Open vSwitch

December 10-11, 2019 | Westford, MA

OVN for Network functions with K8s

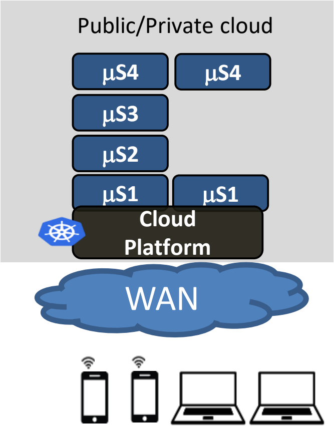
Srinivasa Addepalli, Ritu Sood - Intel

Agenda

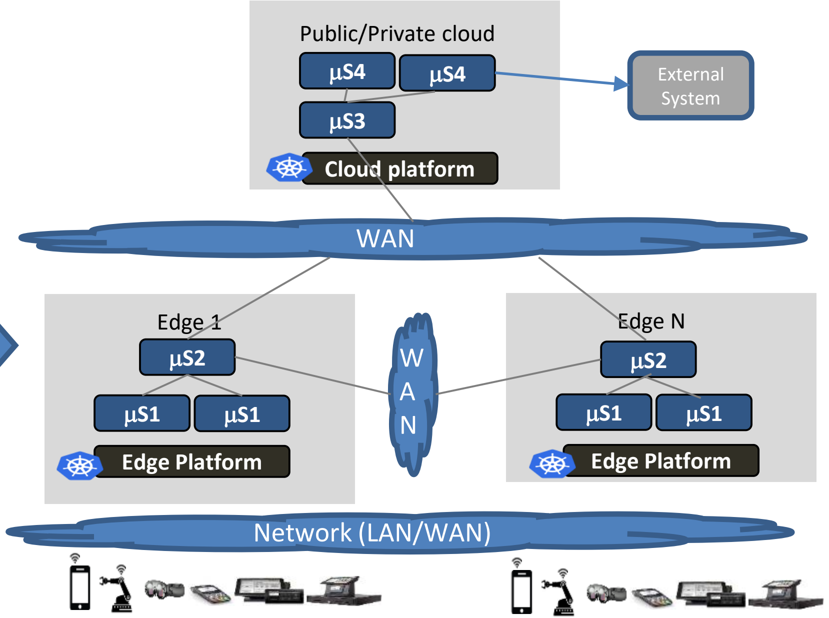
- Why network functions in Edge & K8s clusters?
- Edge-computing scenario to describe the K8s networking requirements
- Networking requirements
- OVN-for-K8s-NFV architecture blocks
- OVN-for-K8s-NFV details
- Current Status and roadmap
- Q&A

Application Transformation

(AR/VR apps, Gaming, Analytics and Even traditional applications due to sovereignty and context)



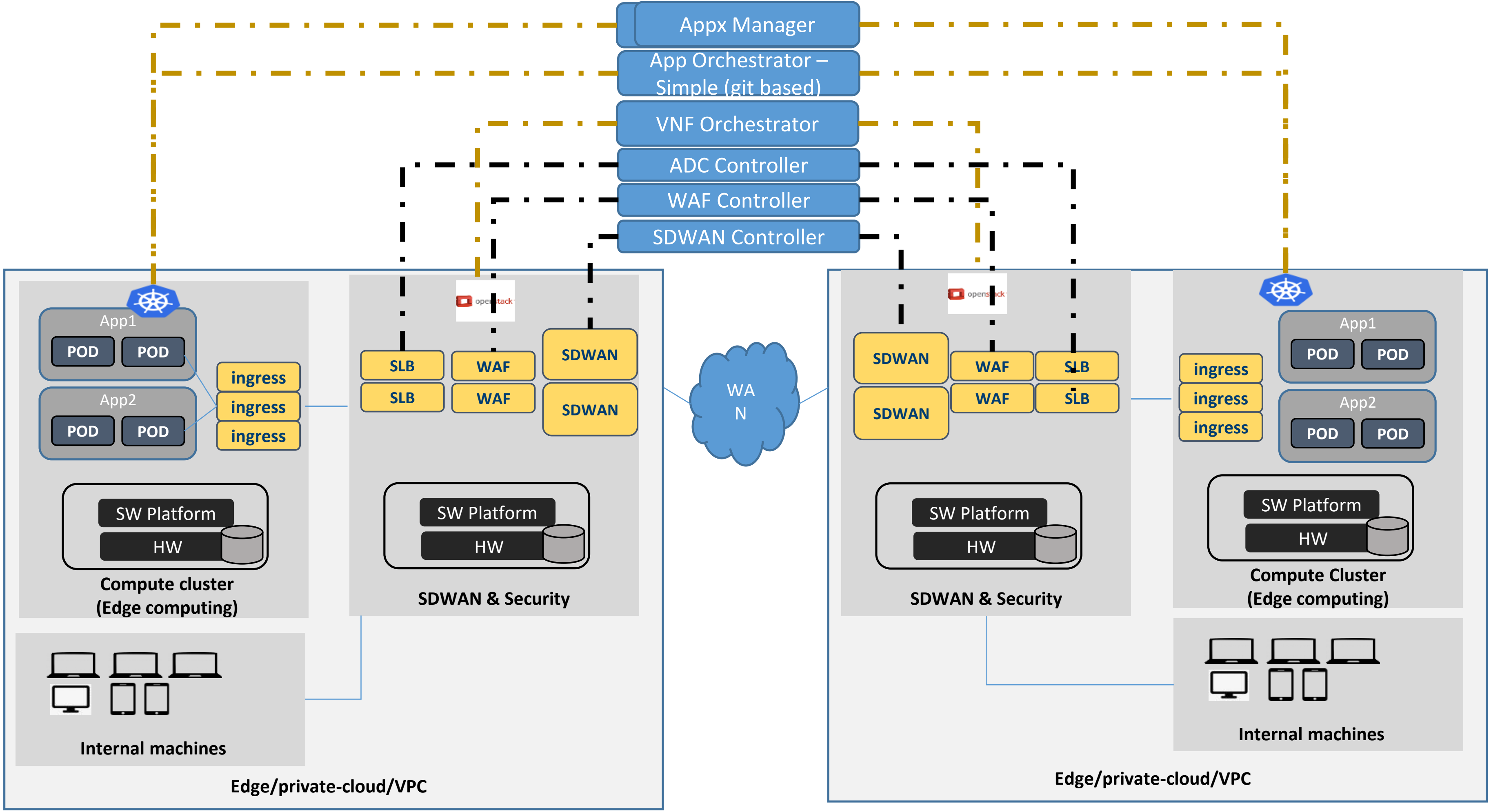
- Proximity
- Data sovereignty
- Economics
- Context



