OVN4NFV – SFC

By:
Kuralamudhan Ramakrishnan(kuralamudhan.ramakrishnan@intel.com) | October 8th, 2020

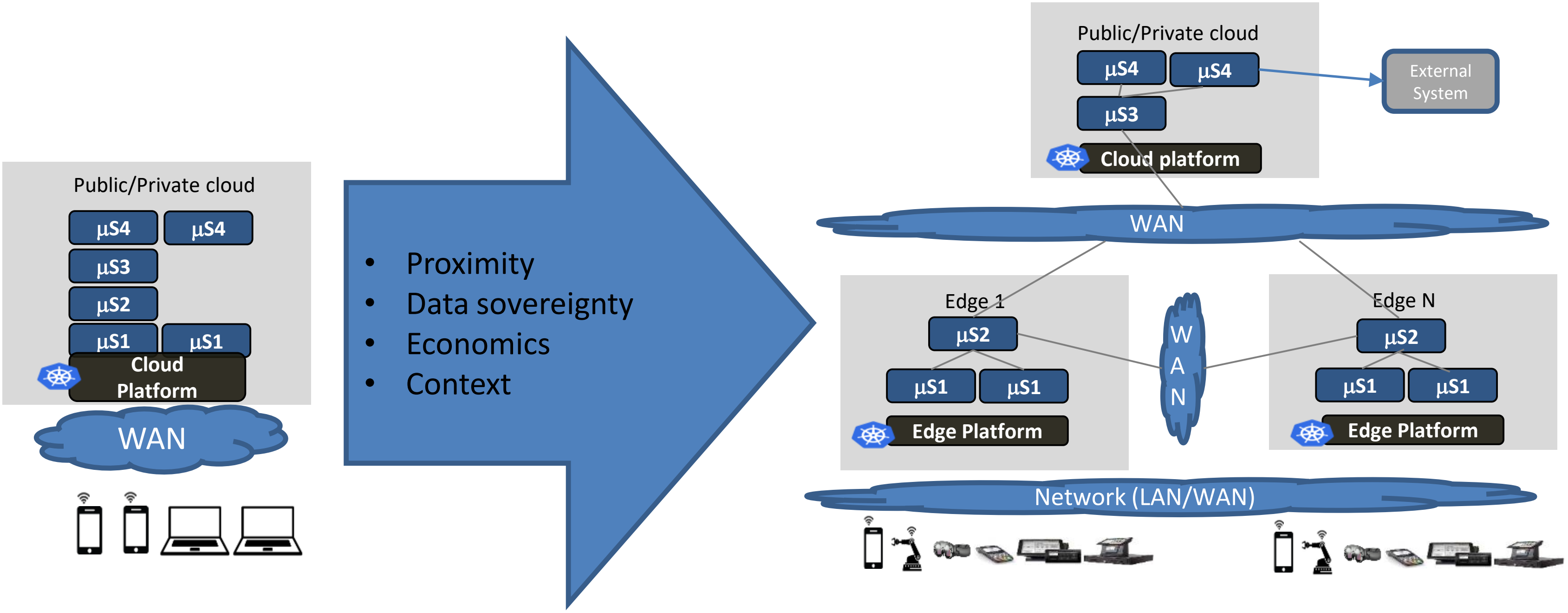
Acknowledgement:
Srinivasa Addepalli, Ritu Sood

Agenda

- Why network functions in Edge & K8s clusters?
- Edge-computing scenario to describe the K8s networking requirements
- Networking requirements
- OVN4NFV –SFC
- SDEWAN Use case

