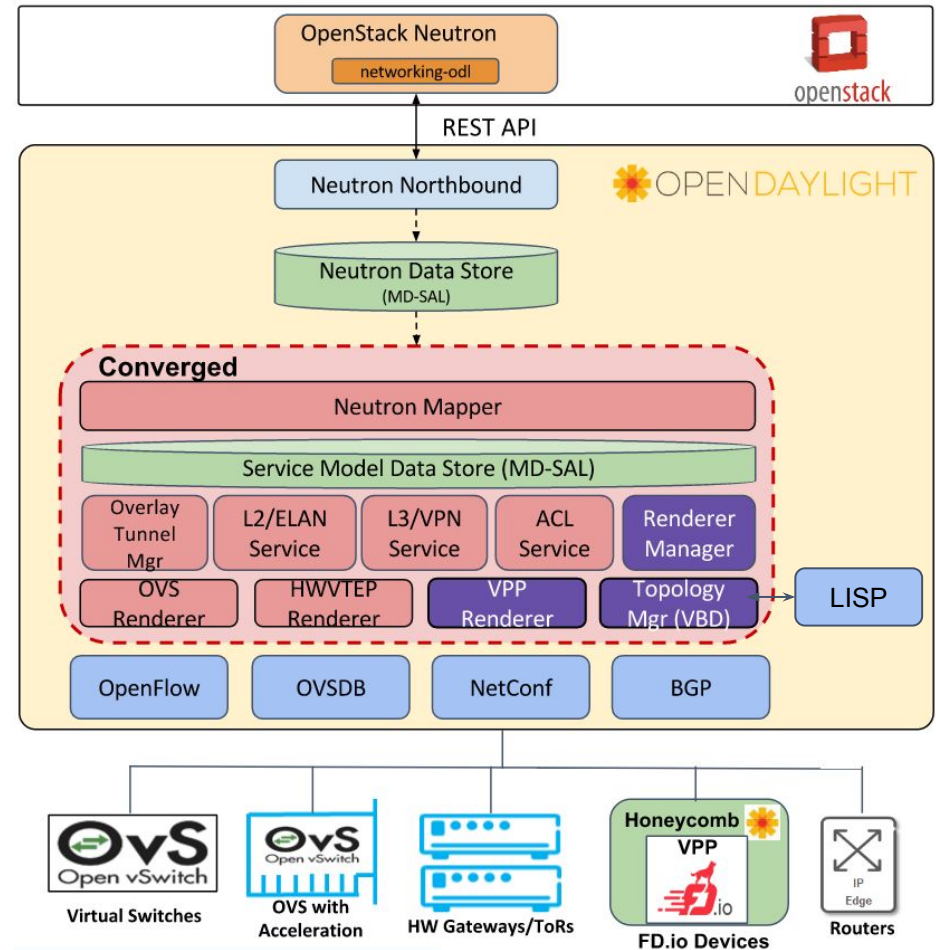
Two Control Plane solutions in OpenDaylight



Recap - what we want

Integrated Control Solution in OpenDaylight

- **Nirvana**: Enable rapid innovation in all layers of the stack:
 - Combine GBP+NetVirt into an integrated solution
- Services: L2, L3, BGP L3VPN, EVPN, ACL, DHCP, QoS, SFC, IPv6, L2GW
- Diverse set of forwarders: OpenFlow and OVSDB based devices, Netconf/YANG based devices (FD.io, physical routers and switches)
- BGP and Netconf to interwork with physical routers and switches



OpenDaylight Integrated Control Solution

Proof-of-Concept in ODL community

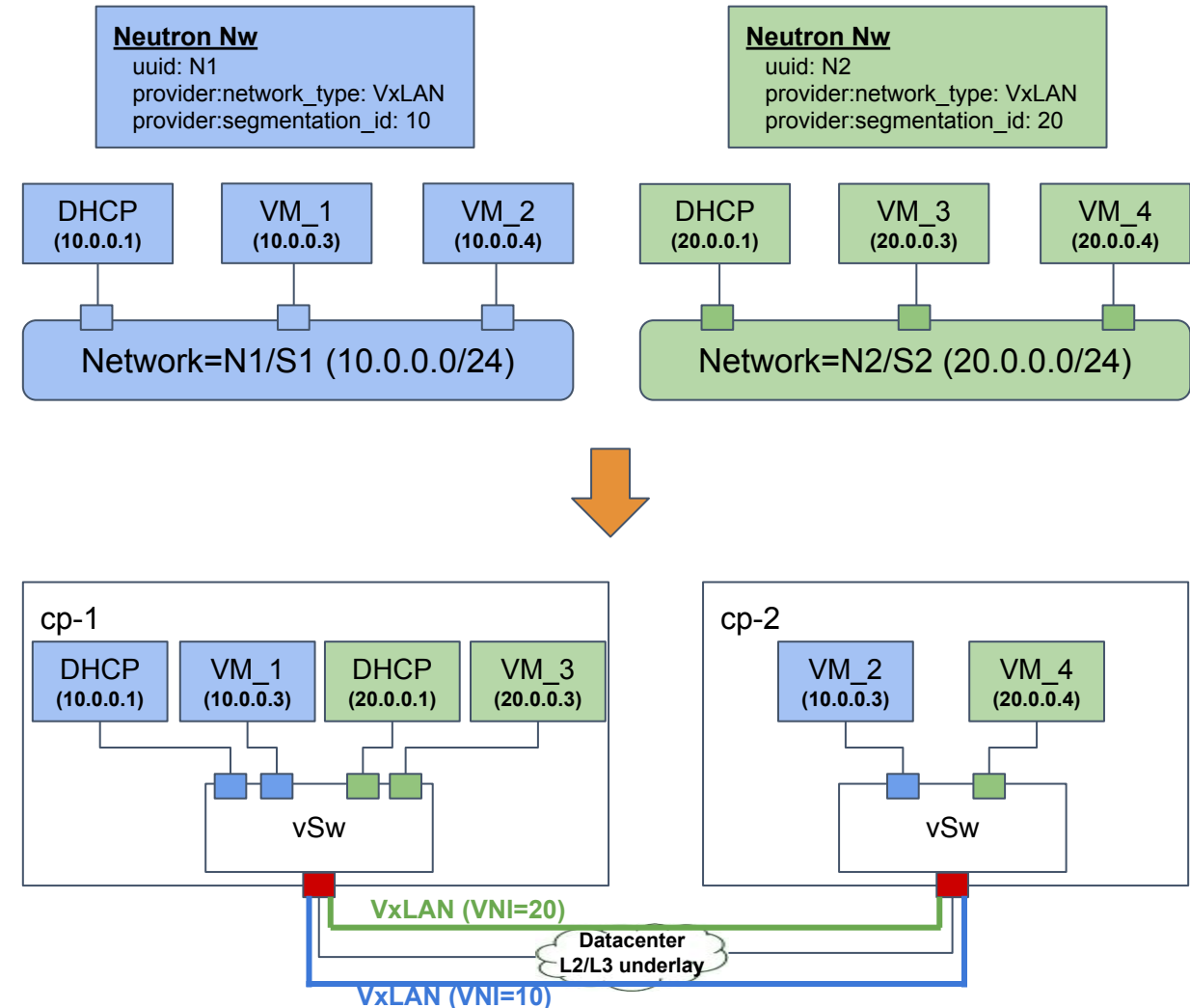
- Objective: Combine ODL NetVirt and GBP solutions for a simple service like ELAN with VPP as forwarder
- ELAN/L2 Service
 - Layer 2 connectivity between OpenStack VMs using VxLAN overlay
 - Security Groups
 - VLAN transparency/trunking
 - Q-in-Q
 - Connectivity to legacy BM/SR-IOV appliances

Agenda

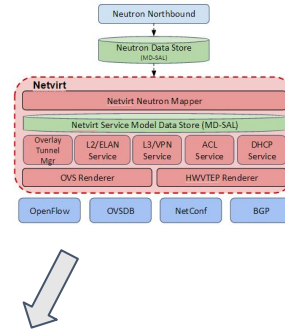
- ELAN/L2 Service
- How ELAN/L2 Service works in ODL NetVirt solution
- How ELAN/L2 Service works in ODL GBP solution
- ELAN/L2 Service architecture in ODL Integrated solution
- OPNFV Full Stack Integration
- Demo - 10min

ELAN/L2 Service

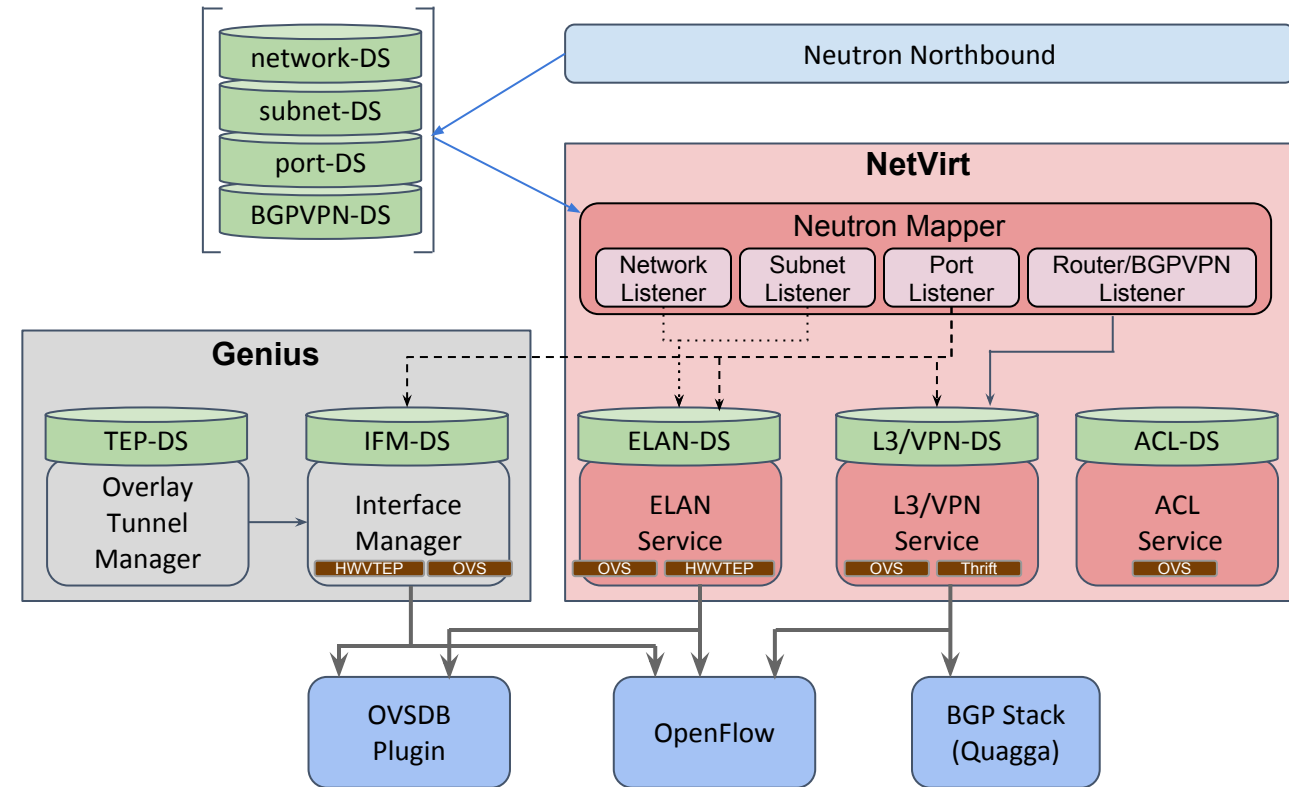
- OpenStack Orchestration:
 - Create Network
 - `neutron net-create <name>`
 - Create Subnet in Network with DHCP enabled
 - `neutron subnet-create <net-id>`
 - Create VMs in Network
 - `nova boot --nic net-id=<net-id>`



ODL NetVirt Architecture

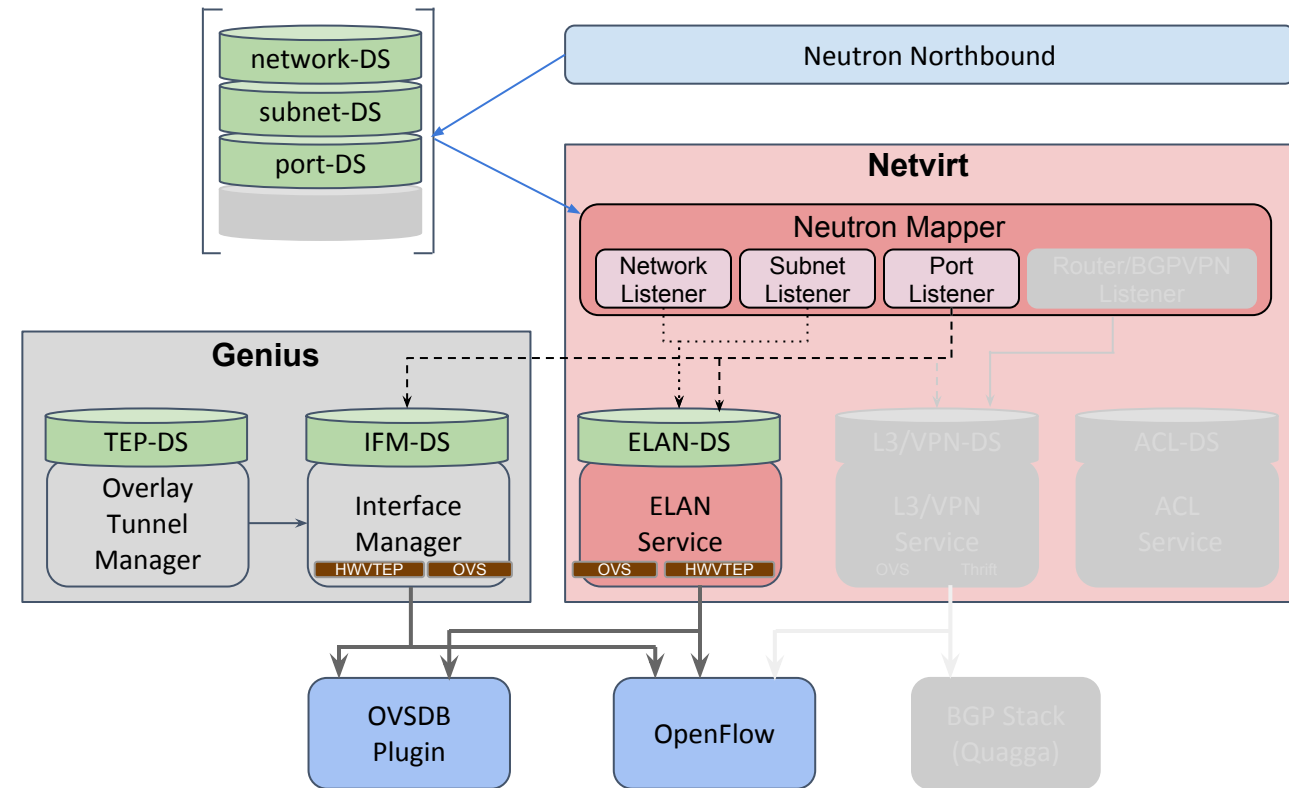


- NetVirt listens to MD-SAL DCNs of Neutron Data Store events
- Neutron Mapper maps neutron constructs into NetVirt specific models
 - Network CRUD → ELANInstance models
 - Router and BGPVPN CRUD → VPNInstance models
 - Port CRUD → ELANInterface & VPNInterface models
- NetVirt services operate on service models and renderers translate them into device specific commands



ODL NetVirt ELAN Service

- Pt-to-pt and Pt-to-multipoint L2 forwarding service (MAC lookup only)
- Operates on ELAN Yang Models
 - ELANInstance, ElanInterface
- Forwarding between switches using VxLAN or VLAN mechanisms
- Maintains local and remote broadcast groups per switch for flooding
- Supports OVS and HWVTEP renderers