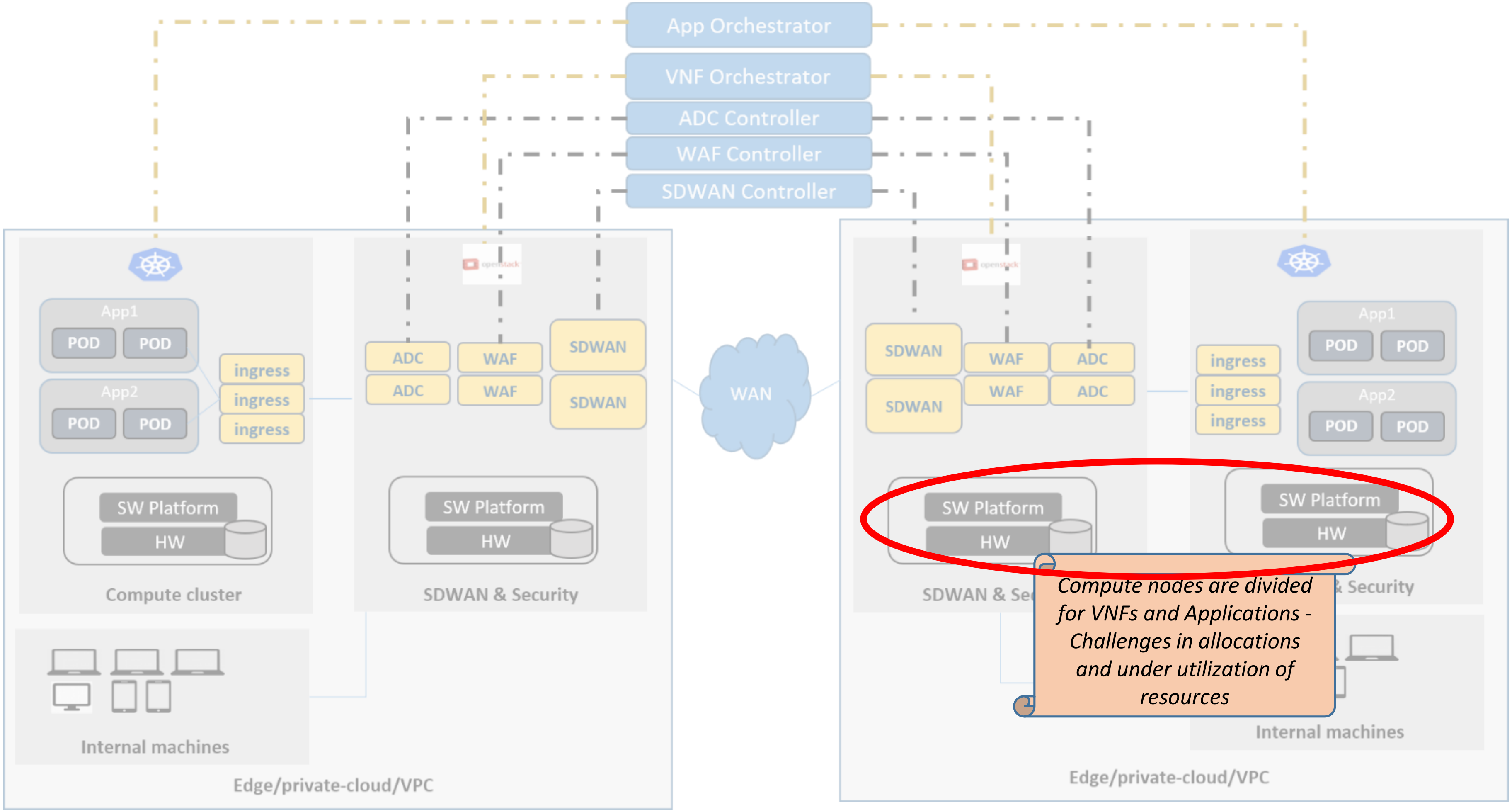
Centralized computing to Geo distributed computing

An App consisting of four Micro-services
ms1 talks to ms2, ms2 to ms3 and ms3 to ms4
 ms1” is user facing service
“ms1”, “ms2” are expected to be there together
“ms3” is stateful and hence need to talk to each