Application Transformation

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



Public/Private cloud

µS4 µS4

µS3

µS2

µS1 µS1

Cloud Platform

WAN

- Proximity
- Data sovereignty
- Economics
- Context

Public/Private cloud

µS4 µS4

µS3

Cloud platform

External System

WAN

Edge 1

µS2

µS1 µS1

Edge Platform

W A N

Edge N

µS2

µS1 µS1

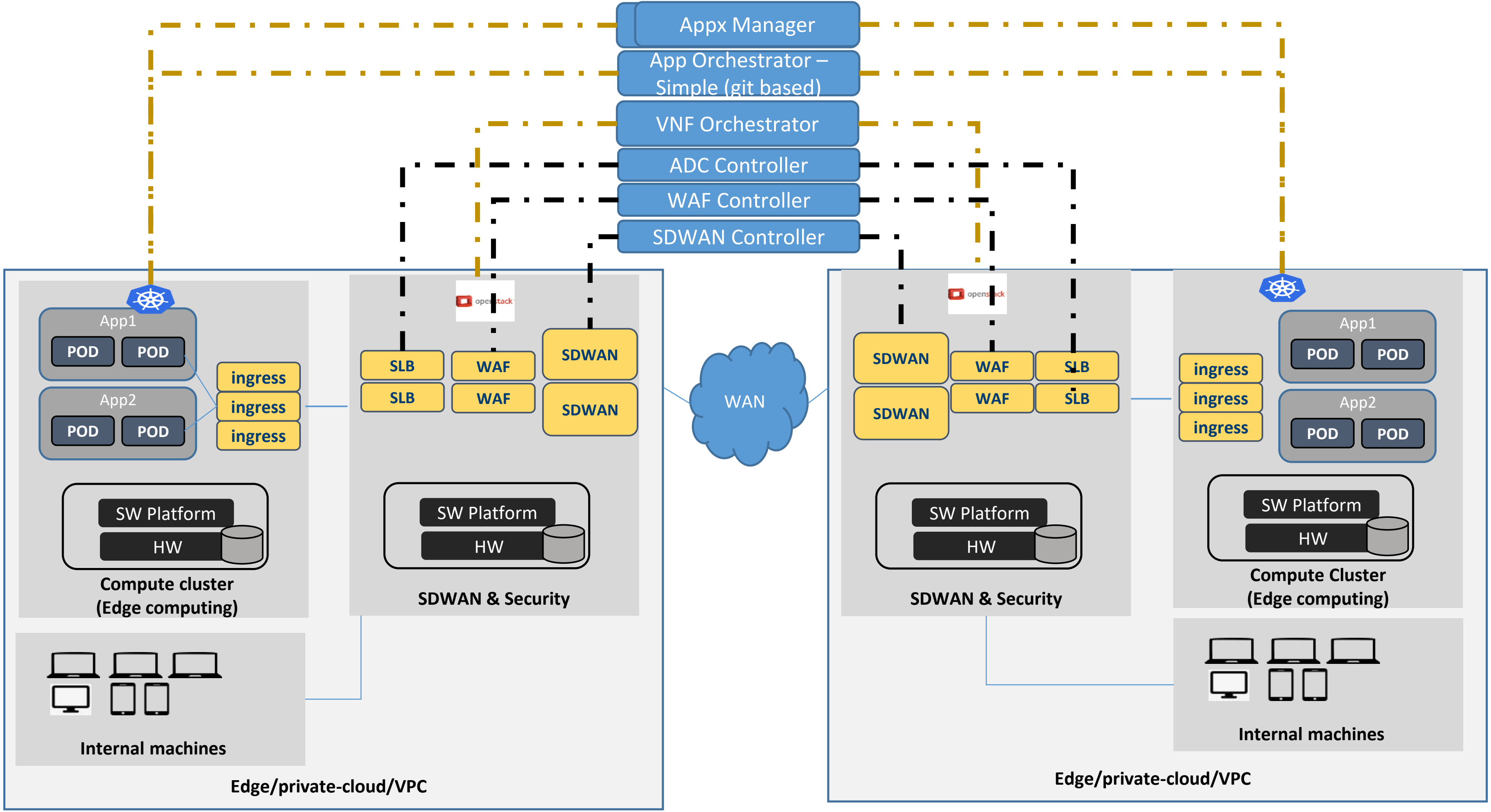
Edge Platform

Network (LAN/WAN)