NetVirt ELAN Service Yang Models

OpenStack CLI

Neutron Nw

uuid: N1
provider:network_type: VxLAN
provider:segmentation_id: 10

Neutron Subnet

uuid: S1
Network: N1
subnet: 10.0.0.0/24
Enable-dhcp: true

Nova VM

uuid: VM1
Network: N1

OpenStack Neutron

Neutron Nw

uuid: N1
provider:network_type: VxLAN
provider:segmentation_id: 10

Neutron Subnet

uuid: S1
Network: N1

DHCP Port

uuid: P1
vif_type: normal
device-owner: dhcp
fixed-ip: 10.0.0.1
Mac-addr: M1

Nova Port

uuid: P2
vif_type: normal
device-owner: compute
fixed-ip: 10.0.0.2
Mac-addr: M2

ELAN Models

ELANInstance

Name: N1
elan-tag: 1234
segment_type: VxLAN
segment_id: 10

Interface

Name: P1
Type: I2_vlan
Parent-refs: host-id

ELANInterface

Name: P1
Elan-instance: N1
Mac-addr: M1

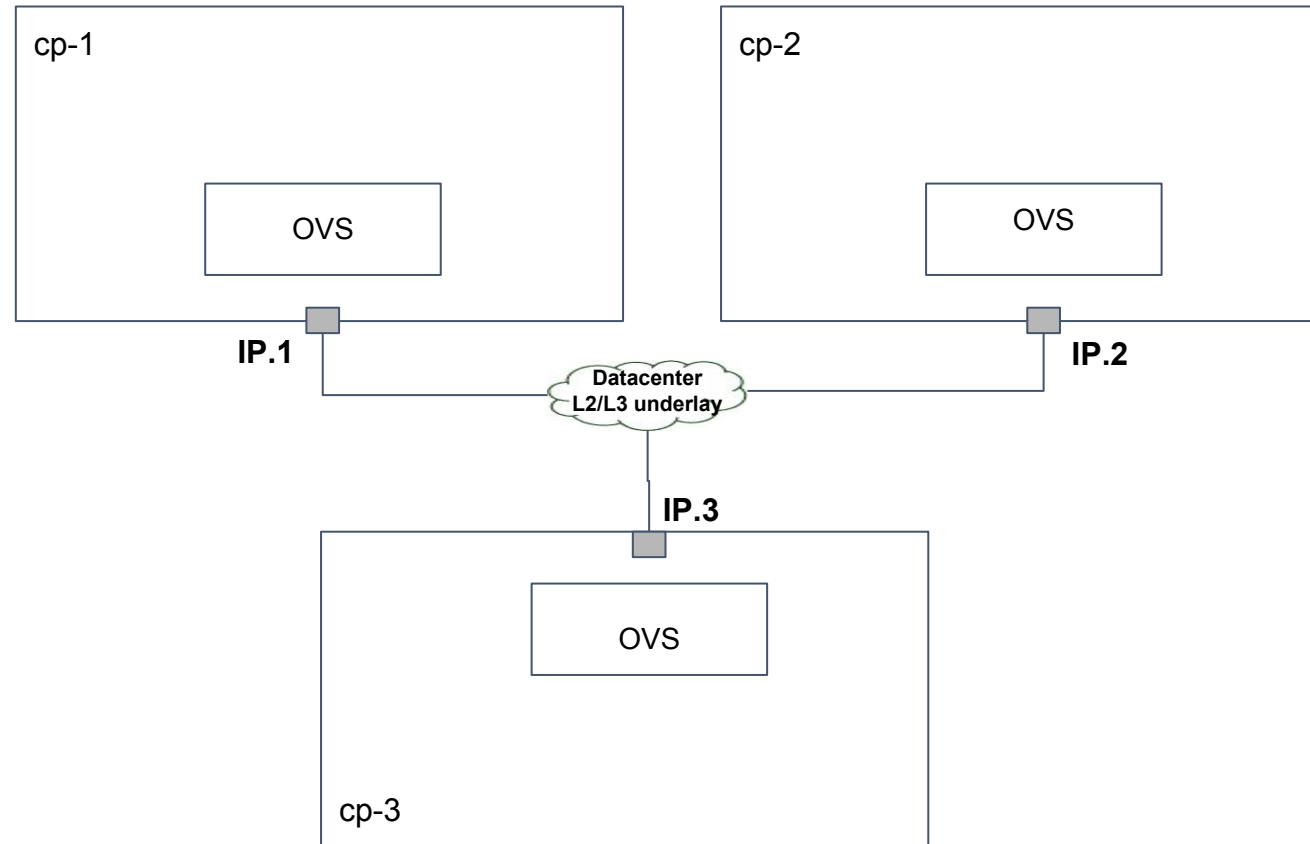
Interface

Name: P2
Type: I2_vlan
Parent-refs: host-id

ELANInterface

Name: P2
Elan-instance: N1
Mac-addr: M2

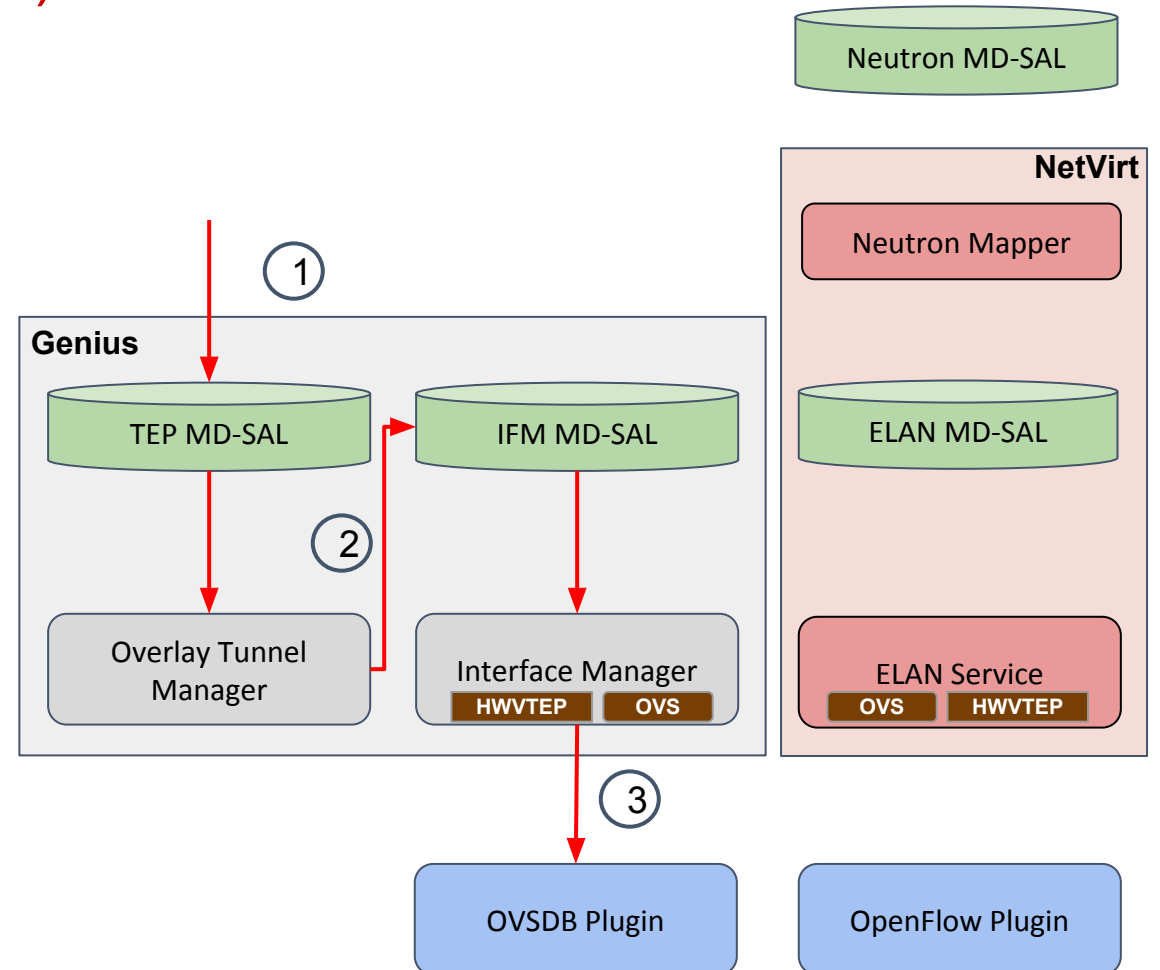
ELAN Service Workflow



ELAN Service Workflow

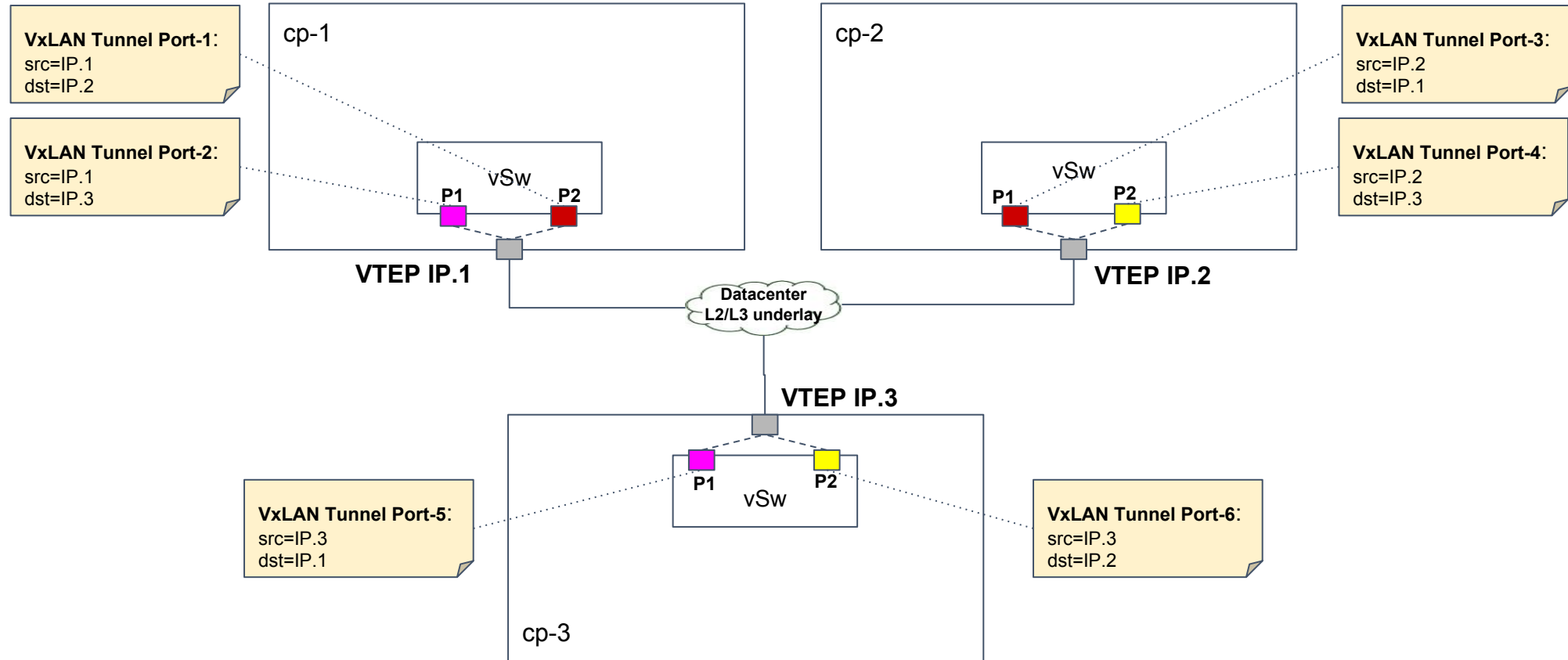
VxLAN Tunnel Creation (Config/Dynamic)

1. NetVirt configures VTEP data store to create a tunnel mesh
2. Tunnel manager listens on DCNs from VTEP data store and translate them into tunnel-interface yang models
3. Renderers in Interface manager send commands to OVSDB plugin to configure the VTEP ports in respective nodes



ELAN Service Workflow

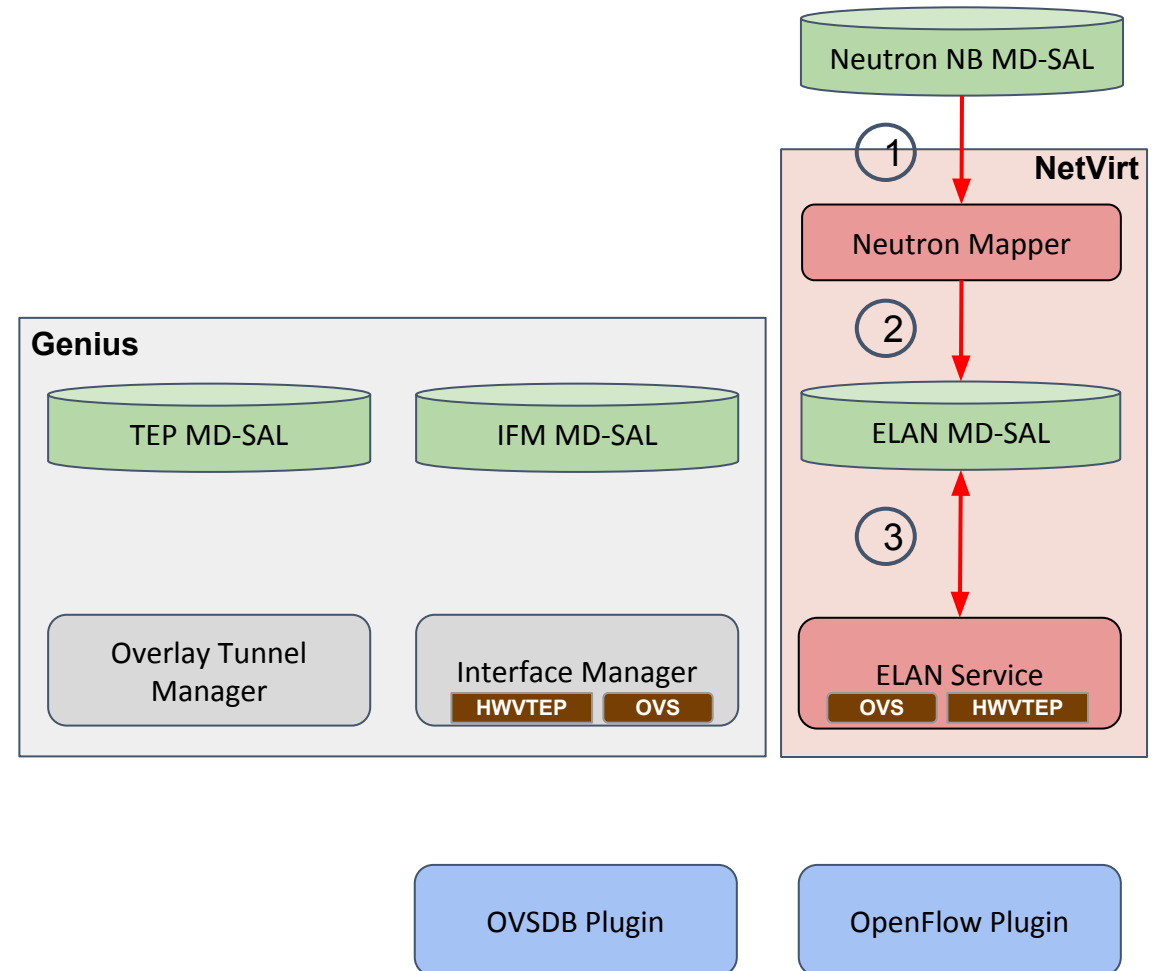
VxLAN Tunnel Creation (Config/Dynamic)



ELAN Service Workflow

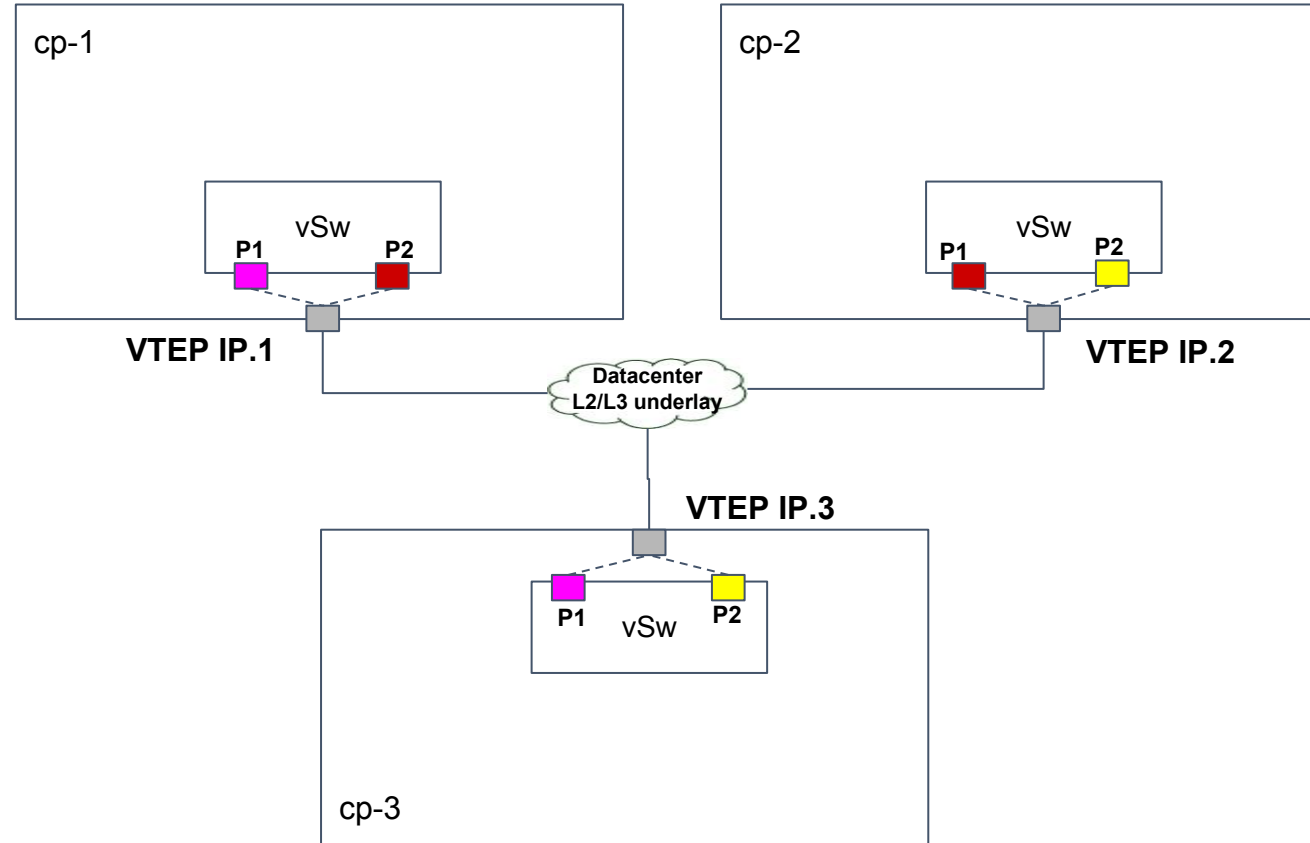
Neutron Network Create

1. NetVirt Neutron Mapper listens on DCNs from Neutron Northbound for network creation
2. Neutron Mapper create a new ELANInstance in ELAN data store
3. ELAN service is activated, it allocates an ELAN tag for this instance
 - a. The ELAN tag is a dataplane identifier for the ELAN instance.



ELAN Service Workflow

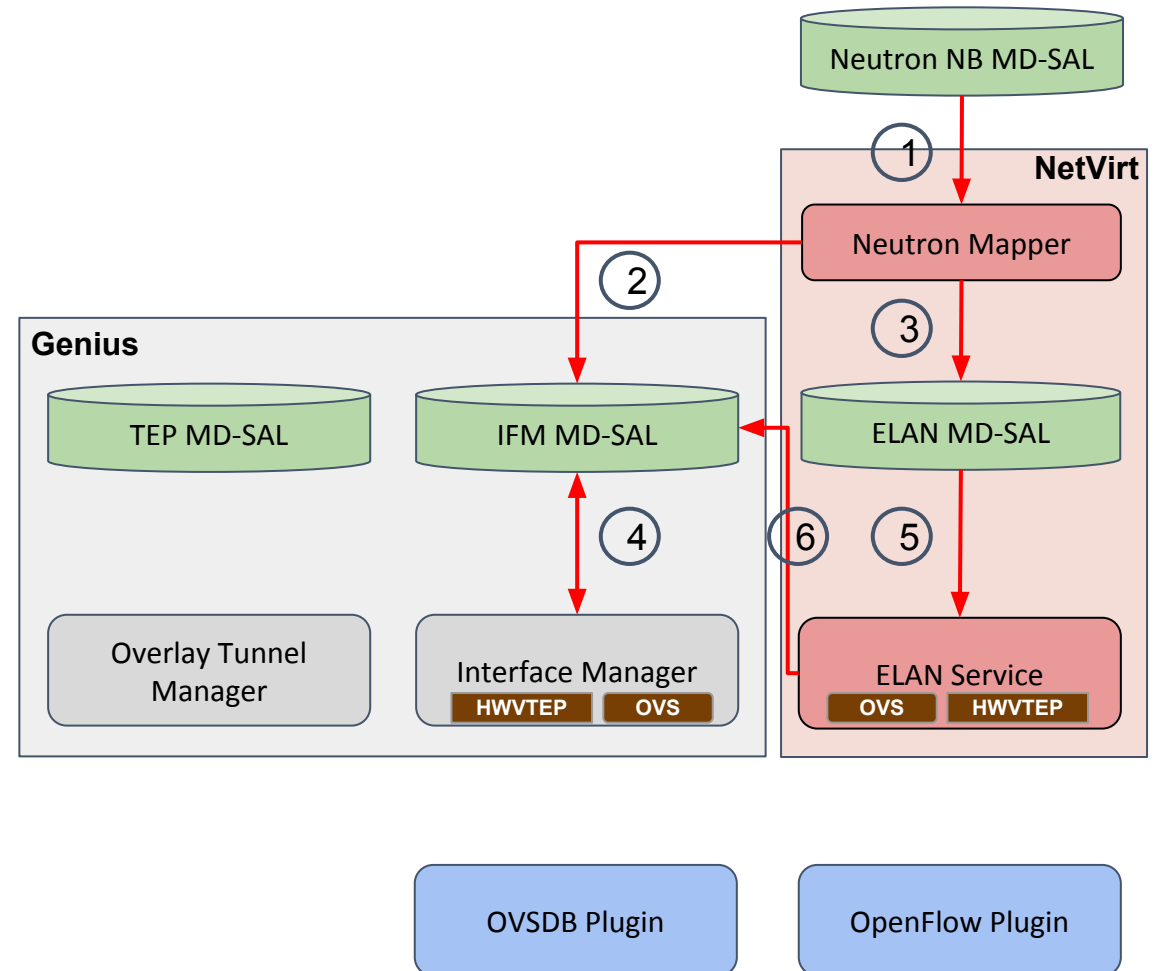
Neutron Network Create - No action in data plane