Current/In-progress Edge computing deployments : Multi-Cloud and Multi-Edge

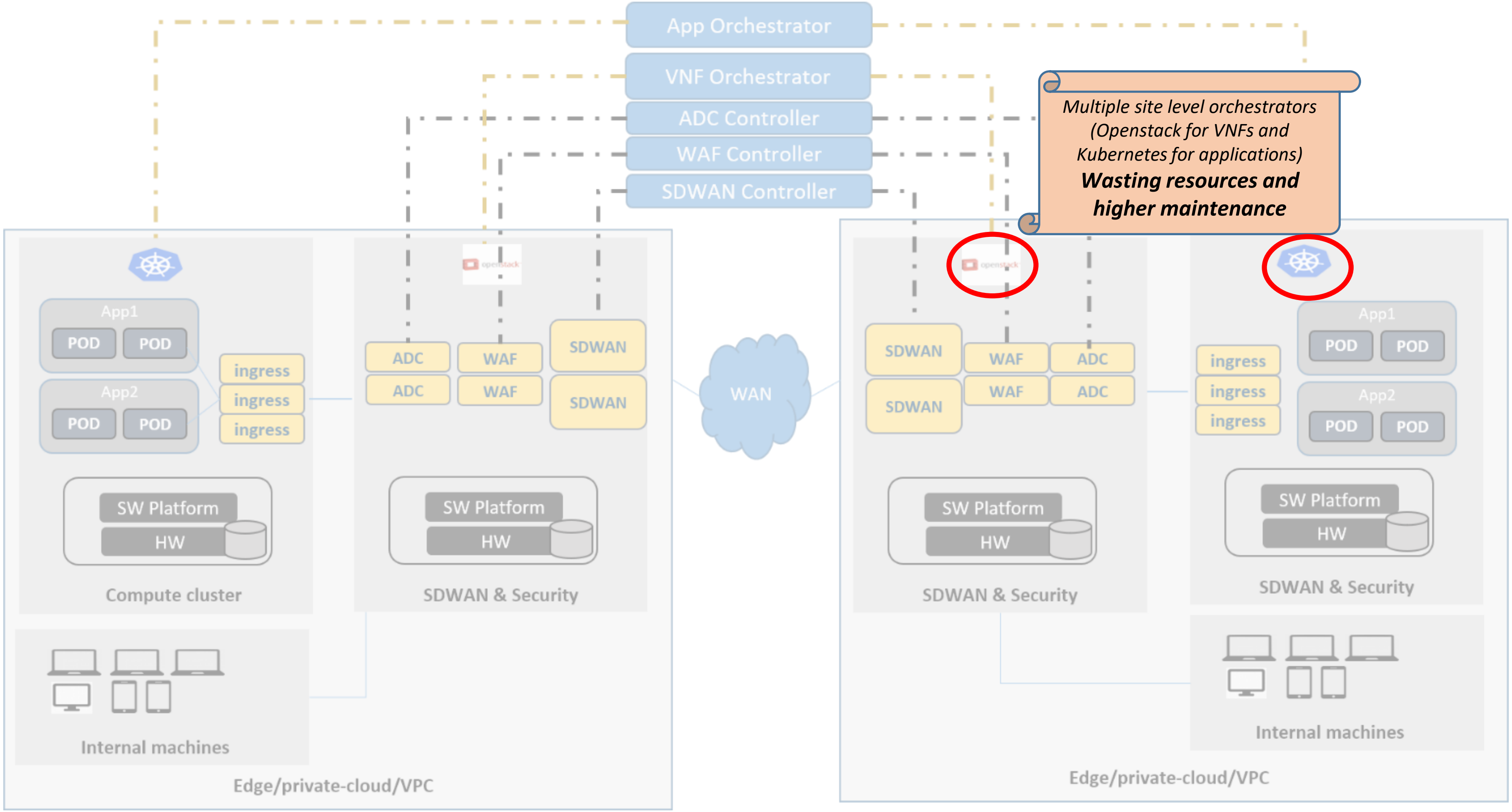


Challenge : Under utilization of resources



Compute nodes are divided for VNFs and Applications - Challenges in allocations and under utilization of resources