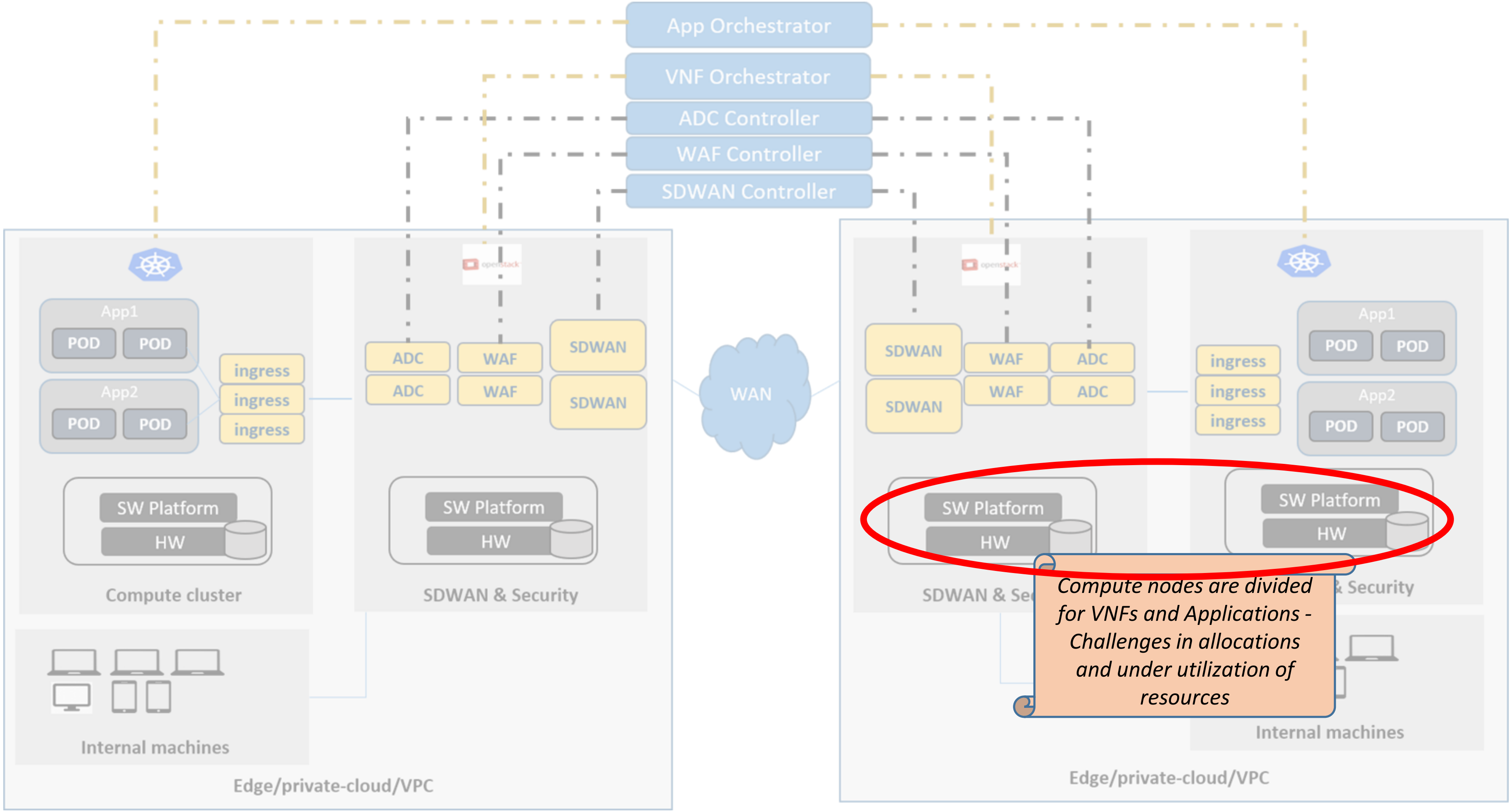
Centralized computing to Geo distributed computing

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
"ms2" is stateful and hence need to talk to each other