ELAN Service Workflow

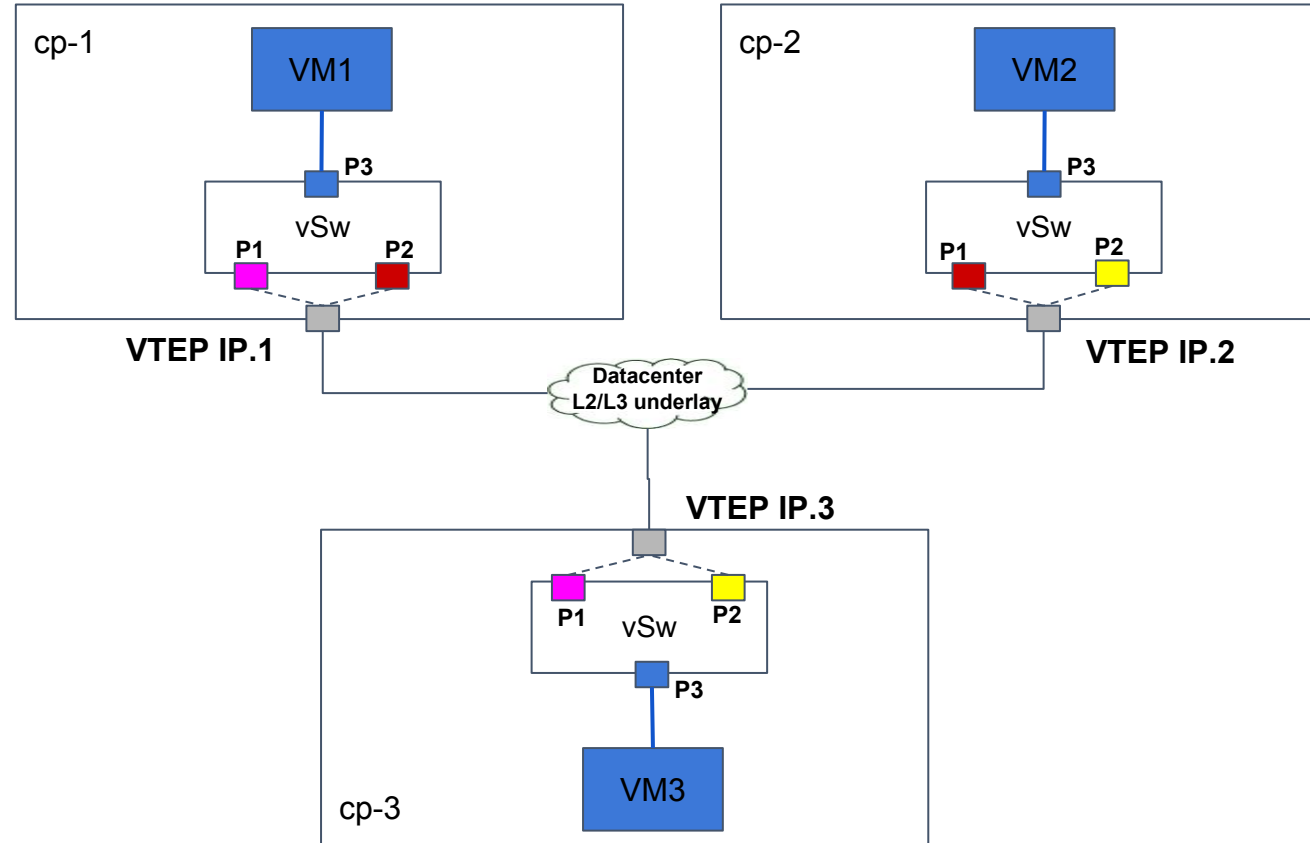
Neutron Port Create

1. Neutron Mapper receives DCNs from Neutron Northbound for port creation
2. Neutron Mapper create a new Interface object in Interface Manager data store with Neutron port UUID as the interface name
3. Neutron Mapper creates an ELANInterface object in ELAN data store for this port
4. Interface manager receives DCN and allocates an lport tag for the interface
5. ELAN service receives a DCN for ELANInterface creation
6. ELAN registers for interface state DCNs with Interface Manager



ELAN Service Workflow

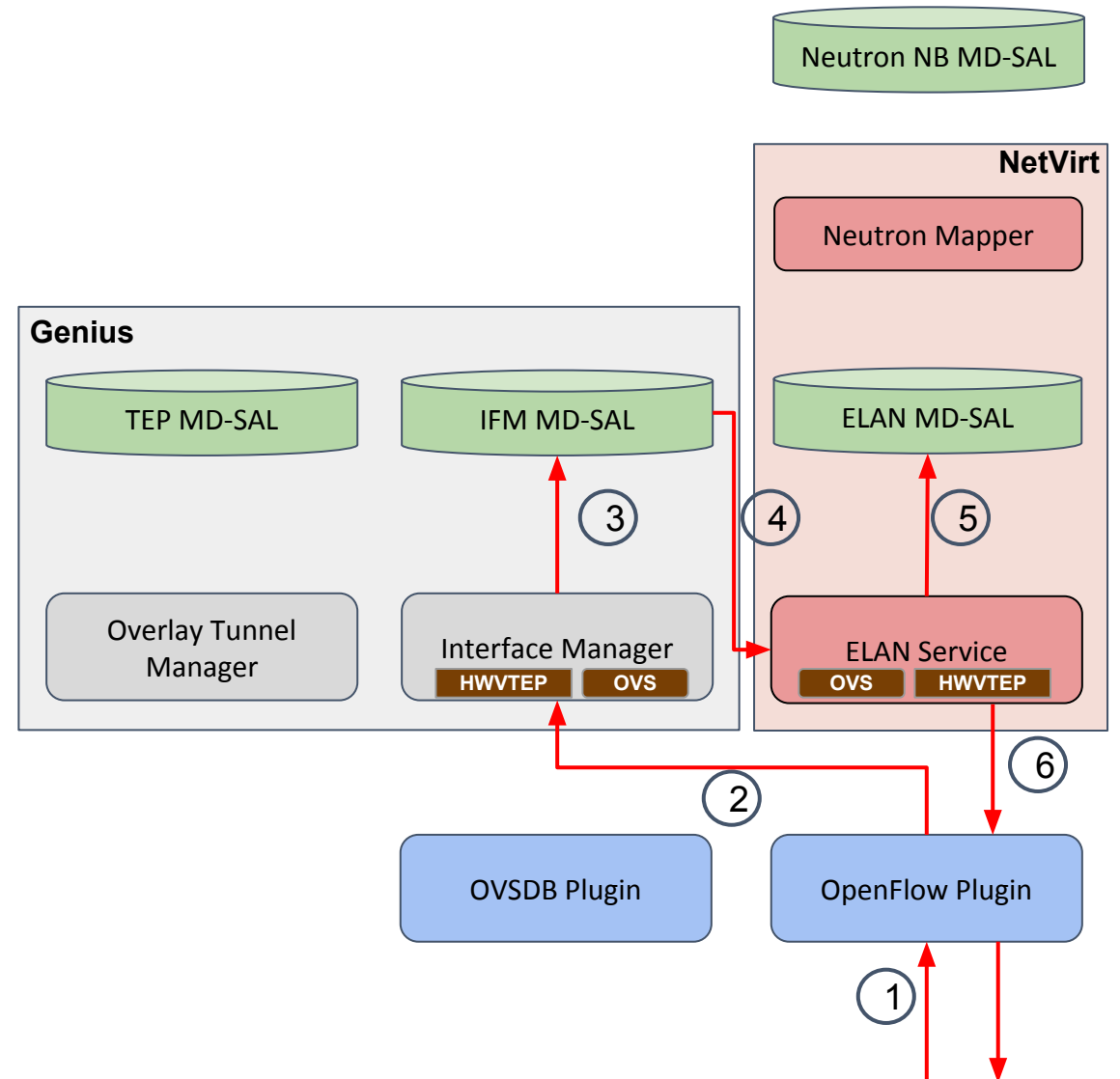
Neutron Port Create - Nova creating tap ports and attach to VMs



ELAN Service Workflow

Neutron Port UP Notification

1. OF Port UP notification from data plane when the port is attached to VM
2. Interface manager receives Port UP notification
3. Interface manager updates the parent-ref for the interface with the OF port and data plane ID
4. ELAN service receives a DCN for the same, with the data plane ID for the interface
5. ELAN service updates its operational data store with list of all data plane nodes per ELAN instance
6. Renderers in ELAN service send commands to populate SMAC, DMAC and Split Horizon tables in the all the relevant data plane devices using respective plugins



ELAN Service Workflow

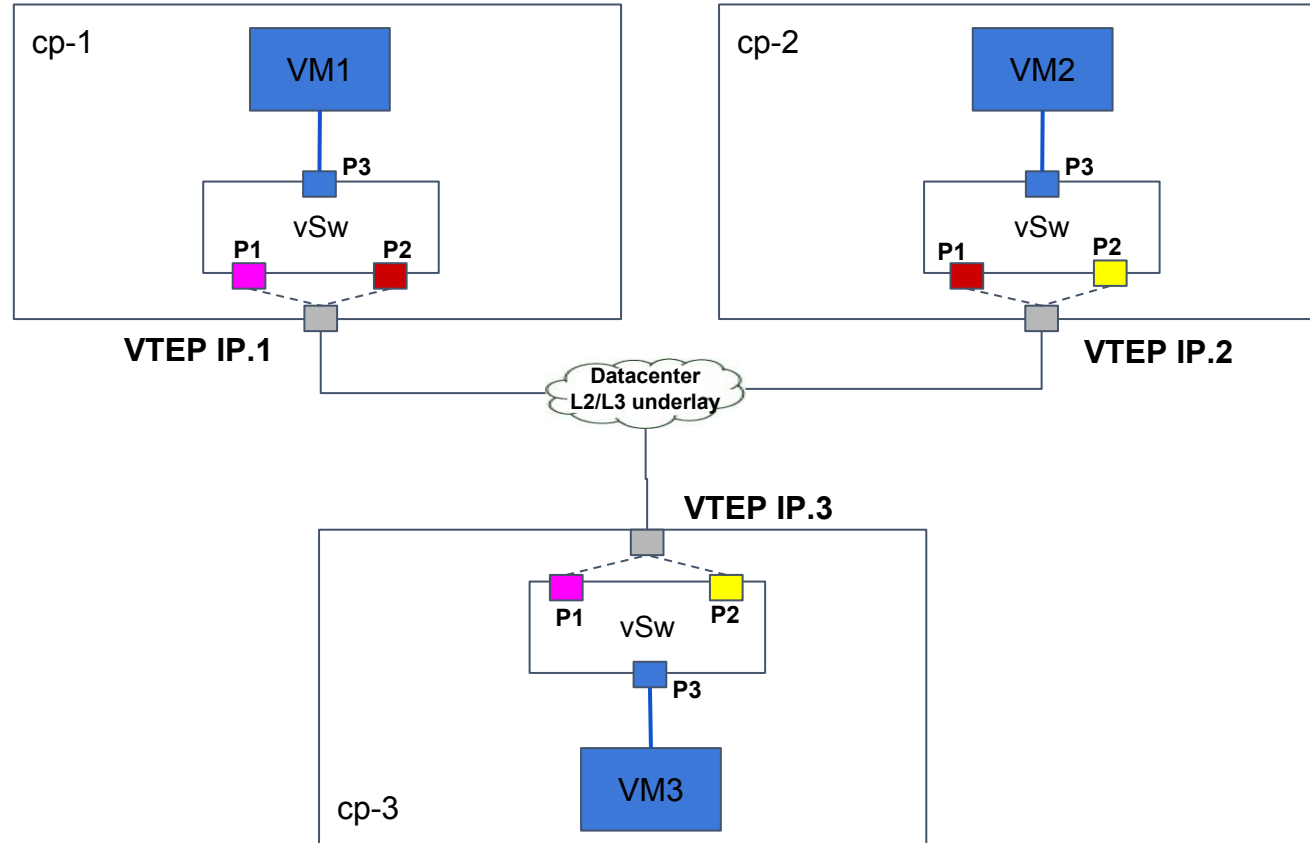
Neutron Port UP Notification - Flow Installation

SMAC Table

Match	Action
ELAN-Tag MAC= VM1	goto DMAC Table
Miss	punt to ODL

DMAC Table

Match	Action
ELAN-Tag, MAC= VM1	output=p3
ELAN-Tag, MAC=VM2	vni=p3, output=p2
ELAN-Tag, MAC=VM3	vni=p3, output=p1



SMAC Table

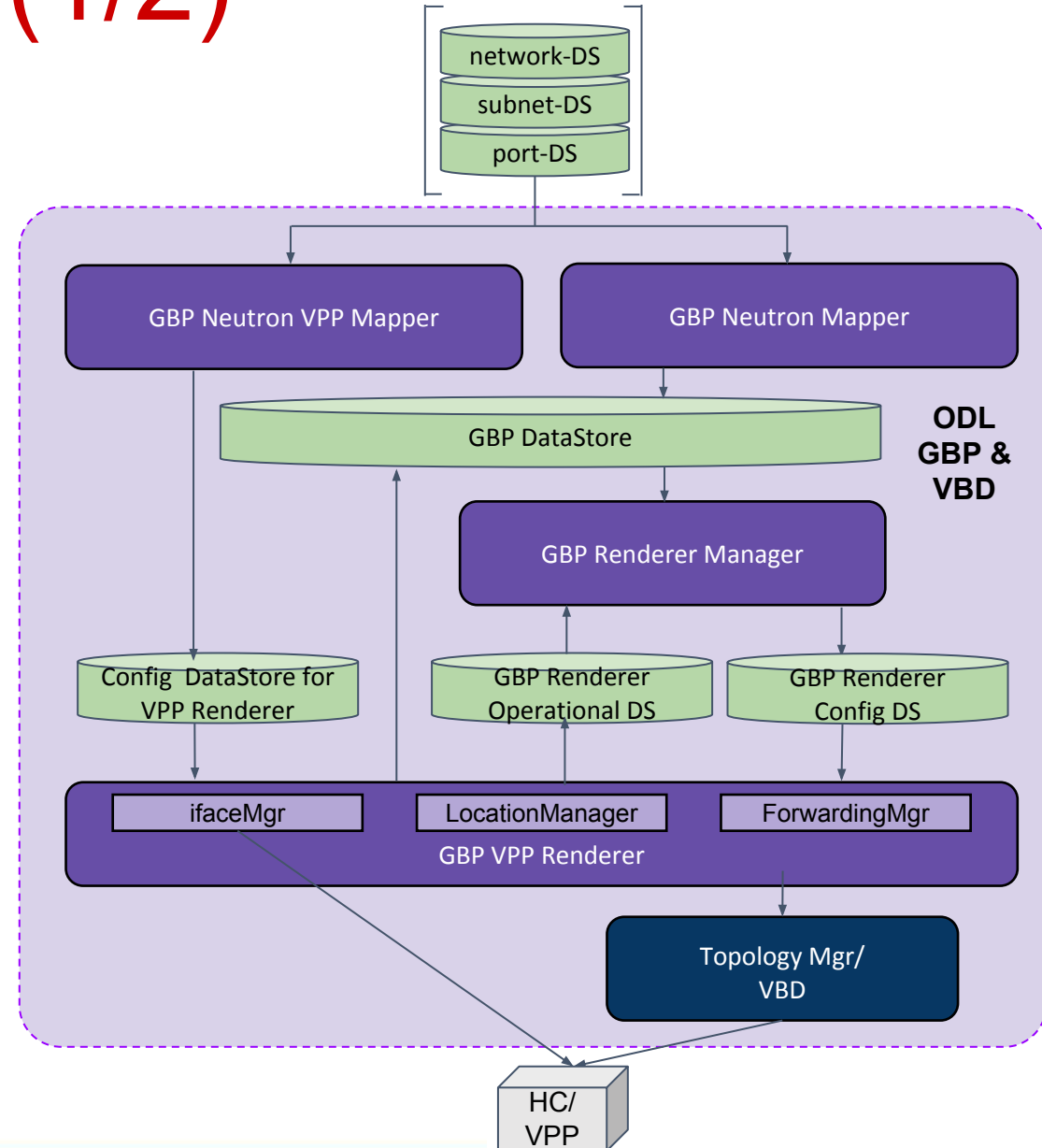
Match	Action
ELAN-Tag MAC= VM2	goto DMAC Table
Miss	punt to ODL

DMAC Table

Match	Action
ELAN-Tag, MAC= VM2	output=p3
ELAN-Tag, MAC=VM1	vni=p3, output=p1
ELAN-Tag, MAC=VM3	vni=p3, output=p2

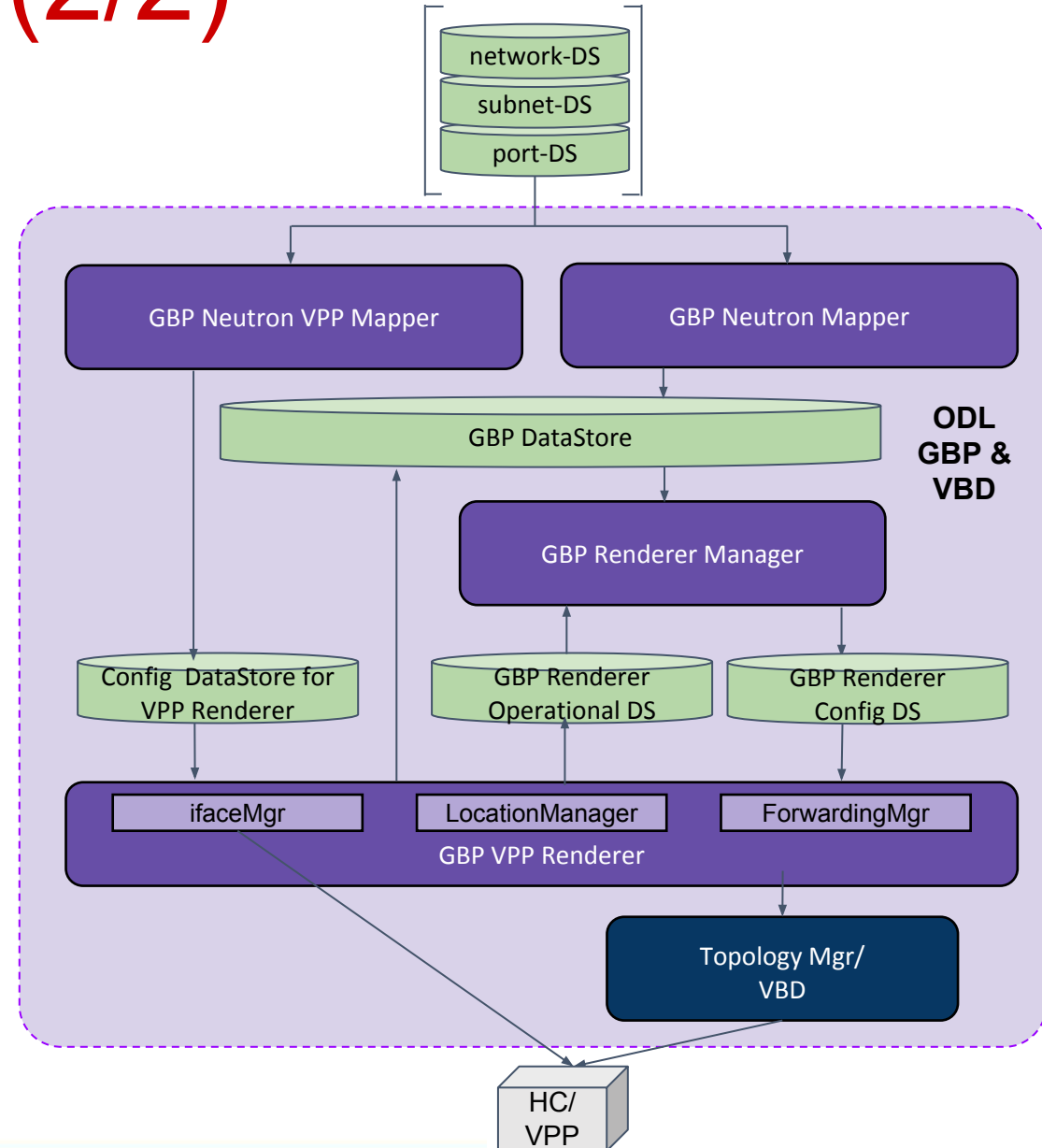
ODL GBP Architecture (1/2)