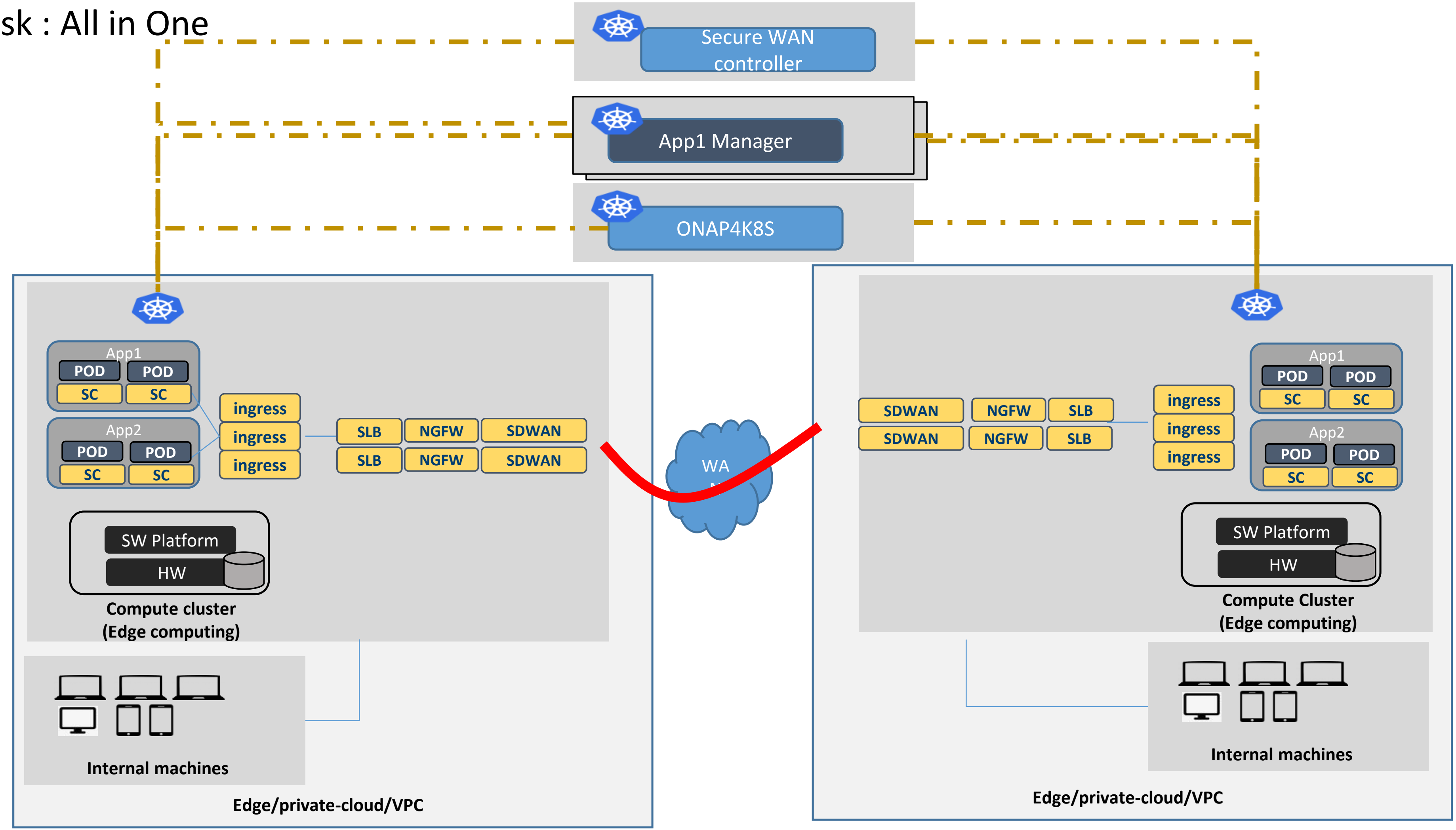
Challenge: Multiple Site level orchestrators leading to wasting of resources



Multiple site level orchestrators (Openstack for VNFs and Kubernetes for applications)

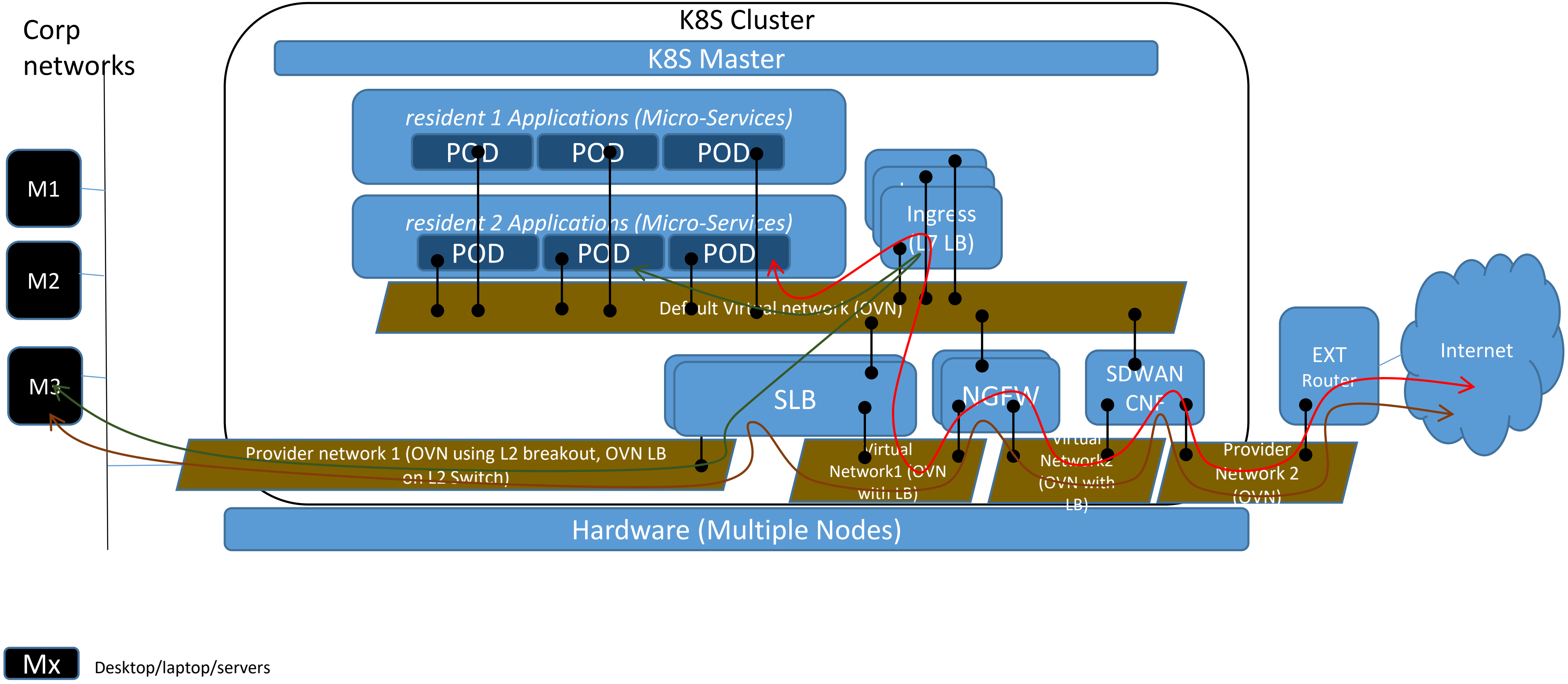
Wasting resources and higher maintenance