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

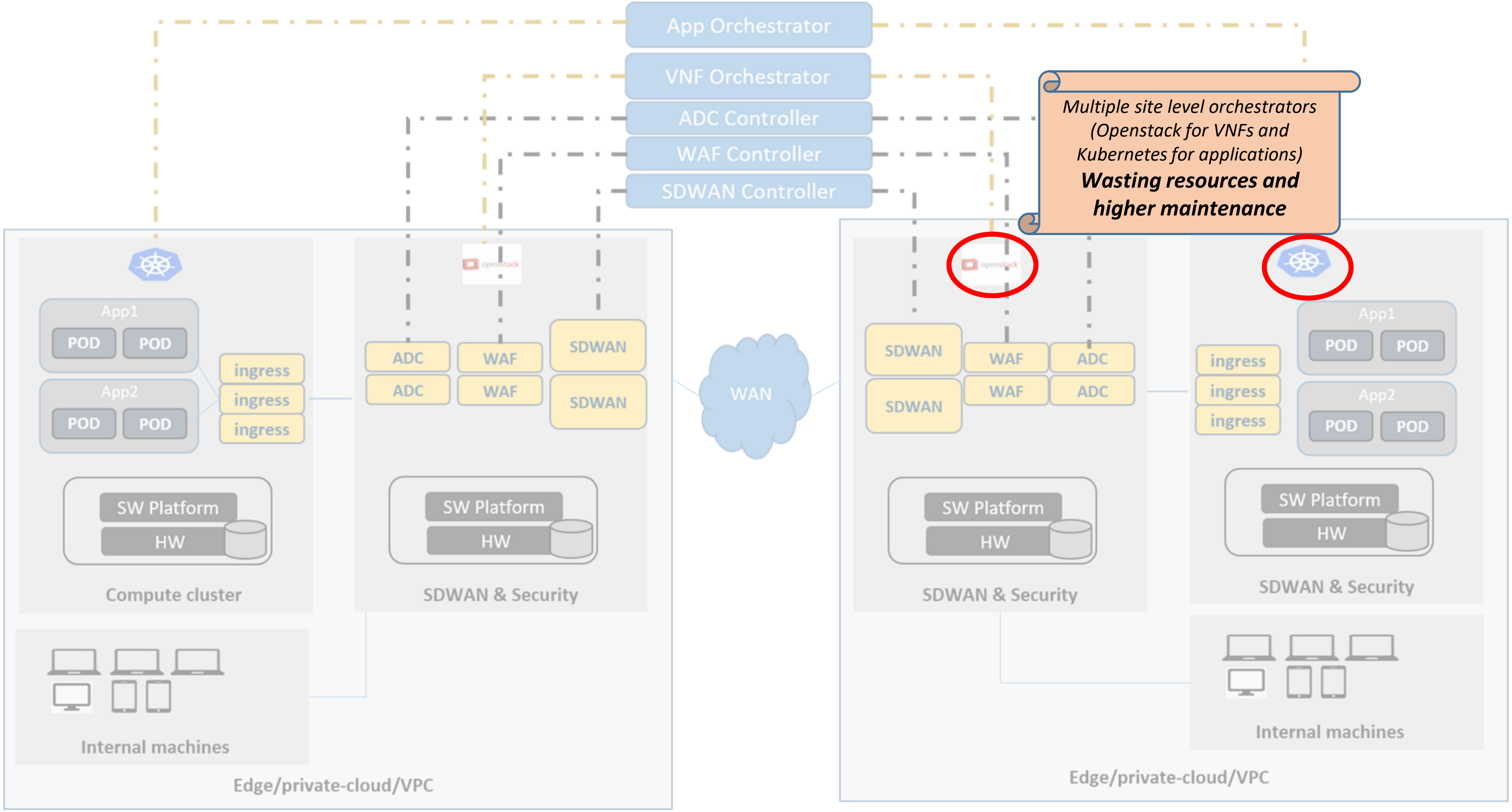


Challenge : Under utilization of resources



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

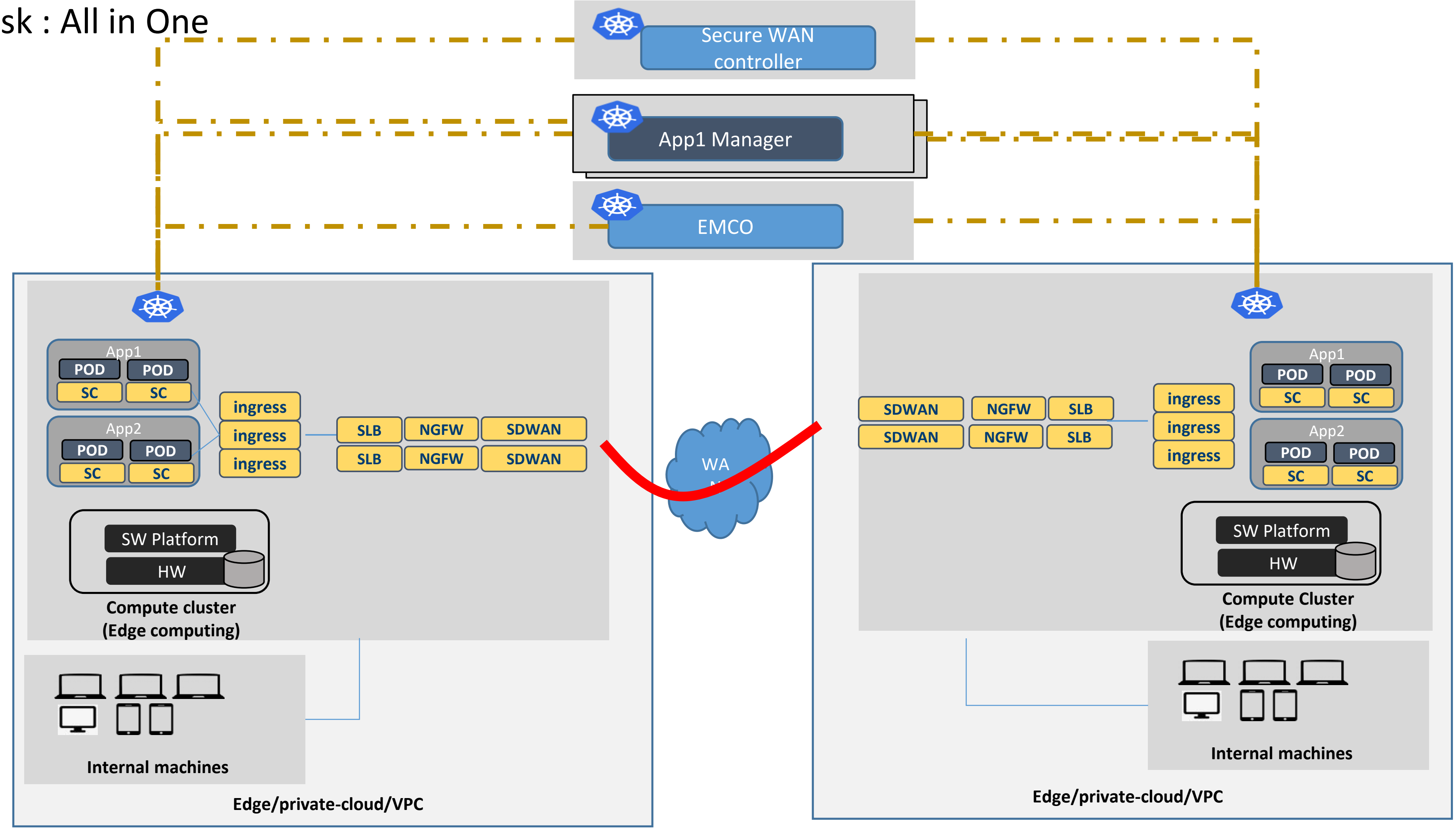
Challenge: Multiple Site level orchestrators leading to wasting of resources