- GBP Neutron Mapper: Maps neutron elements like networks, subnets, security-groups...etc to GBP entities
- GBP Neutron VPP Mapper: Maps VPP specific data of neutron ports/networks to GBP VPP Renderer model objects
- GBP Renderer Manager: Creates configuration for Renderers

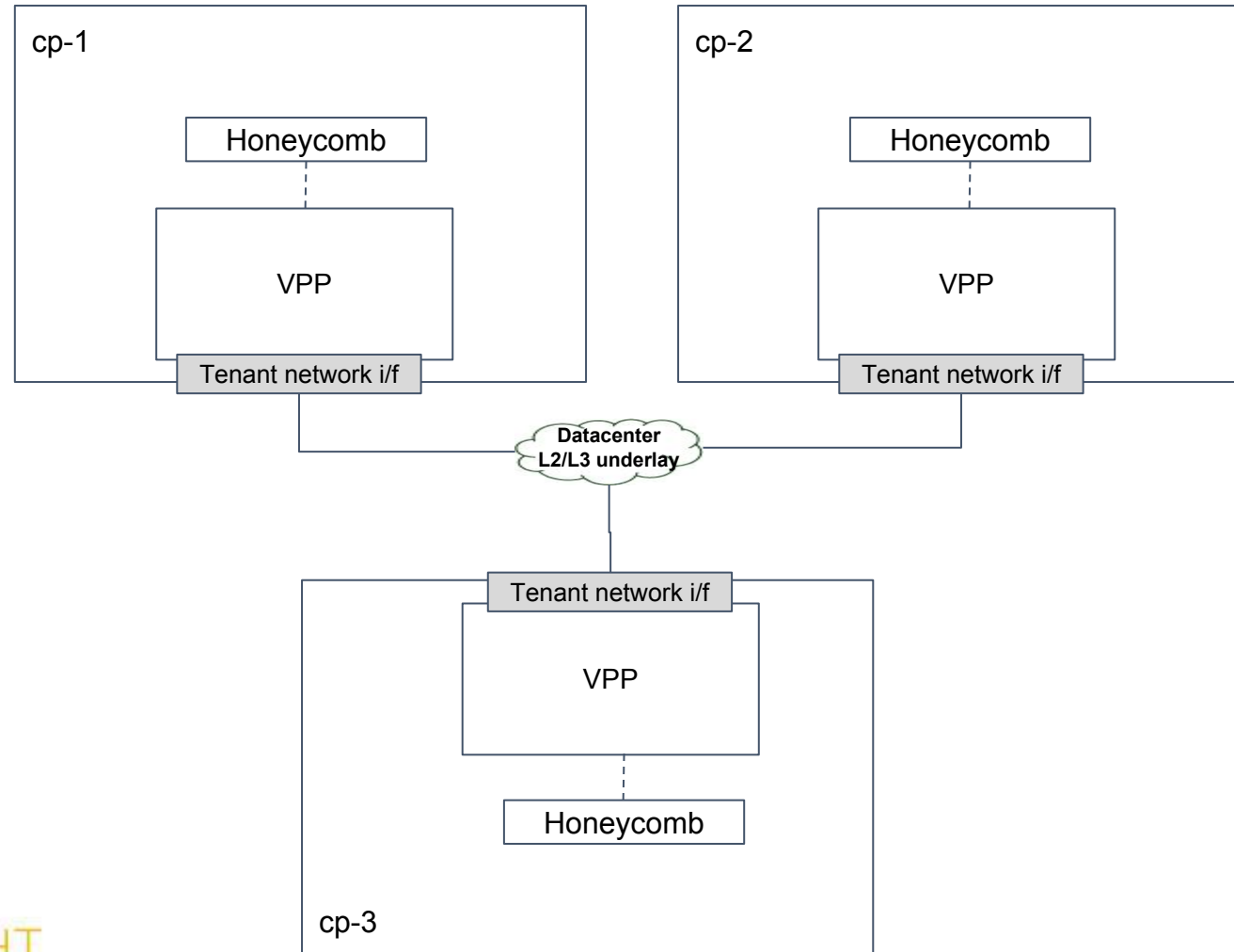


ODL GBP Architecture (2/2)

- GBP VPP Renderer - Interface Manager: Listens to VPP endpoints in the config data store and configures associated interfaces on VPP
- GBP VPP Renderer - Forwarding Manager: Manages the creation of bridge domains using VBD and assign interfaces to bridge domains
- GBP VPP Renderer Location Manager: Provides real location for endpoints (node & interface name)
- Virtual Bridge Domain Manager (VBD): Creates VxLAN tunnels/VLAN trunk links and bridge domains



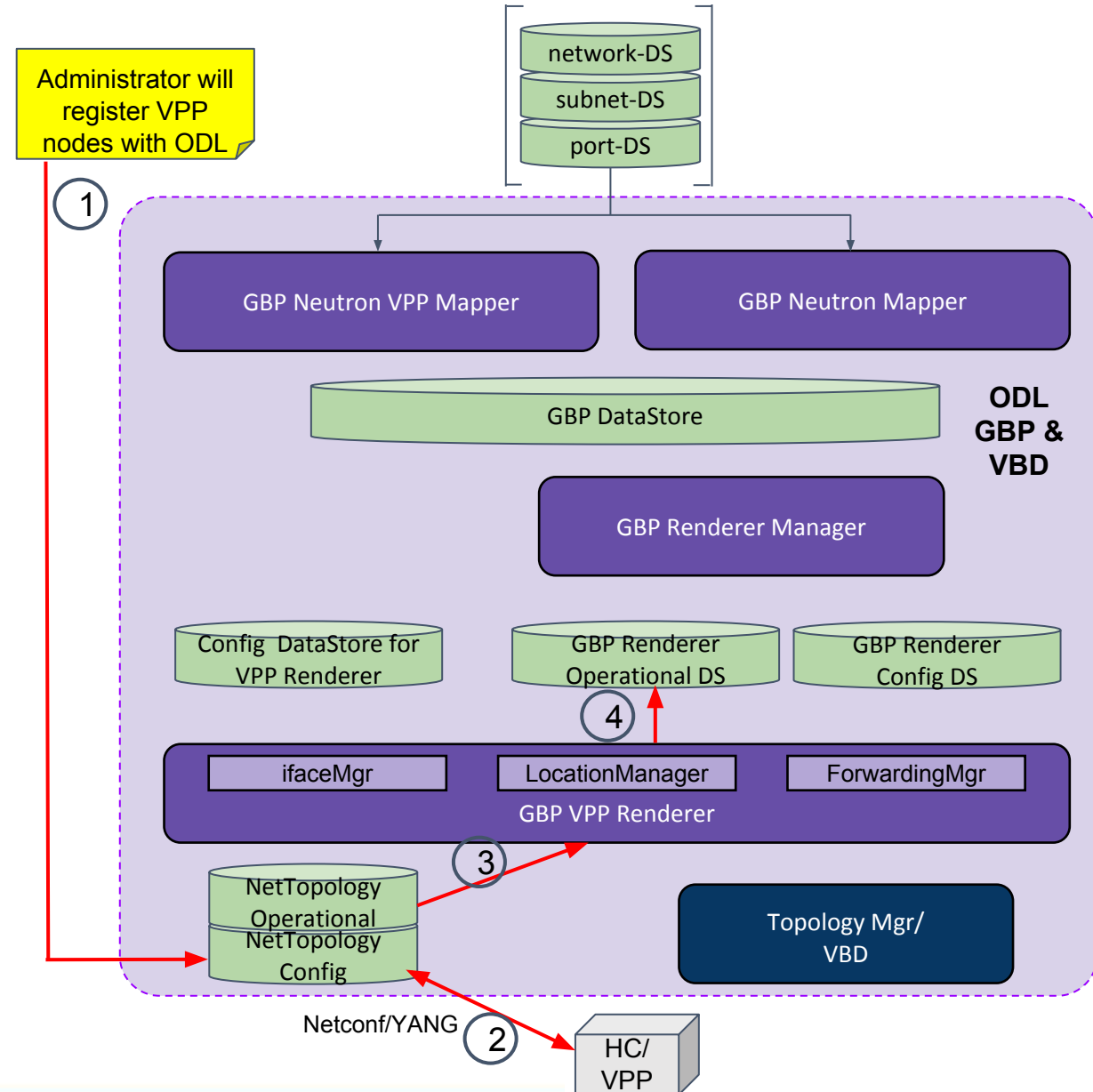
L2 Service Workflow



L2 Service Workflow

Register VPP nodes

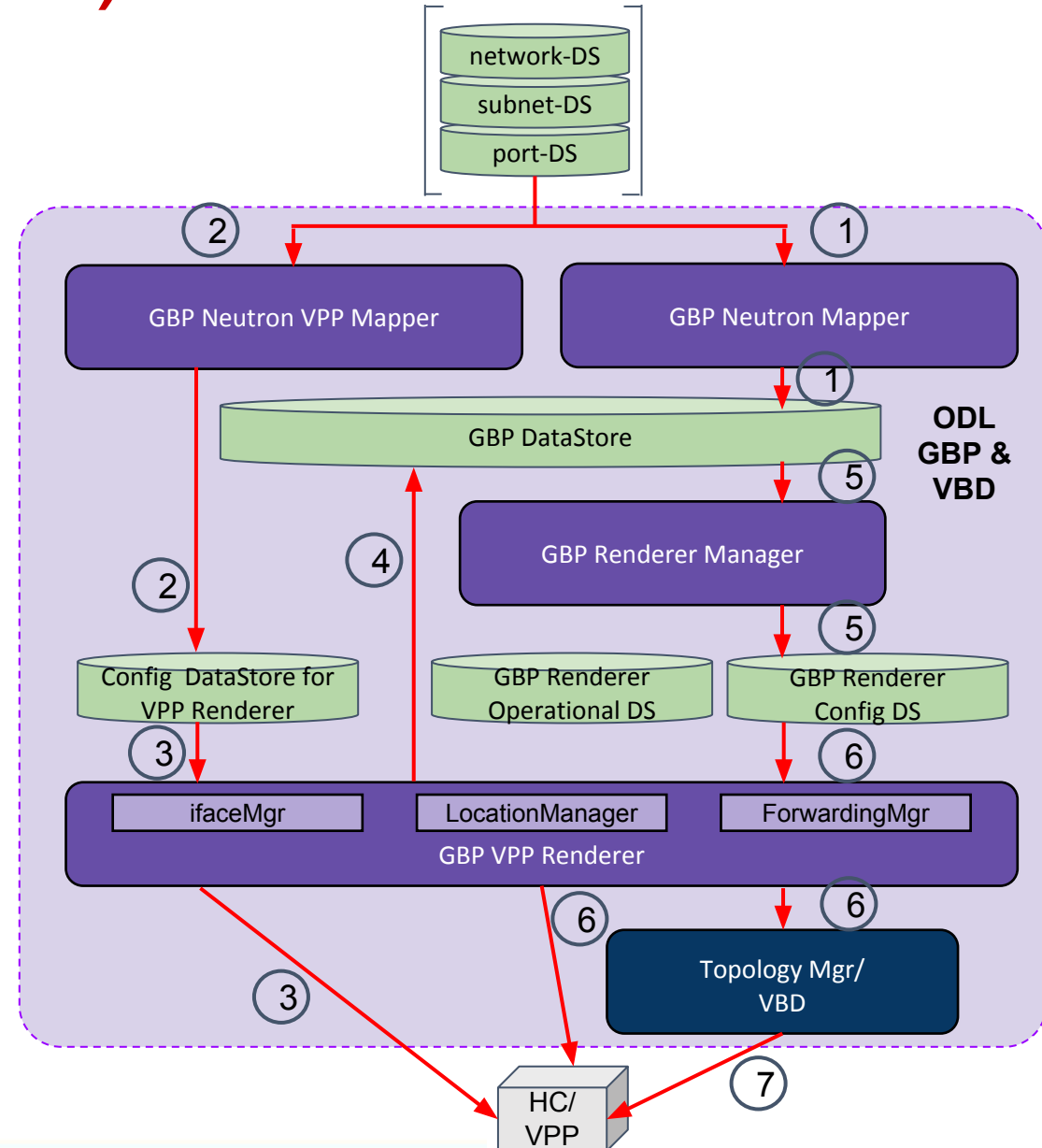
1. User/Administrator will register VPP nodes with ODL Netconf.
2. ODL will create a NETCONF session with HC
3. VPP Renderer will recognize VPP nodes from provided HC Netconf capabilities (RFC6421)
4. Successfully connected and recognized VPP nodes are stored in GBP Renderer Operational DS



L2 Service Workflow (1/2)

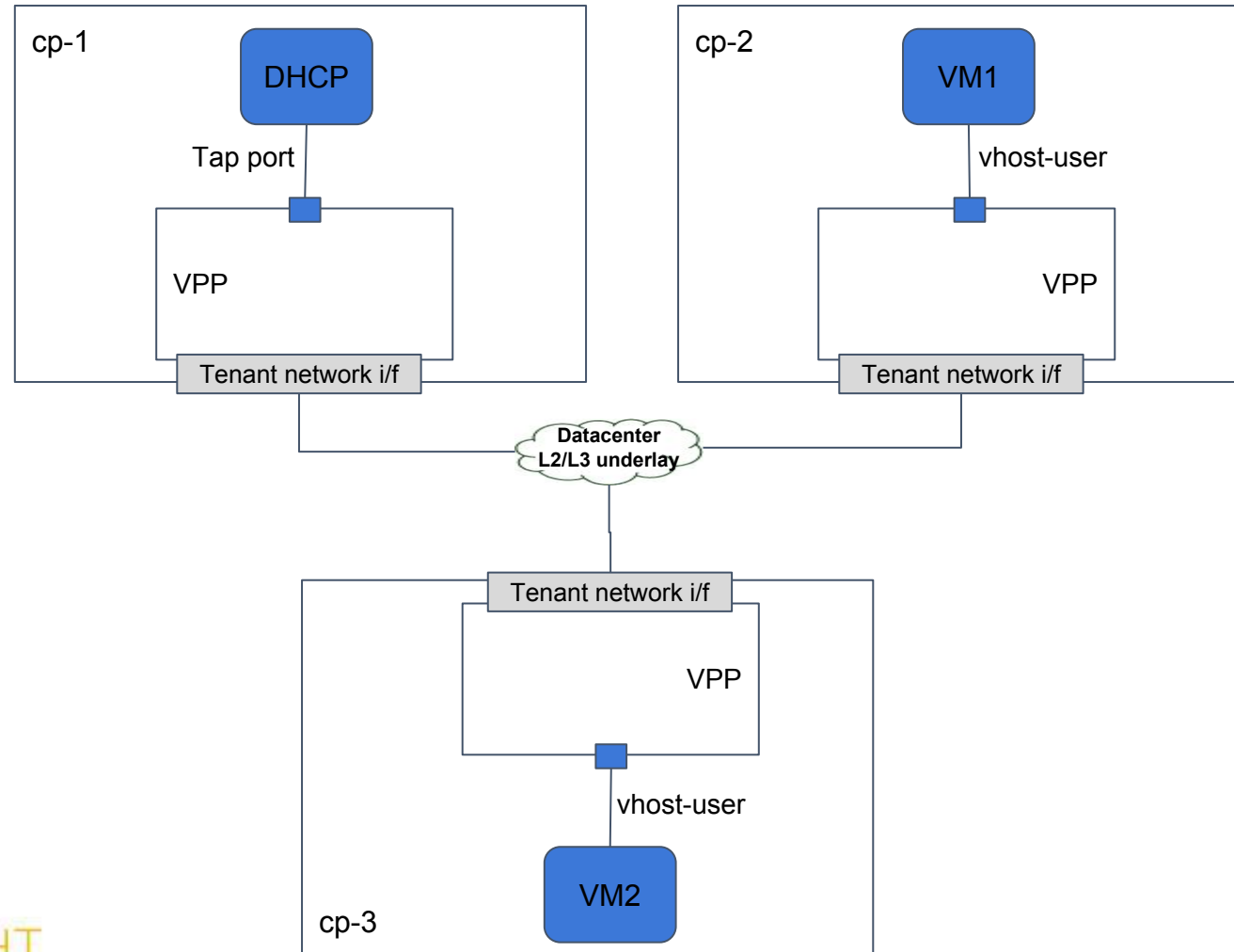
Configure Networks, Subnets and VMs

1. GBP Neutron Mapper receives neutron Data-Change-Notifications (DCN) and translate them into GBP entities
2. GBP Neutron VPP Mapper also listens to neutron DCNs and writes VPP specific data into VPP config DS.
 - a. VppEndpoints ~ Port(id, location, type(vhost,tap,loop)), Bridge-Domains ~ Networks(id, type(vxlan, vlan))
3. GBP ifaceMgr inside VPP renderer listens on created endpoints and creates these interfaces in HC/VPP
4. VPP renderer advertises location of the VPP interfaces to GBP Datastore.