Ask : All in One



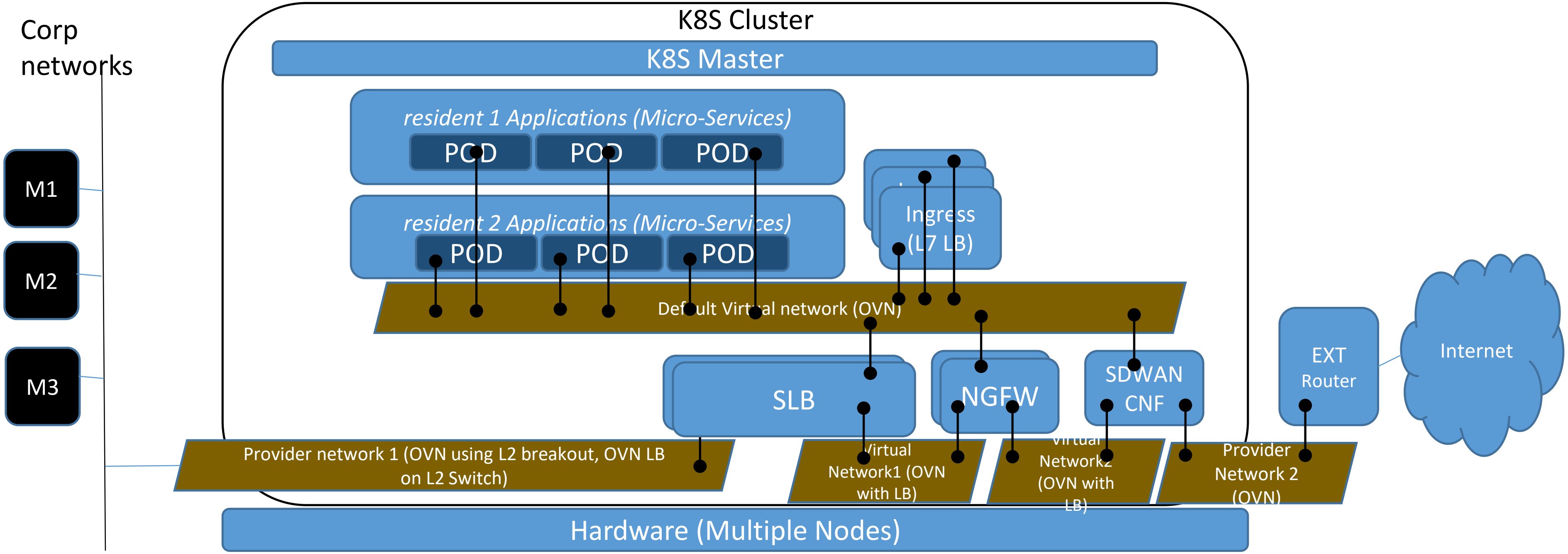
How does NFV based deployment with Cloud-native applications look like (Taking SDWAN with security NFs as an example)

View in Slide show



Corp networks
K8S Cluster
K8S Master
resident 1 Applications (Micro-Services)
POD
POD
POD
resident 2 Applications (Micro-Services)
POD
POD
POD
Ingress (L7 LB)
Default Virtual network (OVN)
M1
M2
M3
SLB
NGFW
SDWAN CNF
EXT Router
Internet
Provider network 1 (OVN using L2 breakout, OVN LB on L2 Switch)
Virtual Network1 (OVN with LB)
Virtual Network2 (OVN with LB)
Provider Network 2 (OVN)
Hardware (Multiple Nodes)
Mx
Desktop/laptop/servers

Networking Requirements


K8S Cluster
Corp networks
K8S Master
M1
M2
M3
resident 1 Applications (Micro-Services)
POD
POD
POD
resident 2 Applications (Micro-Services)
POD
POD
POD
Ingress (L7 LB)
Default Virtual network (OVN)
SLB
NGFW
SDWAN CNF
EXT Router
Internet
Provider network 1 (OVN using L2 breakout, OVN LB on L2 Switch)
Virtual Network1 (OVN with LB)
Virtual Network2 (OVN with LB)
Provider Network 2 (OVN)
Hardware (Multiple Nodes)

Feature Reqmts	Dynamic virtual Networks	Provider networks	Multiple interfaces	Network function chaining	Network function load balancing
Considerations	No changes to NFs	No changes to Apps	Configuration via operators	Cloud Native (No SRIOV requirement)	Smart NIC friendly & AF_XDP for packet processing NFs

Why did we choose OVN in Akraino Edge platform?