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

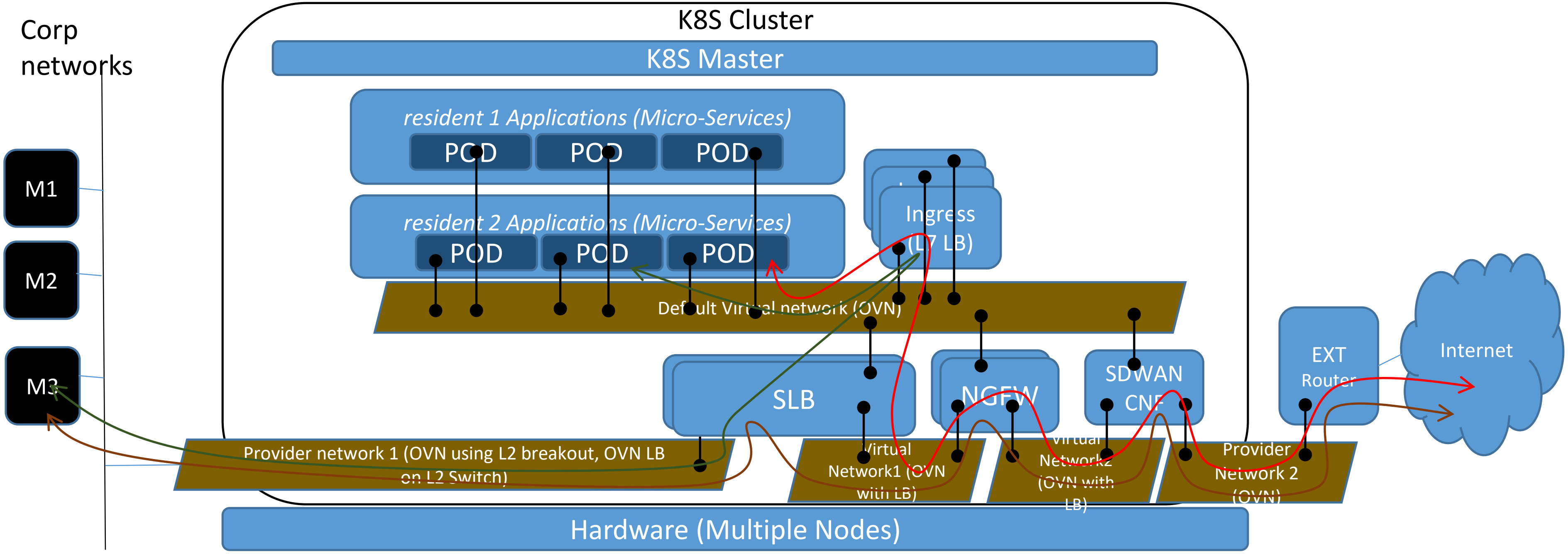
Wasting resources and higher maintenance