L2 Service Workflow

Configure Networks, Subnets and VMs



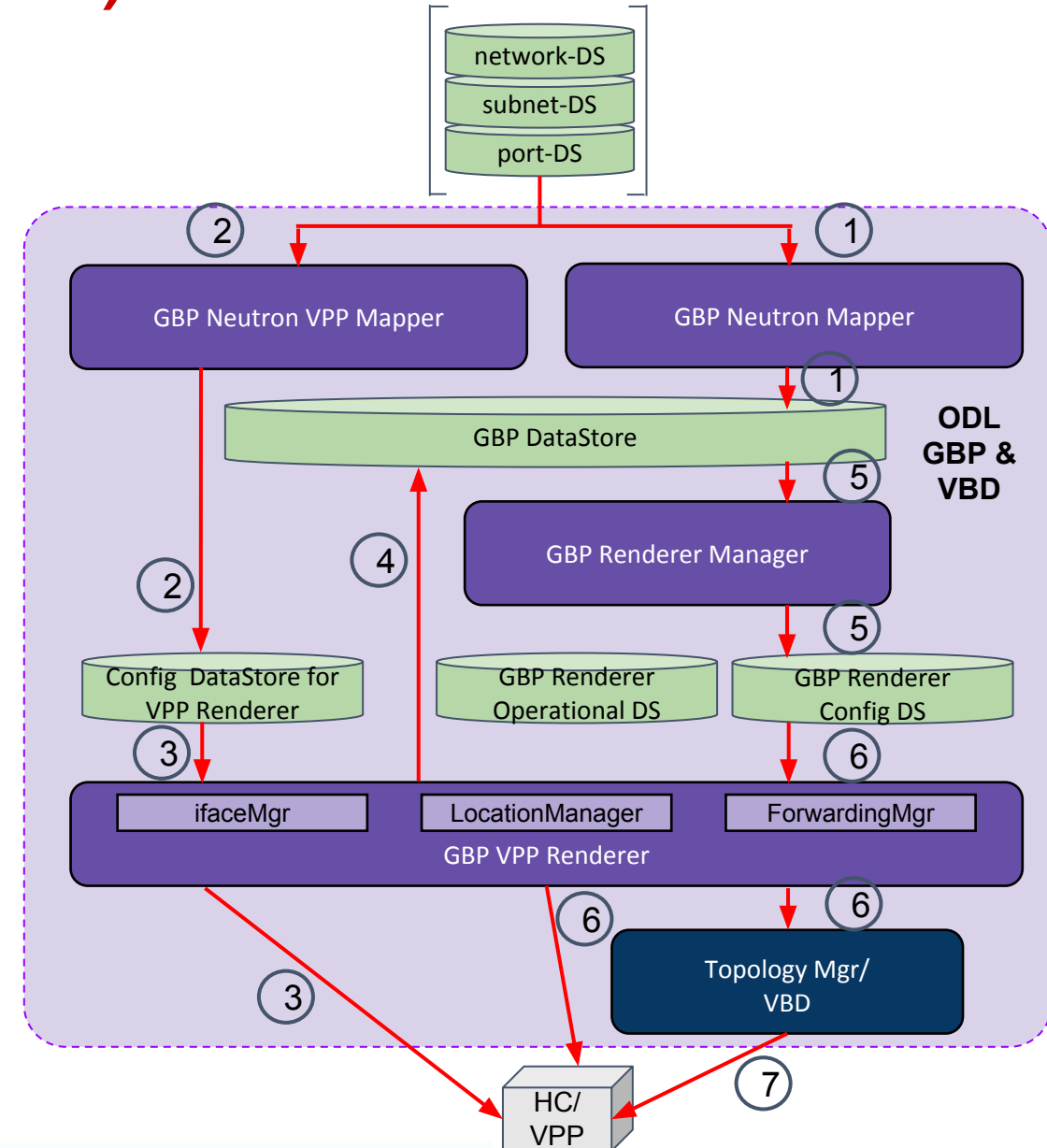
L2 Service Workflow (2/2)

Configure Networks, Subnets and VMs

5. Now GBP-Renderer Manager has all information to resolve policy and to write configuration to GBP Renderer Config DS

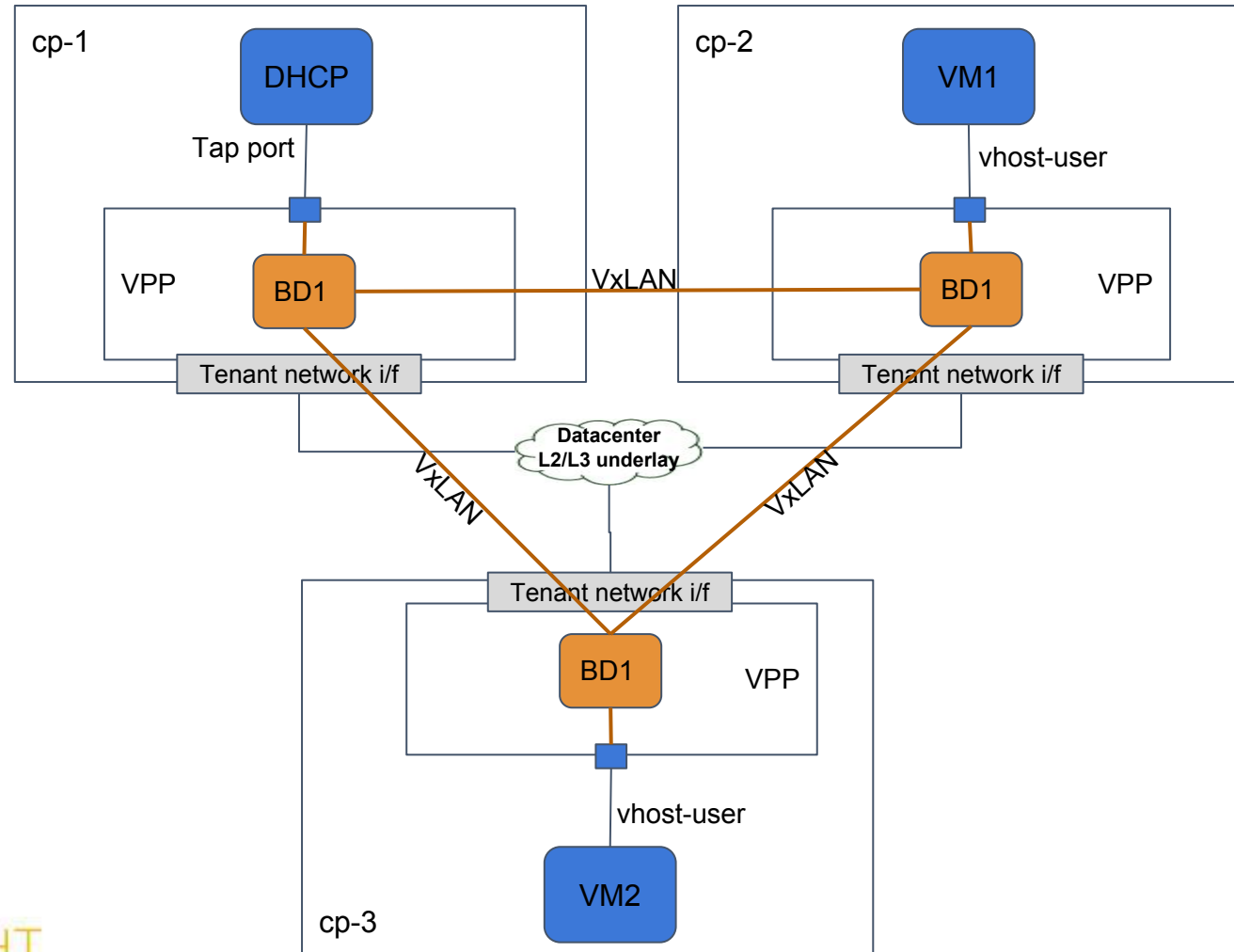
6. Forwarding Manager in VPP renderer will instruct VBD to create required bridge domains. Created ports (that correspond to VPP endpoints) are attached to bridge domains based on resolved configuration.

7. After BDs are created, VBD creates required VXLAN tunnels between bridge domains on VPP.



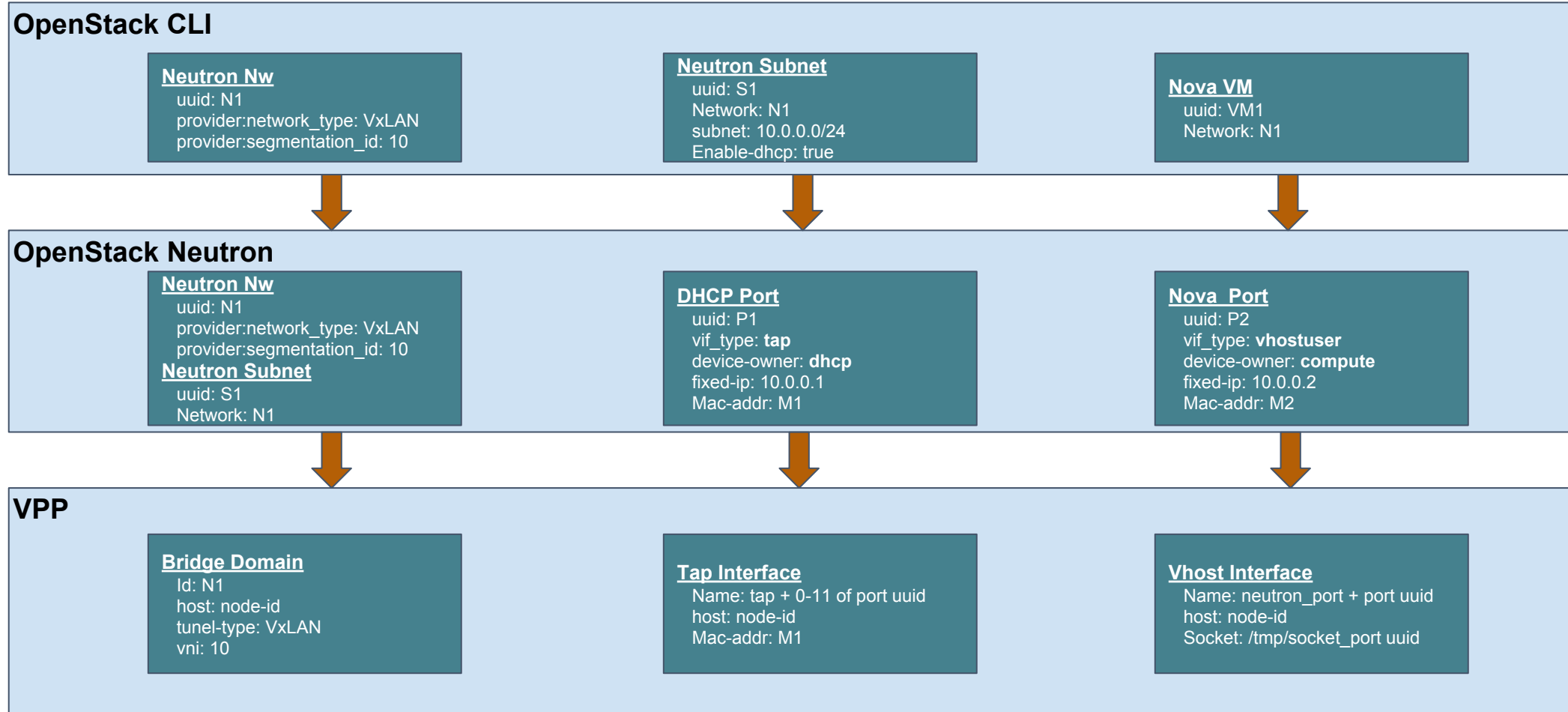
L2 Service Workflow

Configure Networks, Subnets and VMs



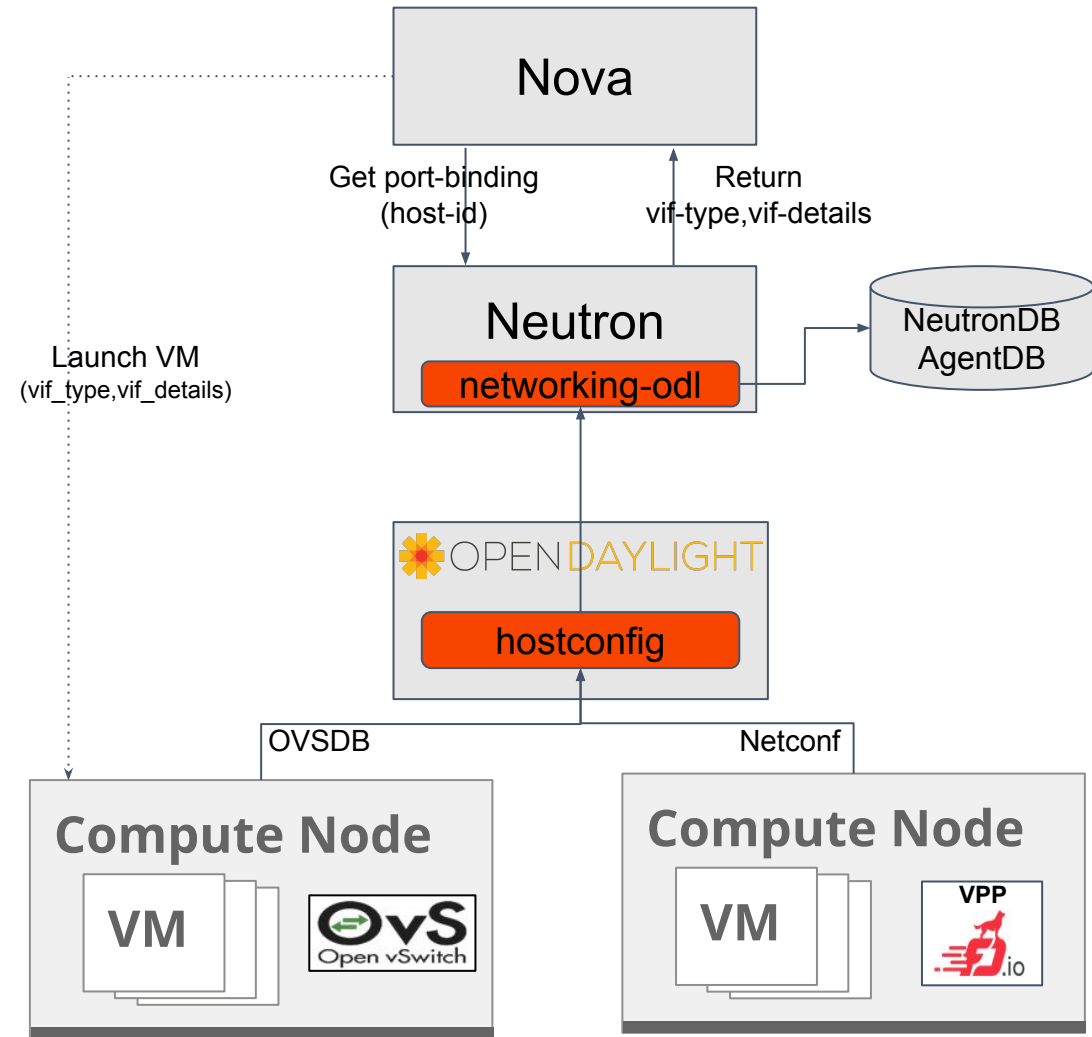
L2 Service Workflow

Neutron network and port instances in VPP



Generalized port binding

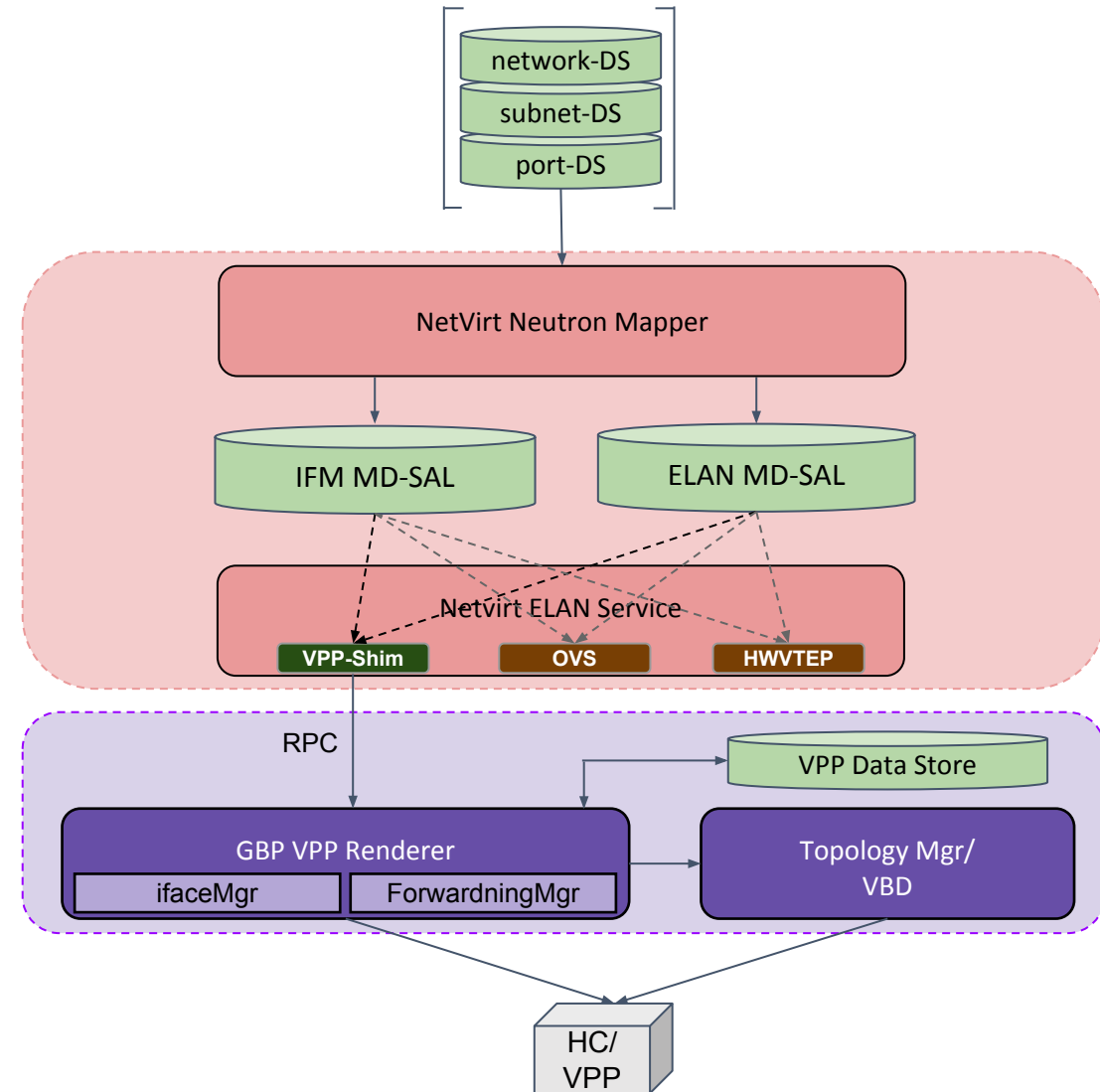
- Hostconfig: ODL reports host configuration to networking-odl which stores in neutron agent-db
- Agnostic to switch technology: OVS, OVS-DPDK, VPP...
- Port-binding: Correlate Neutron Port to compute node