One of the best programmable controller

Hides OVS complexity

Broader eco-system

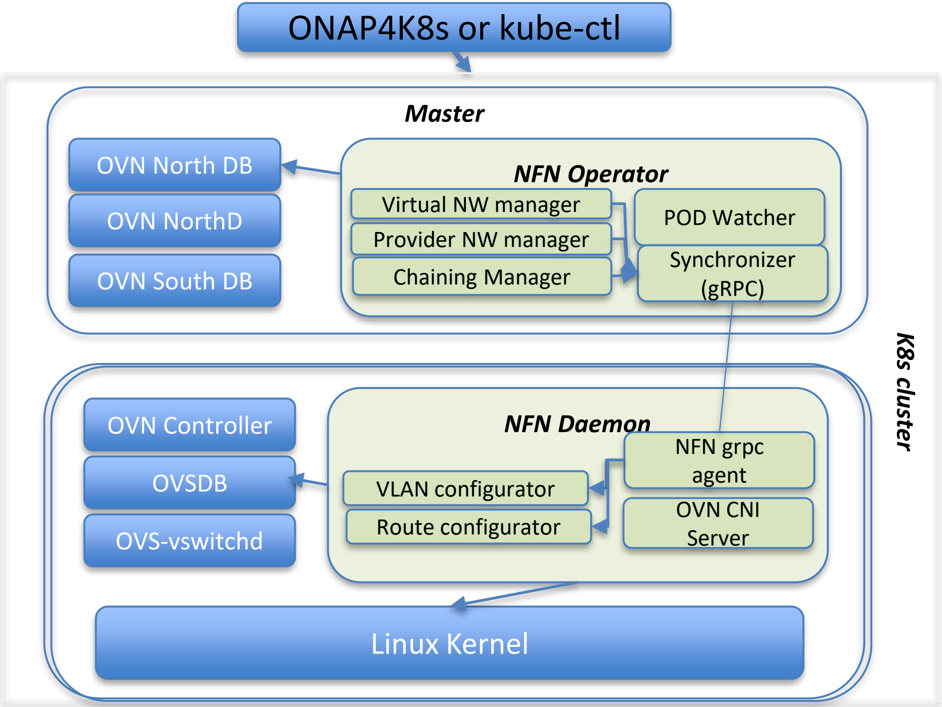
L2 CNI – Support for unicast, multicast, broadcast applications

One site level IPAM – No IP address restriction with number of nodes

Possible to implement critical features with table based pipeline
(Firewall, Routing, Switching, Load balancing)

SmartNIC friendly

OVN for K8S and NFV Architecture blocks



NFN Operator:

- Exposes virtual, provider, chaining CRDs to external world.
- Programs OVN to create L2 switches.
- Watches for PODs being coming up
  - Assigns IP addresses for every network of the deployment.
  - Looks for replicas and auto create routes for chaining to work.
  - Create LBs for distributing the load across CNF replicas.

NFN Daemon:

- Performs CNI operations.
- Configures VLAN and Routes in Linux kernel (in case of routes, it could do it in both root and network namespaces)
- Communicates with OVSDB to inform of provider interfaces. (creates ovs bridge and creates external-ids:ovn-bridge-mappings)

Proved with service orchestration (ONAP4K8s)

Used by Akarino ICN

Participants : Intel, Verizon, VMWare, F5

Virtual Network CR

```yaml
apiVersion: k8splugin.opnfv.org/v1alpha1
kind: Network
metadata:
  name: ovn-priv-net
spec:
  cniType: Ovn4nfv
  ipv4subnets:
  - subnet: 172.16.33.0/24
    name: subnet1
    gateway: 172.16.33.1/24
    excludeIps: 172.16.33.2 172.16.33.5..172.16.33.10
```

Creates OVN Switch with this configuration

Dynamic Multiple Network Interfaces

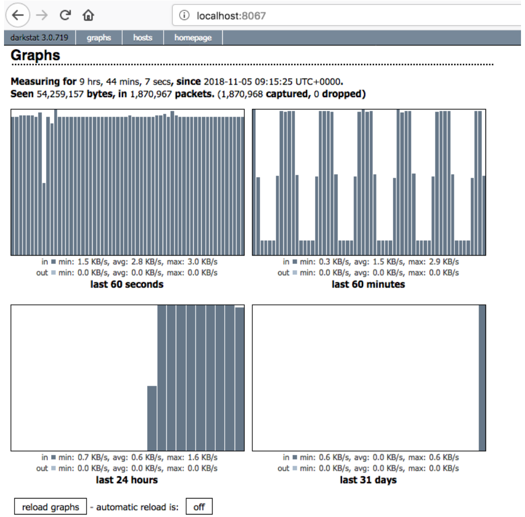
Pod Annotation

```
k8splugin.opnfv.org/nfn-network: '{ "type": "ovn4nfv", "interface": [
               { "name": “ovn-priv-net”, "interfaceRequest": "eth1" },
               { "name":  “ovn-prot-net”, "interfaceRequest": "eth2" }
        ]}’
```

- CNI (ovn4k8s-cni) Works with Multus
- Assumes primary/first interface provided by another CNI
- Supports Static IP addresses