Ask : All in One



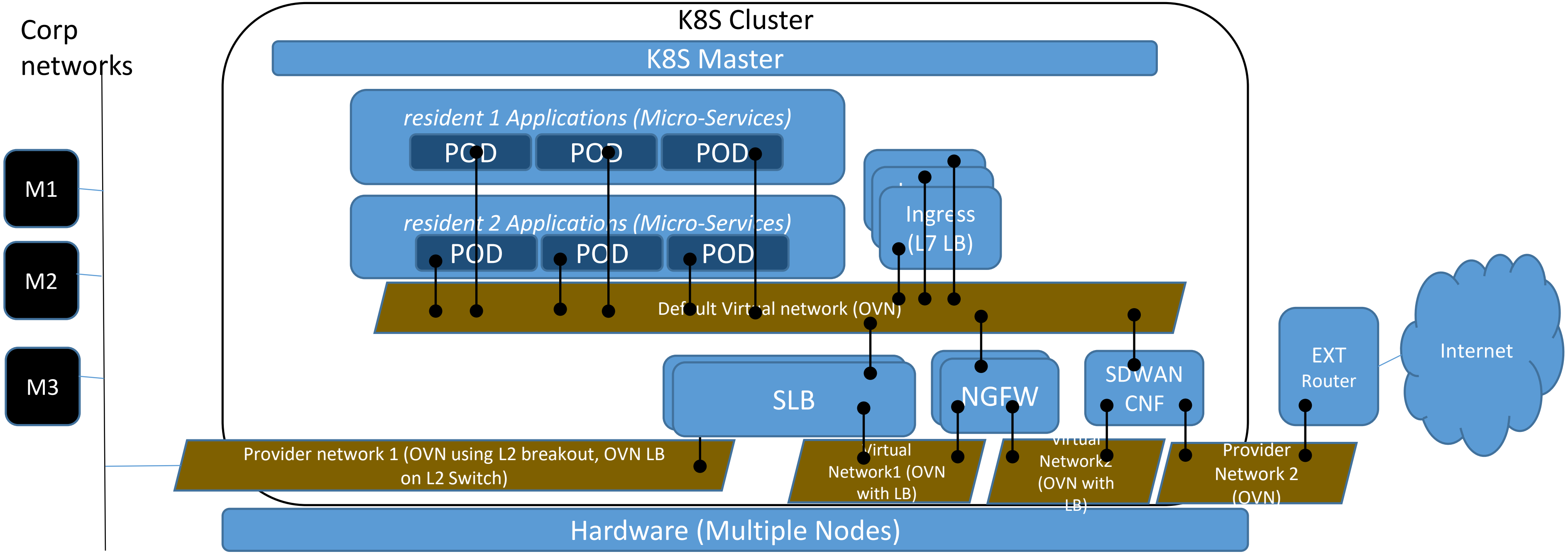
How does NFV based deployment with Cloud-native applications look like (Taking SDWAN with security NFs as an example)

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)
SLB
NGFW
SDWAN CNF
EXT Router
Internet
M1
M2
M3
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