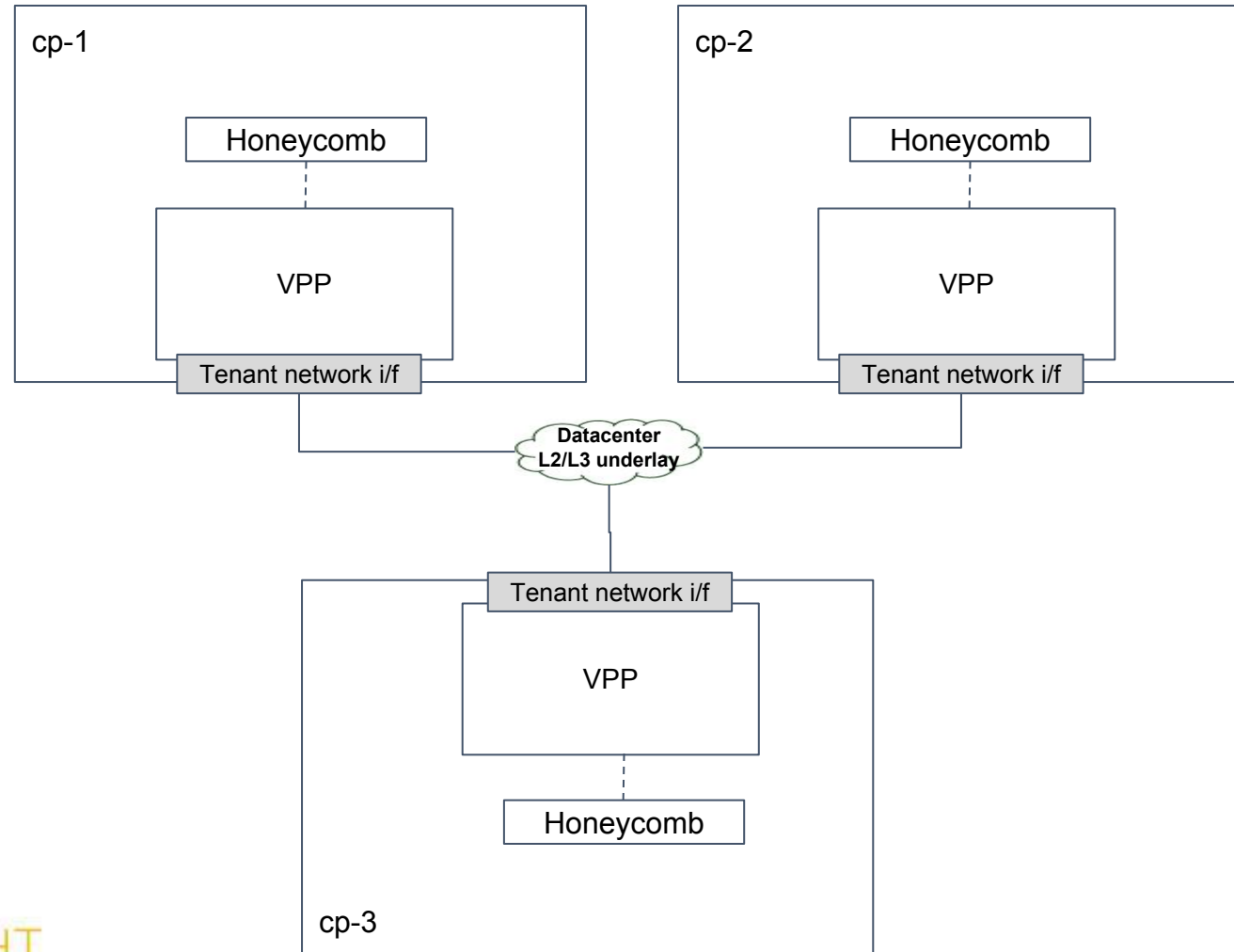
```
hostconfig:
  • host-id: overcloud-controller-0.opnfv1f.org
  • host-type: "ODL L2"
  • config:
    • allowed_network_types: [local, vlan, vxlan, gre],
    • supported_vnic_types:
      • vnic_type:normal,
      • vif_type:vhostuser,
      • vif_details:{
        • support_vhost_user:True,
        • port_prefix:socket_,
        • vhostuser_socket_dir:/tmp,
        • vhostuser_mode:server,
        • vhostuser_socket:/tmp/socket_${PORT_ID}
```

Integrated Solution - ELAN/L2 Service

- Realization of ELAN/L2 functionality in converged architecture
- Introduced VPP-Shim layer in NetVirt ELAN service component
 - Listens to same ELAN service data models and translate them in to VPP renderer RPCs
- Genius Interface data model enhanced to contain neutron port binding information:
 - vif_type
 - vhostuser_mode
 - vhostuser_socket



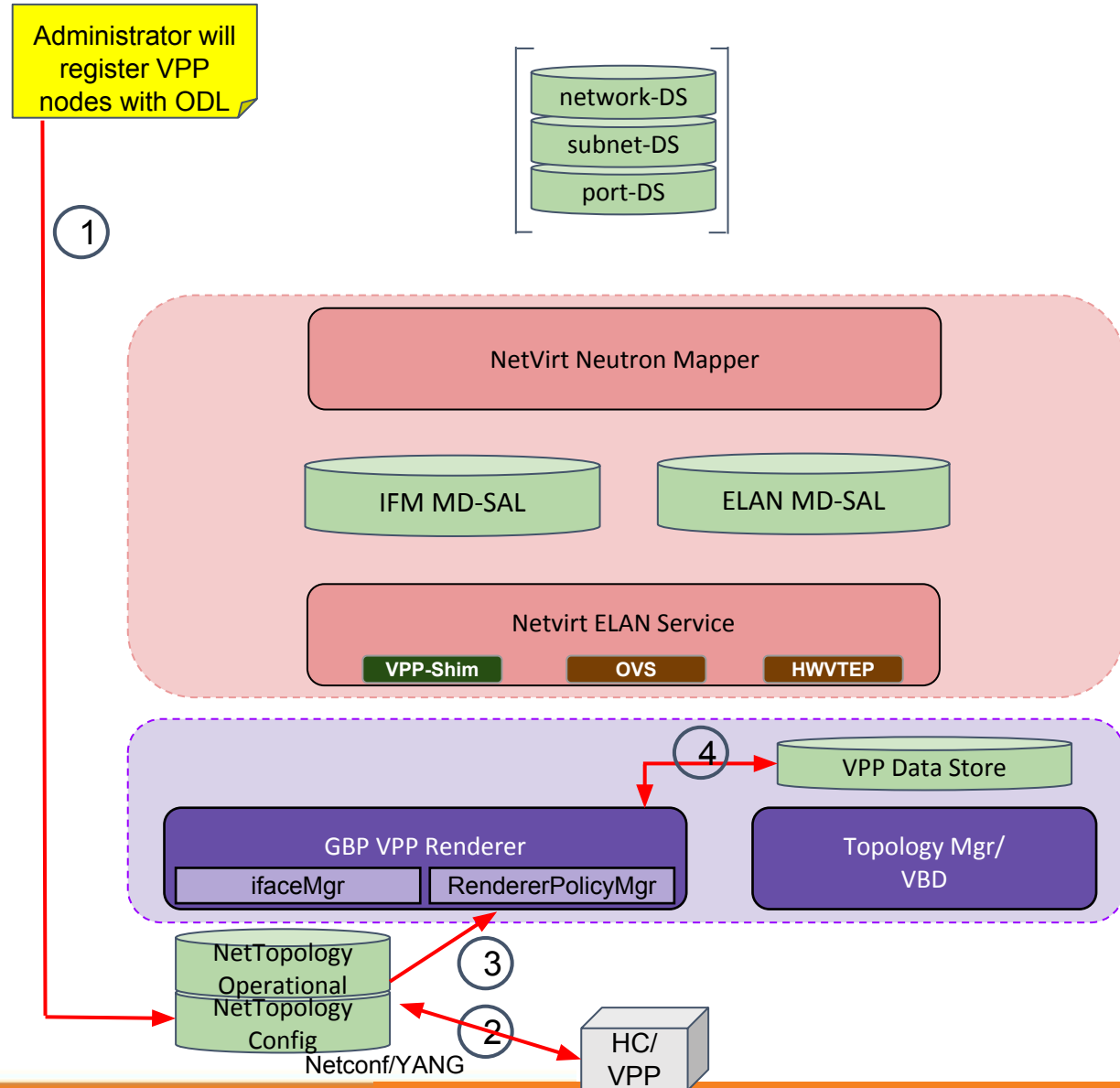
ELAN Service Workflow



ELAN Service Workflow

Register VPP nodes

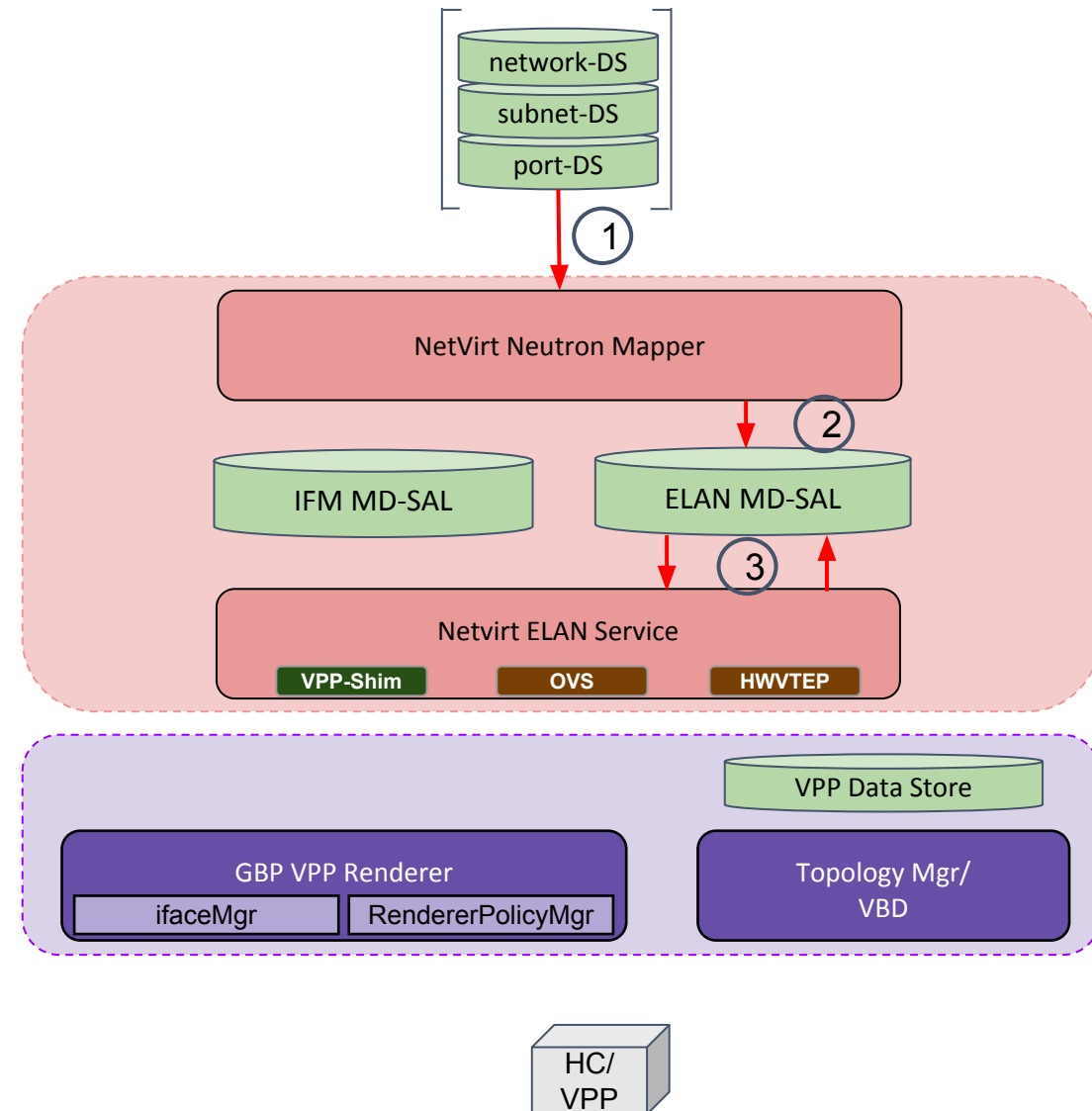
1. User/Administrator will register VPP nodes with ODL Netconf.
2. ODL will create a NETCONF session with HC
3. VPP Renderer will recognize VPP nodes from provided HC Netconf capabilities (RFC6421)
4. Successfully connected and recognized VPP nodes are stored in VPP Renderer Operational DS. VPP nodes are now available to VPP renderer in GBP Renderer Operational DS



ELAN Service Workflow

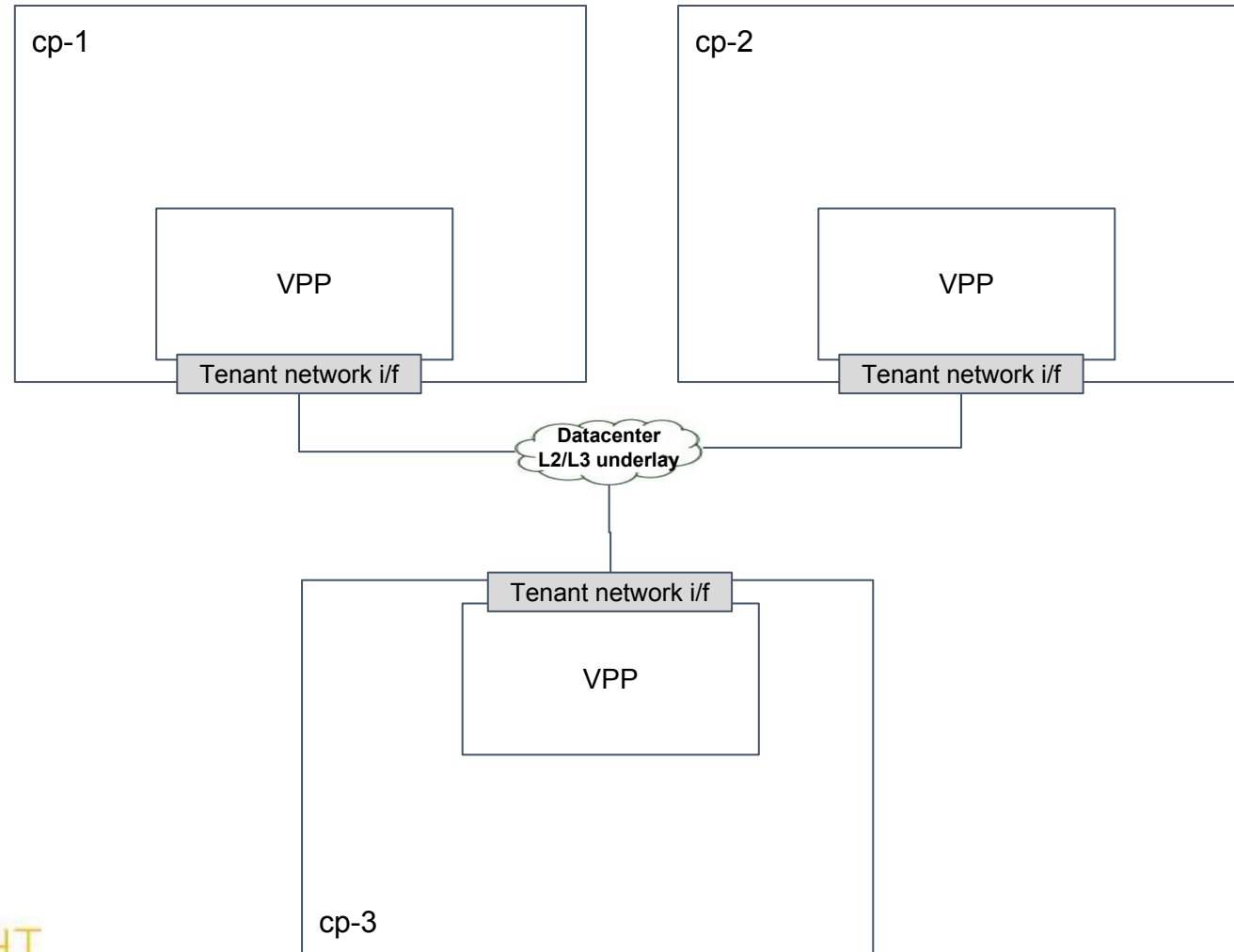
Neutron Network Create

1. NetVirt Neutron Mapper listens on DCNs from Neutron Northbound for network creation
2. Neutron Mapper create a new ELANInstance in ELAN data store
3. ELAN service is activated, it allocates an ELAN tag for this instance
 - a. The ELAN tag is a dataplane identifier for the ELAN instance.



ELAN Service Workflow

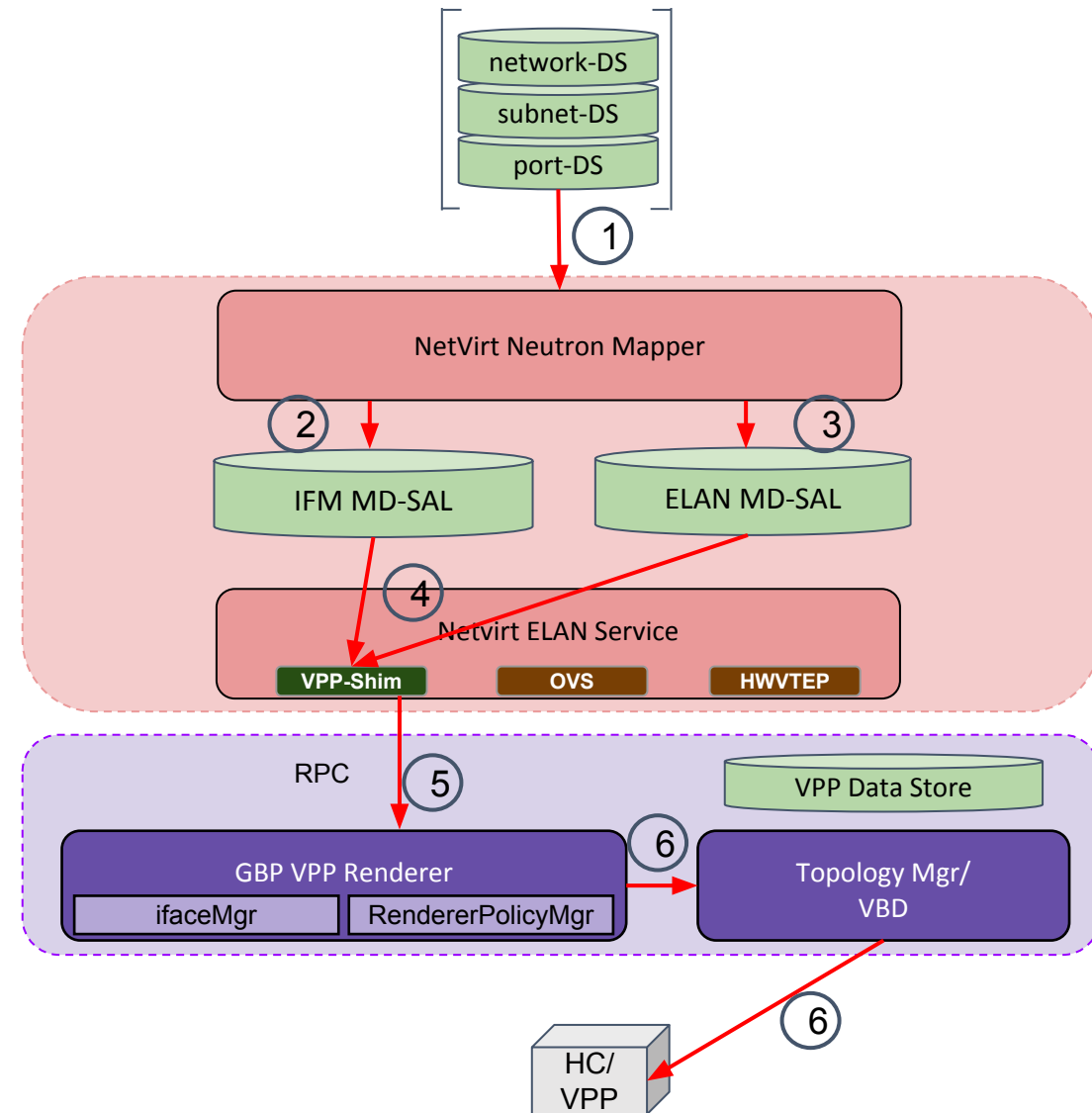
Neutron Network Create - No action in data plane