ONAP vFW Use case using OVN dynamic networks

localhost



localhost:8067

darkstat 3.0.719 | graphs | hosts | homepage

Graphs

Measuring for 9 hrs, 44 mins, 7 secs, **since** 2018-11-05 09:15:25 UTC+0000.
Seen 54,259,157 **bytes, in** 1,870,967 **packets.** (1,870,968 **captured,** 0 **dropped)**

in ■ min: 1.5 KB/s, avg: 2.8 KB/s, max: 3.0 KB/s
out ■ min: 0.0 KB/s, avg: 0.0 KB/s, max: 0.0 KB/s
last 60 seconds

in ■ min: 0.3 KB/s, avg: 1.5 KB/s, max: 2.9 KB/s
out ■ min: 0.0 KB/s, avg: 0.0 KB/s, max: 0.0 KB/s
last 60 minutes

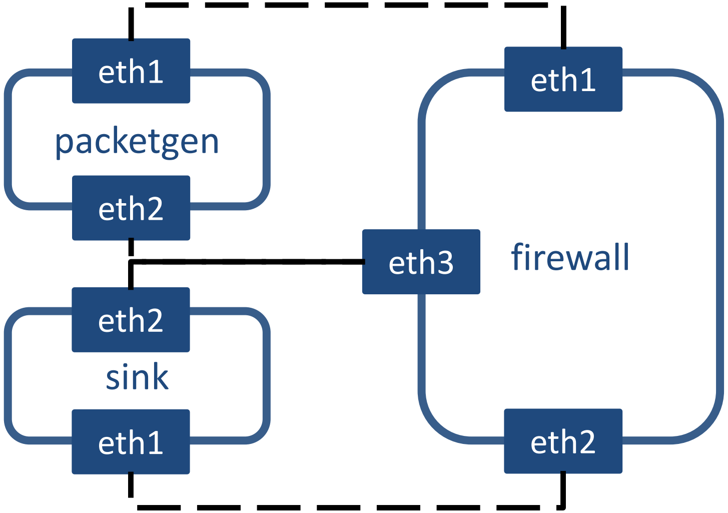
in ■ min: 0.7 KB/s, avg: 0.6 KB/s, max: 1.6 KB/s
out ■ min: 0.0 KB/s, avg: 0.0 KB/s, max: 0.0 KB/s
last 24 hours

in ■ min: 0.6 KB/s, avg: 0.0 KB/s, max: 0.6 KB/s
out ■ min: 0.0 KB/s, avg: 0.0 KB/s, max: 0.0 KB/s
last 31 days

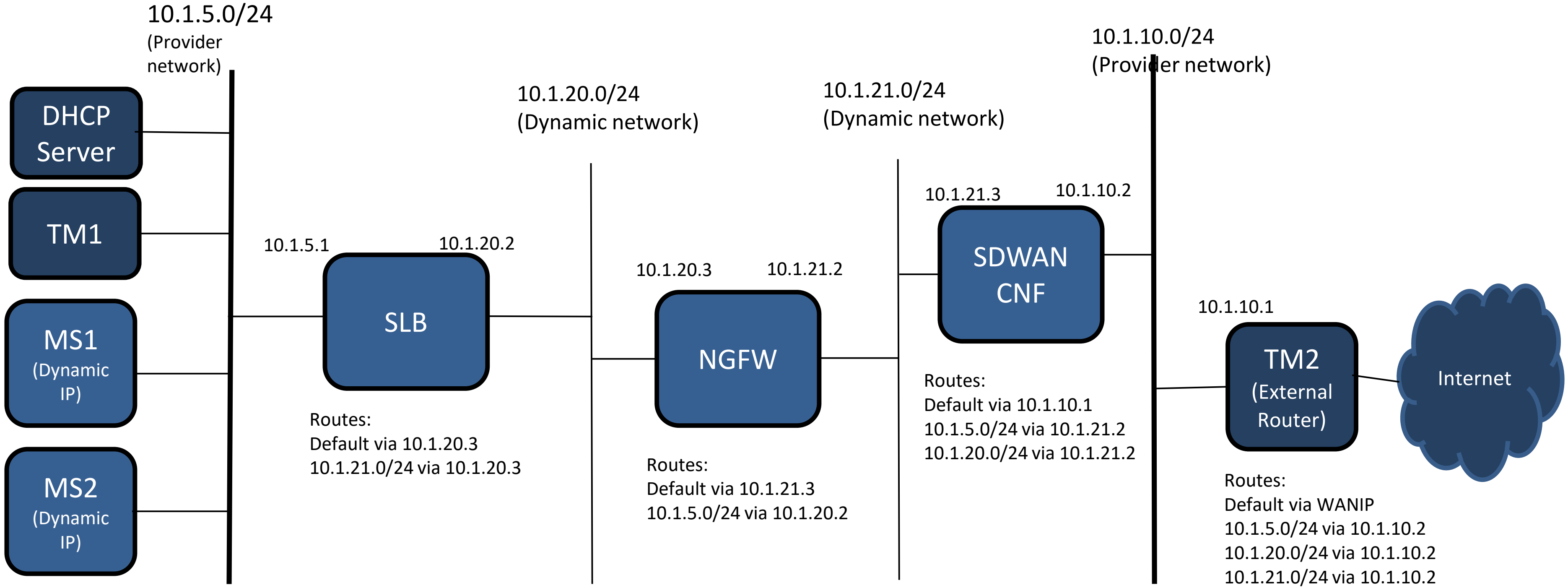
reload graphs - automatic reload is: off

x.x.x.x

kubernetes



Test scenario – to comprehend multiple deployment variations



10.1.5.0/24
(Provider network)