ELAN Service Workflow

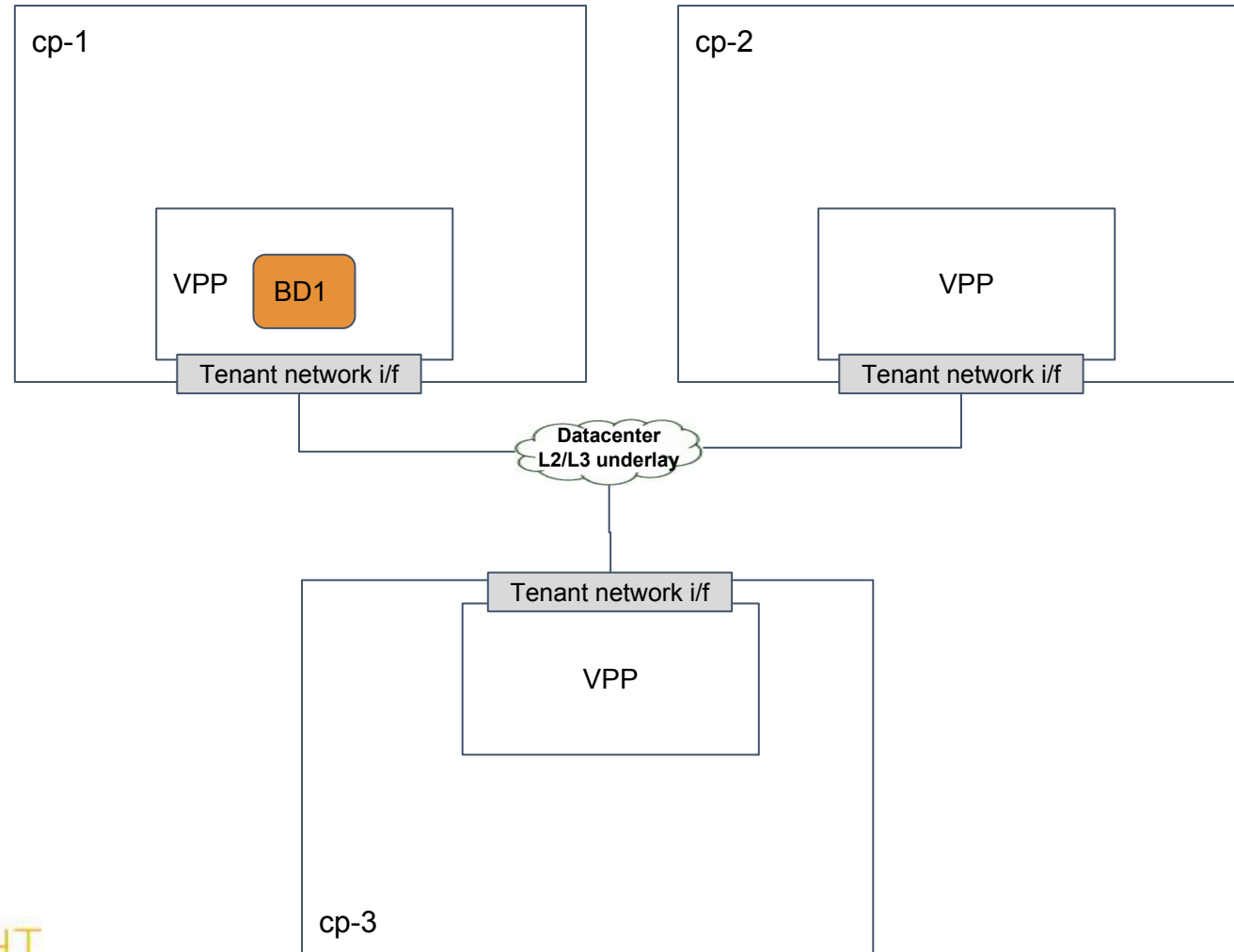
Neutron Port Create/Update (1/2)

1. NetVirt Neutron Mapper listens on DCNs from Neutron Northbound for port creation
2. Neutron Mapper creates Interface object
 - a. Populates Interface data model with port-binding info received from neutron
3. Neutron Mapper creates ELANInterface object
4. VPP-Shim receives a DCN for Interface and ELANInterface creation/Update
 - a. Processes the DCNs only if the associated host is of **Netconf** device type and interface is of **vhostuser** type
5. VPP-Shim invokes createBridgeDomain RPC calls
6. VPP Renderer creates Bridge Domains in VPP nodes using VBD



ELAN Service Workflow

Neutron Port Create/Update (1/2)



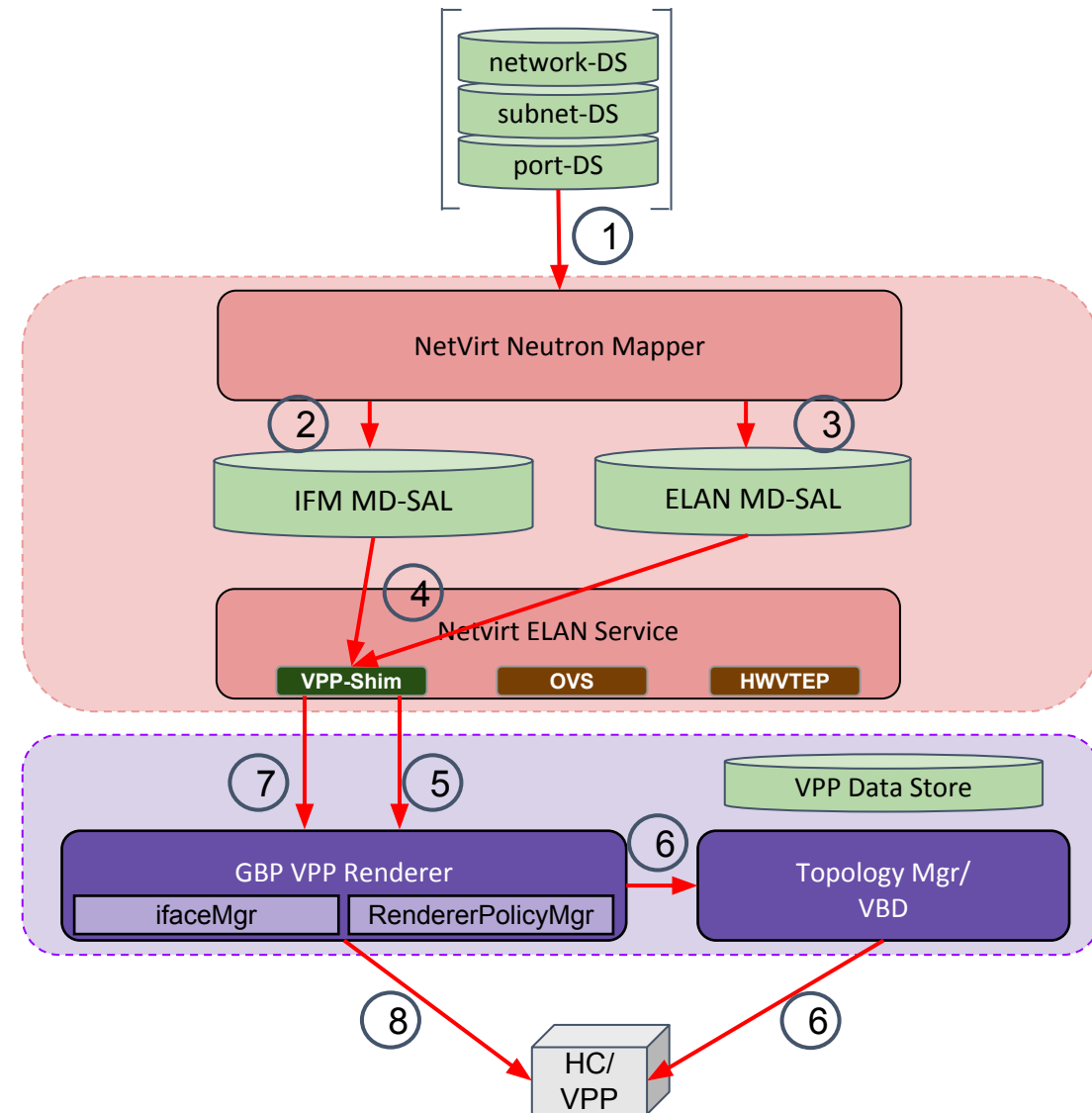
ELAN Service Workflow

Neutron Port Create/Update (2/2)

7. After BDs are created, VPP-Shim invokes `createVppInterfaceOnNode` and `addVppInterfaceToBridgeDomain` RPC calls

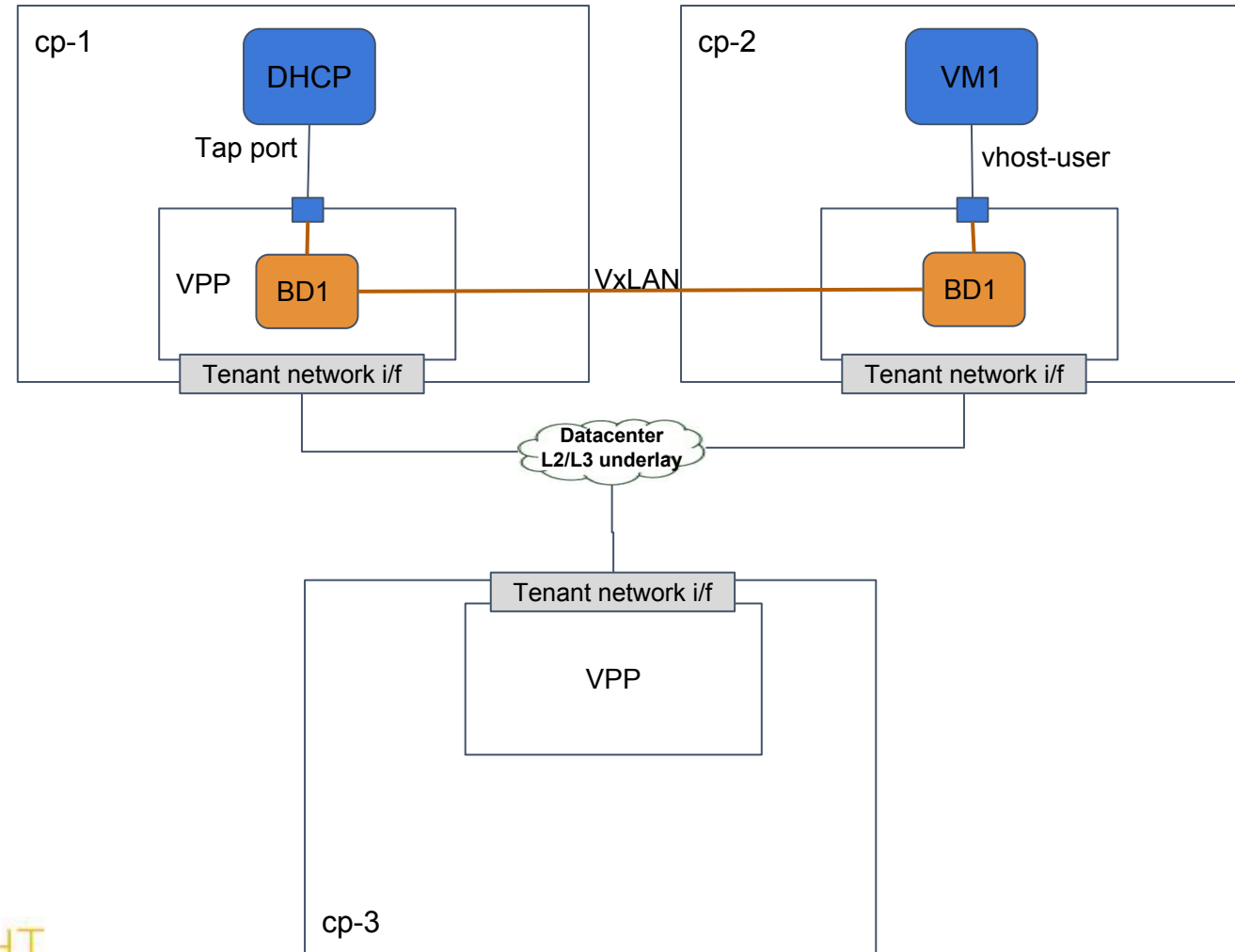
8. `ifaceMgr` in VPP Renderer will create tap or vhostuser interfaces in VPP nodes and add them to the specified bridge domain

VPP-Shim invokes `cloneBridgeDomain` RPC call if an existing bridge domain need to be created on other nodes before creating interfaces on those nodes. GBP VPP Renderer triggers creates VxLAN tunnel interfaces for each bridge domain as needed.



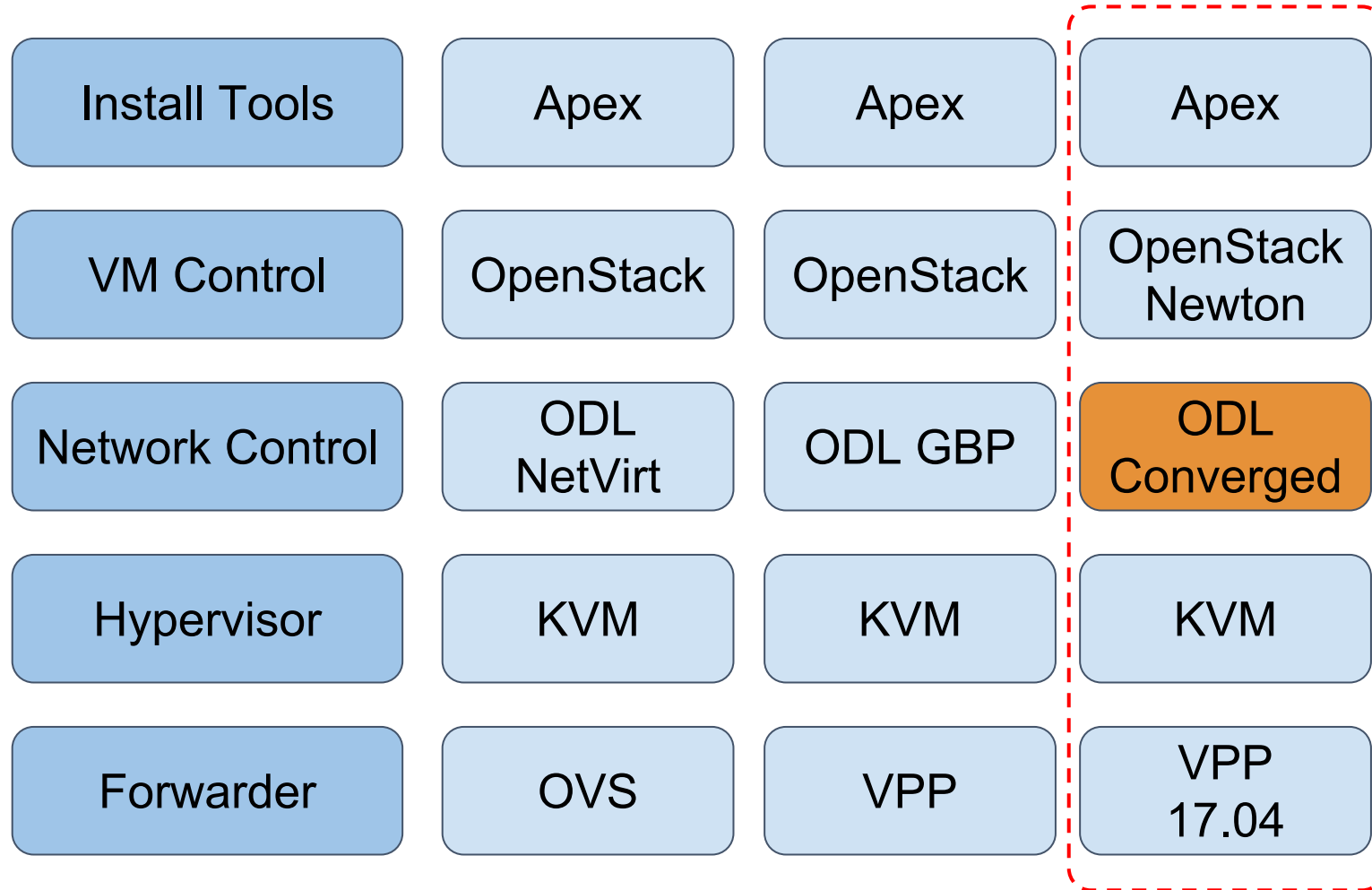
ELAN Service Workflow

Neutron Port Create/Update (2/2)



OPNFV “Full Stack Integration”

- Automated Installation: Integration into OPNFV APEX installer
 - APEX is based on OpenStack Triple-O
- New Scenario (planned for OPNFV Danube 3.0)
 - OpenStack (Newton) - ODL Converged - FD.io (17.04)

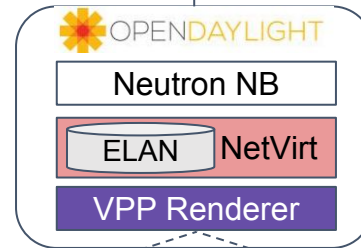
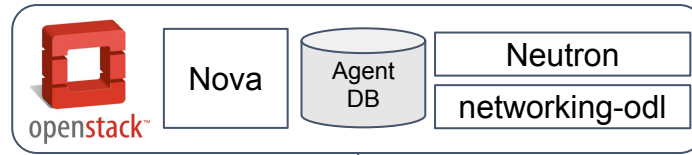


References

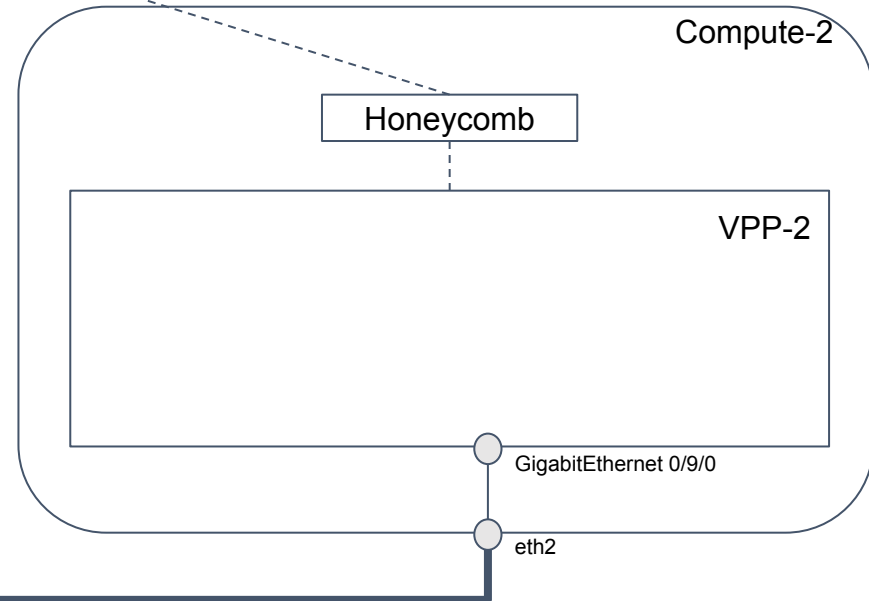
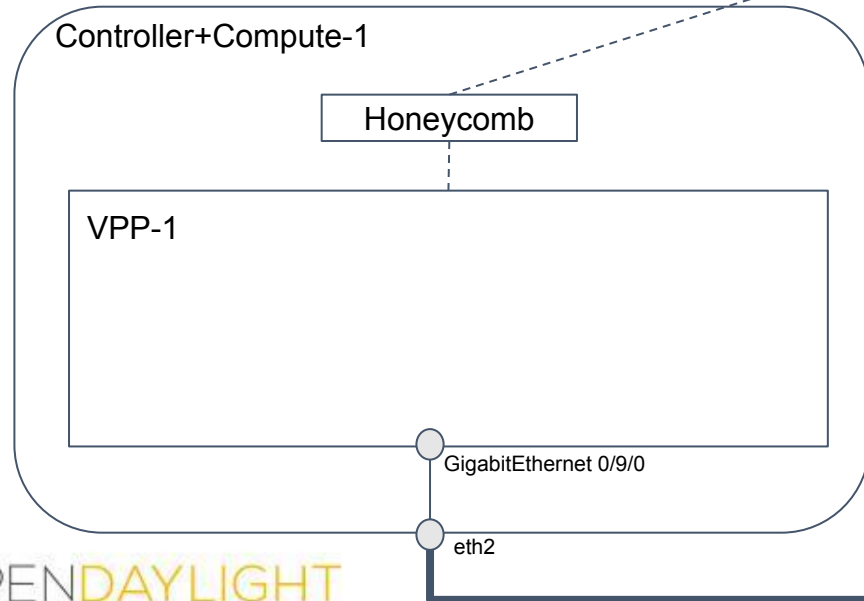
- Code for ELAN Service with ODL Integrated Solution
 - NetVirt Enhancements: <https://git.opendaylight.org/gerrit/#/c/50259/>
 - Genius Enhancements: <https://git.opendaylight.org/gerrit/#/c/53632/>
- APEX Scenario with ODL Integrated Solution
 - <https://gerrit.opnfv.org/gerrit/33433>
- GBP interface for remote calls on VPP
 - <https://git.opendaylight.org/gerrit/#/c/48962/>

Demo

2 node setup - 1 x Controller, 2 x Compute



Netconf



Register VPP nodes

OpenDaylight reads host-config information

networking-odl driver pulls and updates Neutron agent database with host-config

Pre-reqs

Get an image

```
[root@overcloud-controller-0 ~]# wget -O /tmp/cirros-0.3.4-x86_64-disk.img http://download.cirros-cloud.net/0.3.4/cirros-0.3.4-x86_64-disk.img
```

Create an image in Glance

```
[root@overcloud-controller-0 ~]# glance image-create --name "cirros-0.3.4-x86_64" --file /tmp/cirros-0.3.4-x86_64-disk.img --disk-format qcow2 --container-format bare --visibility public --progress
```

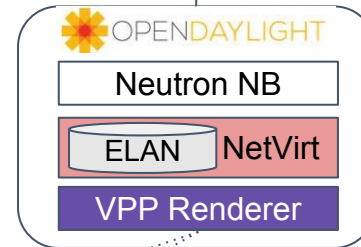
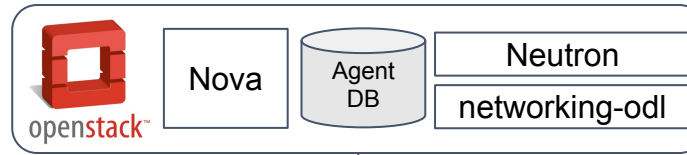
Create a flavor with hugepages support

```
[root@overcloud-controller-0 ~]# nova flavor-create --is-public true opnfv 42 768 1 1  
[root@overcloud-controller-0 ~]# nova flavor-key opnfv set hw:mem_page_size=large
```

Make glance image support hugepages

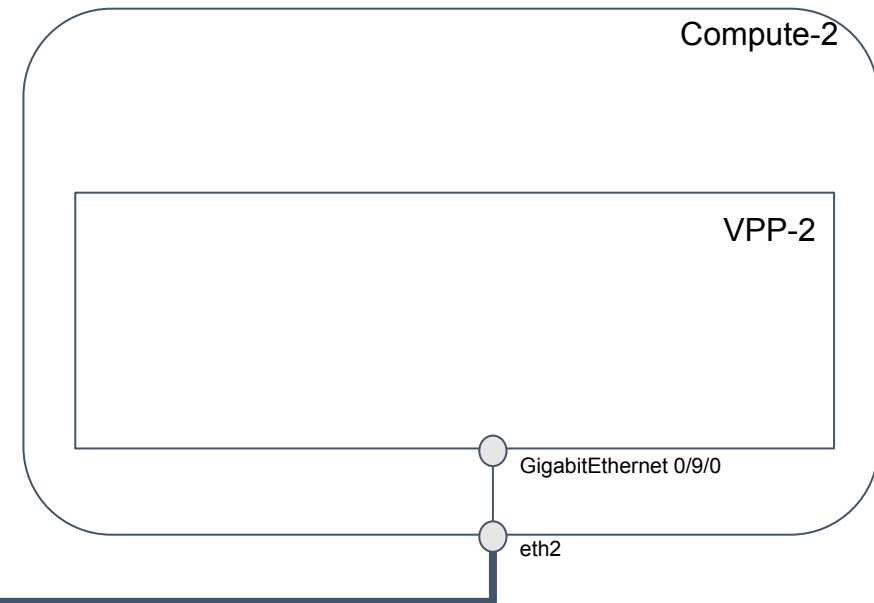
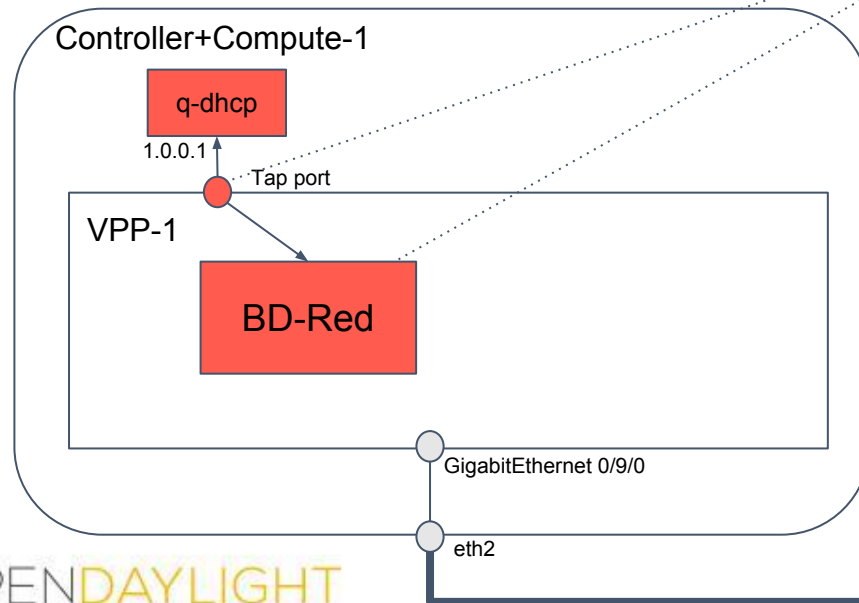
```
[root@overcloud-controller-0 ~]# glance image-update --property hw_mem_page_size=large ad74564d-fd22-414b-9fa3-619e87f781a9
```

Create Neutron Network and Subnet

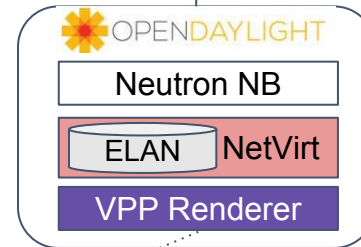
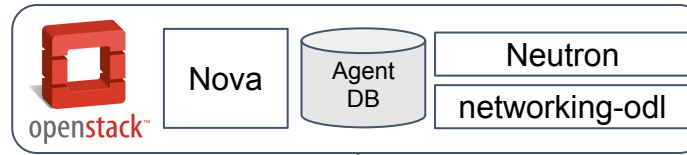


neutron net-create red

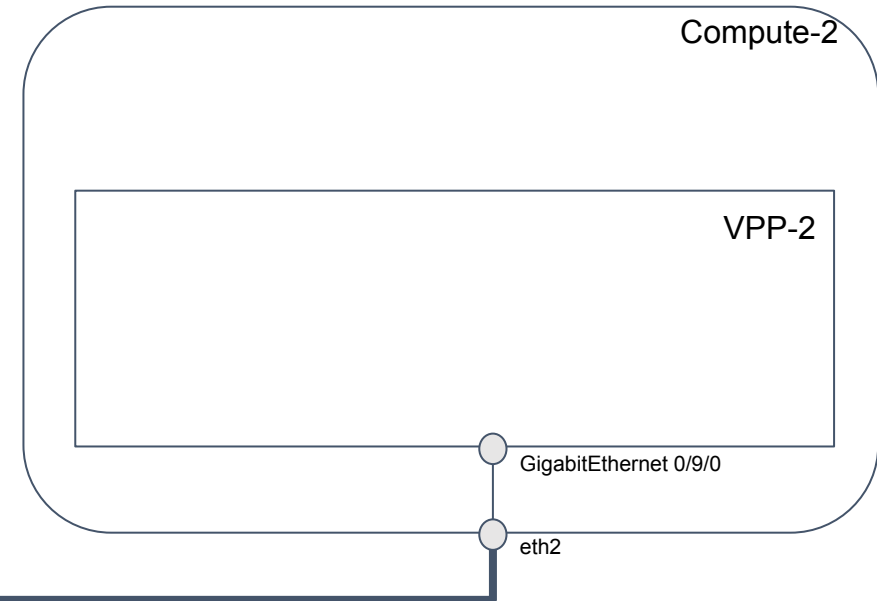
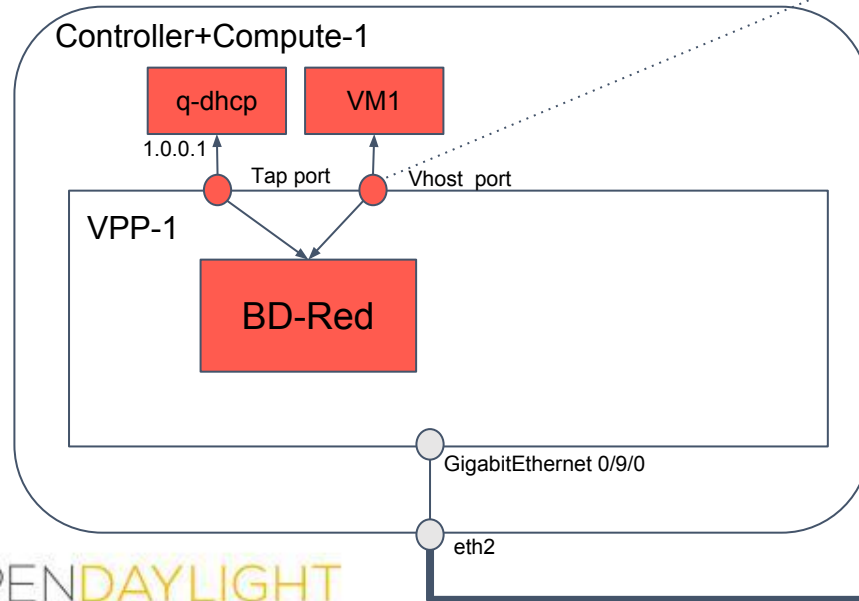
neutron subnet-create red 1.0.0.0/24



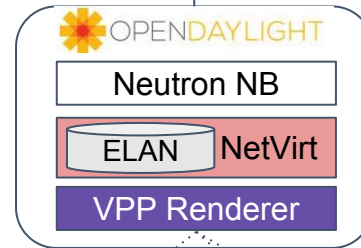
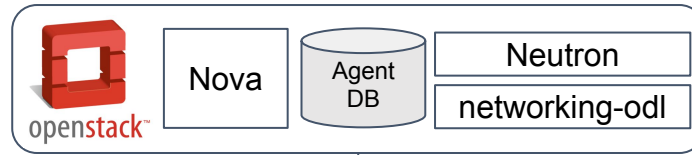
Create VM1



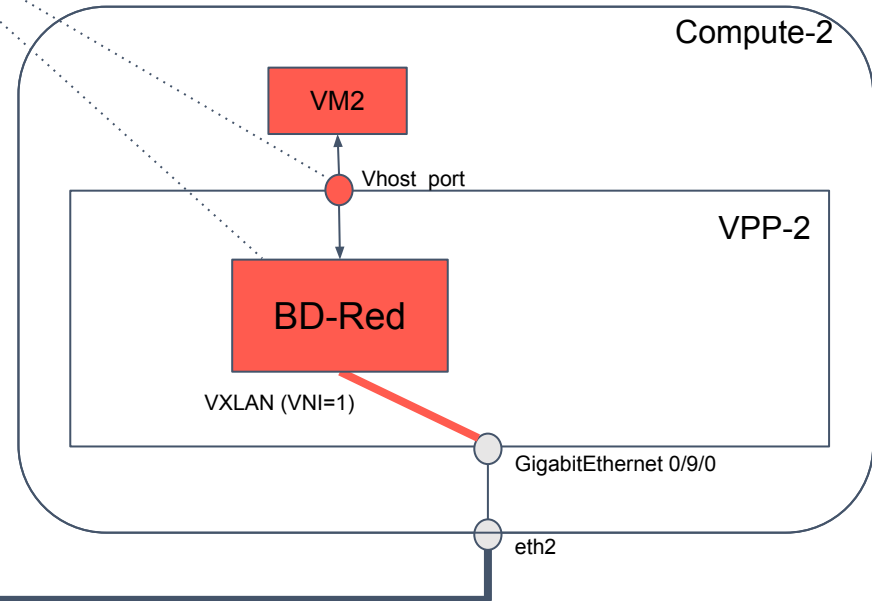
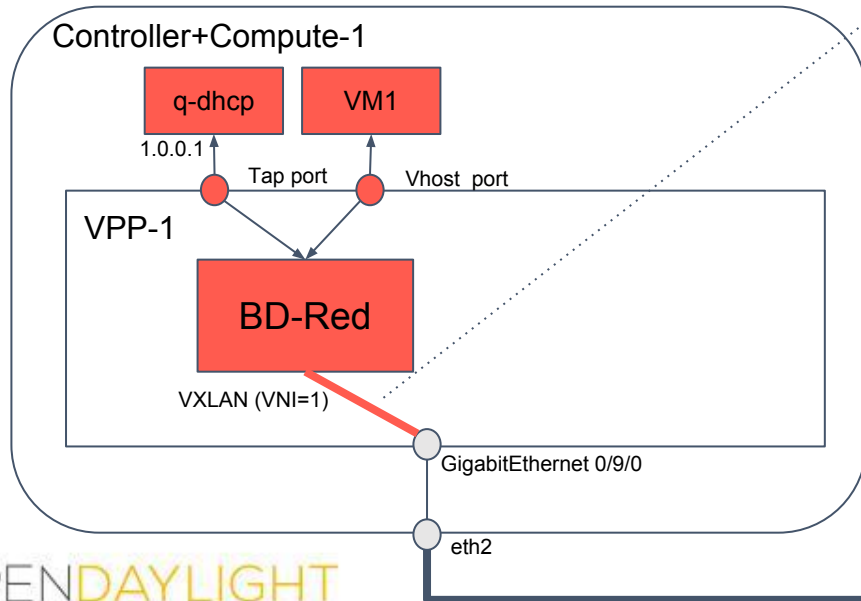
```
nova boot vm1 --nic net-id=red
```



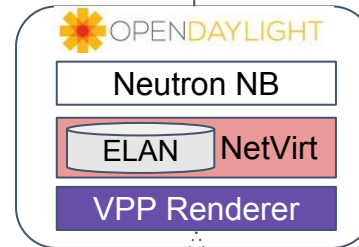
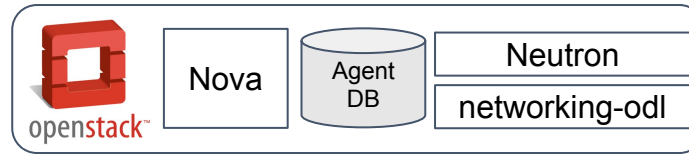
Create VM2



`nova boot vm2 --nic net-id=red`



Create second network and VMs in it

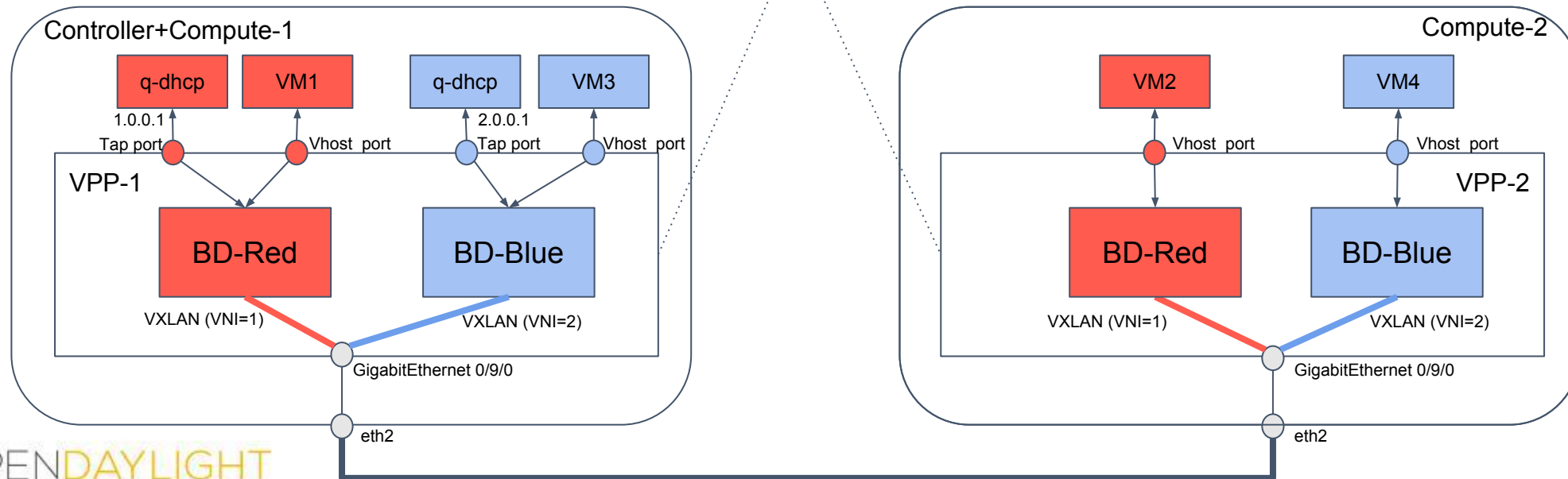


```
neutron net-create blue
```

```
neutron subnet-create blue 2.0.0.0/24
```

```
nova boot vm3 --nic net-id=blue
```

```
nova boot vm4 --nic net-id=blue
```



Near term roadmap

- Realization of L3VPN functionality in a converged architecture
- Realization of Service Function Chaining (SFC) functionality in a converged architecture
- Other ELAN/L2 Service features
 - Security Groups
 - VLAN transparency/trunking
 - Q-in-Q
 - Connectivity to legacy BM/SR-IOV appliances
- Hybrid deployments (different types of forwarders)



Thank You