10.1.20.0/24
(Dynamic network)

10.1.21.0/24
(Dynamic network)

10.1.10.0/24
(Provider network)

DHCP Server

TM1

MS1
(Dynamic IP)

MS2
(Dynamic IP)

10.1.5.1

10.1.20.2

SLB

Routes:
Default via 10.1.20.3
10.1.21.0/24 via 10.1.20.3

10.1.20.3

10.1.21.2

NGFW

Routes:
Default via 10.1.21.3
10.1.5.0/24 via 10.1.20.2

10.1.21.3

10.1.10.2

SDWAN CNF

Routes:
Default via 10.1.10.1
10.1.5.0/24 via 10.1.21.2
10.1.20.0/24 via 10.1.21.2

10.1.10.1

TM2
(External Router)

Internet

Routes:
Default via WANIP
10.1.5.0/24 via 10.1.10.2
10.1.20.0/24 via 10.1.10.2
10.1.21.0/24 via 10.1.10.2

Default route: 10.1.5.1

■ External existing entities ■ VNF/CNFs

Provider Network CR

```
apiVersion:  k8splugin.opnfv.org/v1alpha1
kind: OvnProviderNetwork
metadata:
  name: ovn-provider-net
spec:
  cniType: Ovn4nfv
  ipv4subnets:
  - subnet: 172.16.33.0/24
    name: subnet1
    gateway: 172.16.33.1/24
    excludeIps: 172.16.33.2 172.16.33.5..172.16.33.10
  providerNetworkType: vlan
  vlan:
    vlanId: 100
    providerInterfaceName: eth0
    Node: node1,node2
    logicalInterfaceName: eth0.100
```

Create OVN Switch and configures nodes

Provider Network Functionality

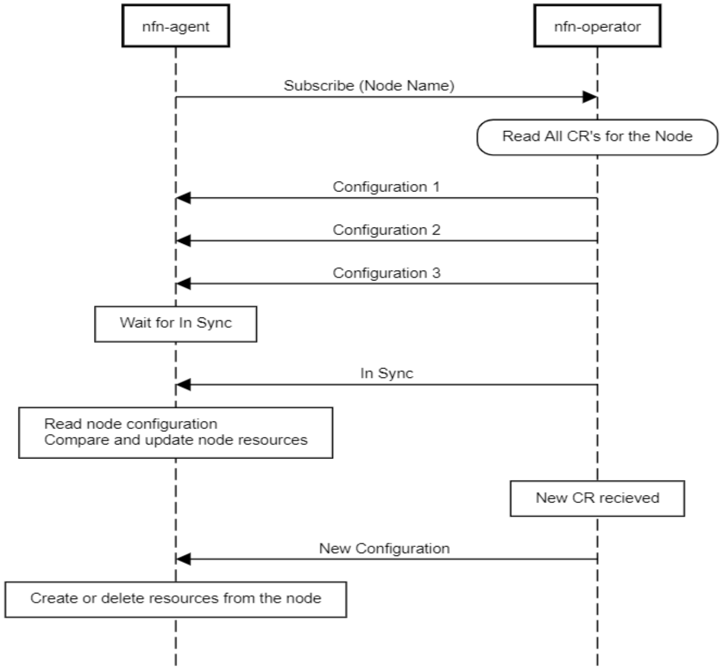
- CR creates OVN Switch
- Per Node (can be list of nodes, “all” nodes or “any” node)
  - Creates VLAN interfaces
  - Creates OVS Bridge and attaches VLAN interface
  - Configure ovs external-ids:ovn-bridge-mappings

- Pod annotation for attaching Provider network to a Pod

```
k8splugin.opnfv.org/nfn-network: '{ "type": "ovn4nfv", "interface": [
            { "name": “ovn-provider-net”, "interfaceRequest": “net0" }
      ]}’
```

NFN Operator and Agent Communication

nfn-operator and agent



nfn-agent

nfn-operator

Subscribe (Node Name)

Read All CR's for the Node

Configuration 1

Configuration 2

Configuration 3

Wait for In Sync

In Sync

Read node configuration
Compare and update node resources

New CR recieved

New Configuration

Create or delete resources from the node

Network Chaining CR

```yaml
apiVersion:  k8splugin.opnfv.org/v1alpha1
kind: NetworkChaining
metadata:
  name: chain1
  namespace: vFW
spec:
  type: Routing
  routingSpec:
    leftNetwork:
      - networkName: ovn-provider1
        gatewayIP: 10.1.5.1
        subnet: 10.1.5.0/24
    rightNetwork:
      - networkName: ovn-provider1
        gatewayIP: 10.1.10.1
        subnet: default
    networkChain: app=slb, ovn-net1, app=ngfw, ovn-net2, app=sdwancnf
```

Inserts routes in Container Namespaces

Status

Current

- Dynamic Network Creation
- Attaching multiple interfaces to Pods
- Multus integration
- Support for Virtlet with Multus
- Provider Network Support – Controller and Agent

Link to Repo: https://github.com/opnfv/ovn4nfv-k8s-plugin

WIP

- Chaining Controller

Roadmap / Wishlist

- Enabling SDWAN use case
  - Support for primary interface with NFN Operator and CNI
  - One switch for the whole deployment
- Network Policy Support
- LB support for NF elasticity
- Integrate with Kubevirt
- Proxy less service mesh with OVN & IPsec in network namespace