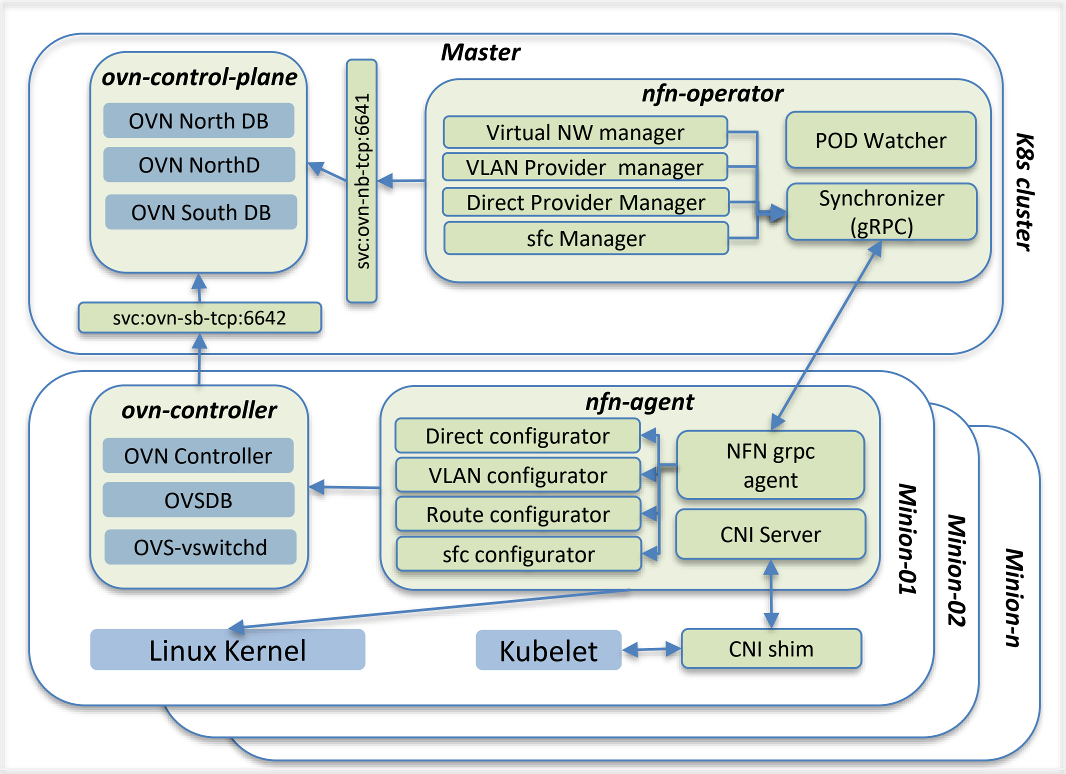

Corp networks
K8S Cluster
K8S Master
resident 1 Applications (Micro-Services)
POD
POD
POD
resident 2 Applications (Micro-Services)
POD
POD
POD
Ingress (L7 LB)
Default Virtual network (OVN)
M1
M2
M3
SLB
NGFW
SDWAN CNF
EXT Router
Internet
Provider network 1 (OVN using L2 breakout, OVN LB on L2 Switch)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

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	Finite network SRIOV Overlay networking	Smart NIC friendly & AF_XDP for packet processing NFs

Why did we choose OVN in for Edge Networking?

- 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



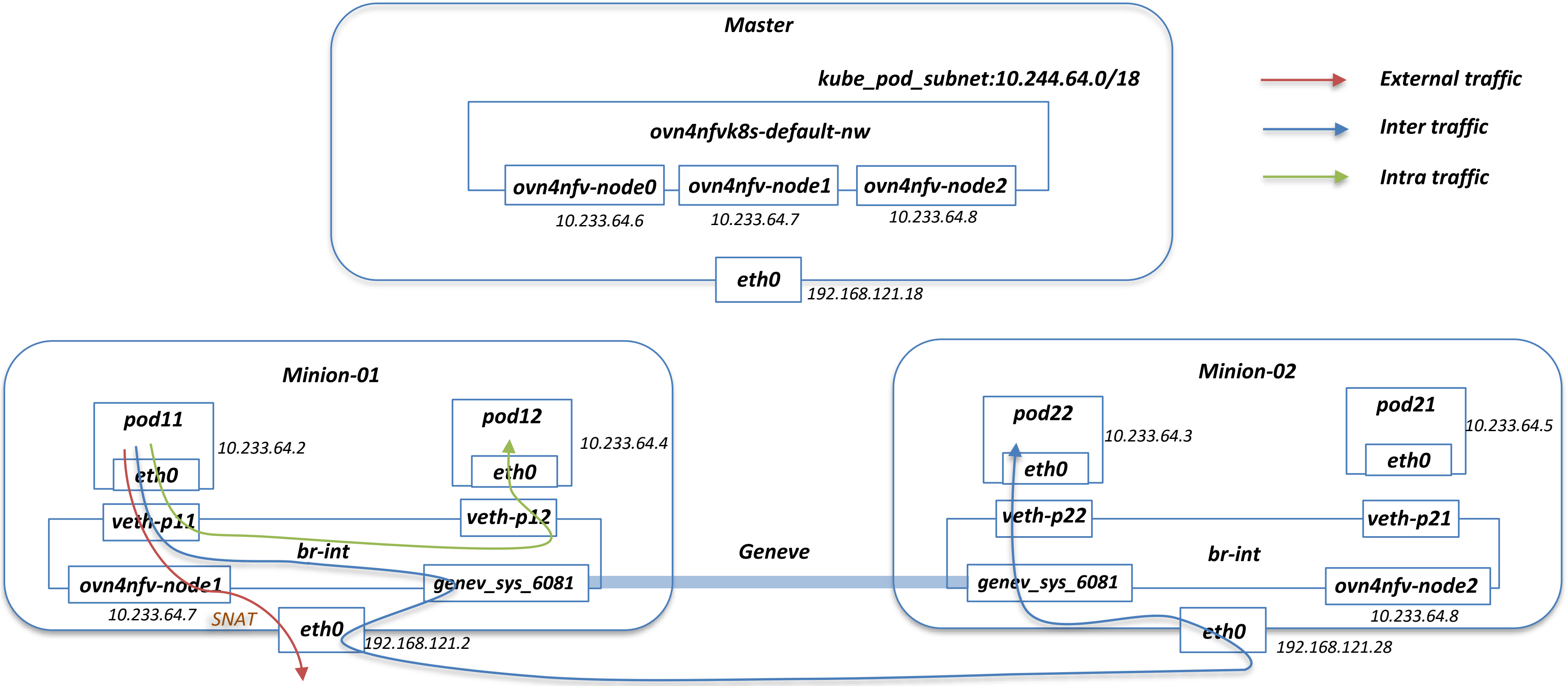
NFN Operator:

- 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 agent:

- 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)

Network traffic between pods



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" }
        ]}’
```

- Assumes primary/first interface provided by another CNI
- Supports Static IP addresses

Provider Network CR

```yaml
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" }
      ]}’
```

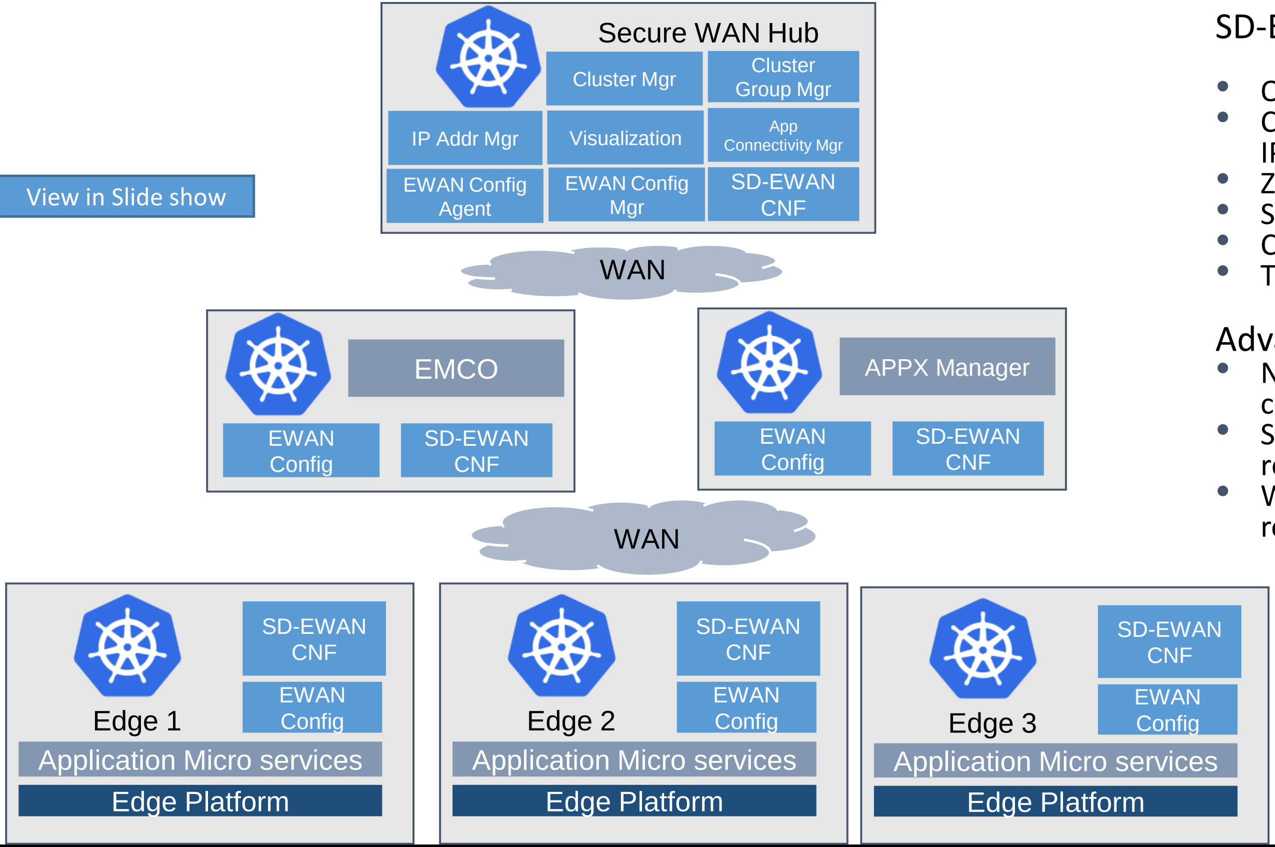
Network Chaining CR

```
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

Software Defined Edge WAN (SDEWAN)

View in Slide show



SD-EWAN

- Open WRT based SE-DWAN CNFS
- Cloud Native based SD-EWAN controller and IPSec controller
- Zero touch automation
- Solution to all Edge Challenges identified
- Centralization controller for configuration
- Traffic Hub for sanitization

Advantages

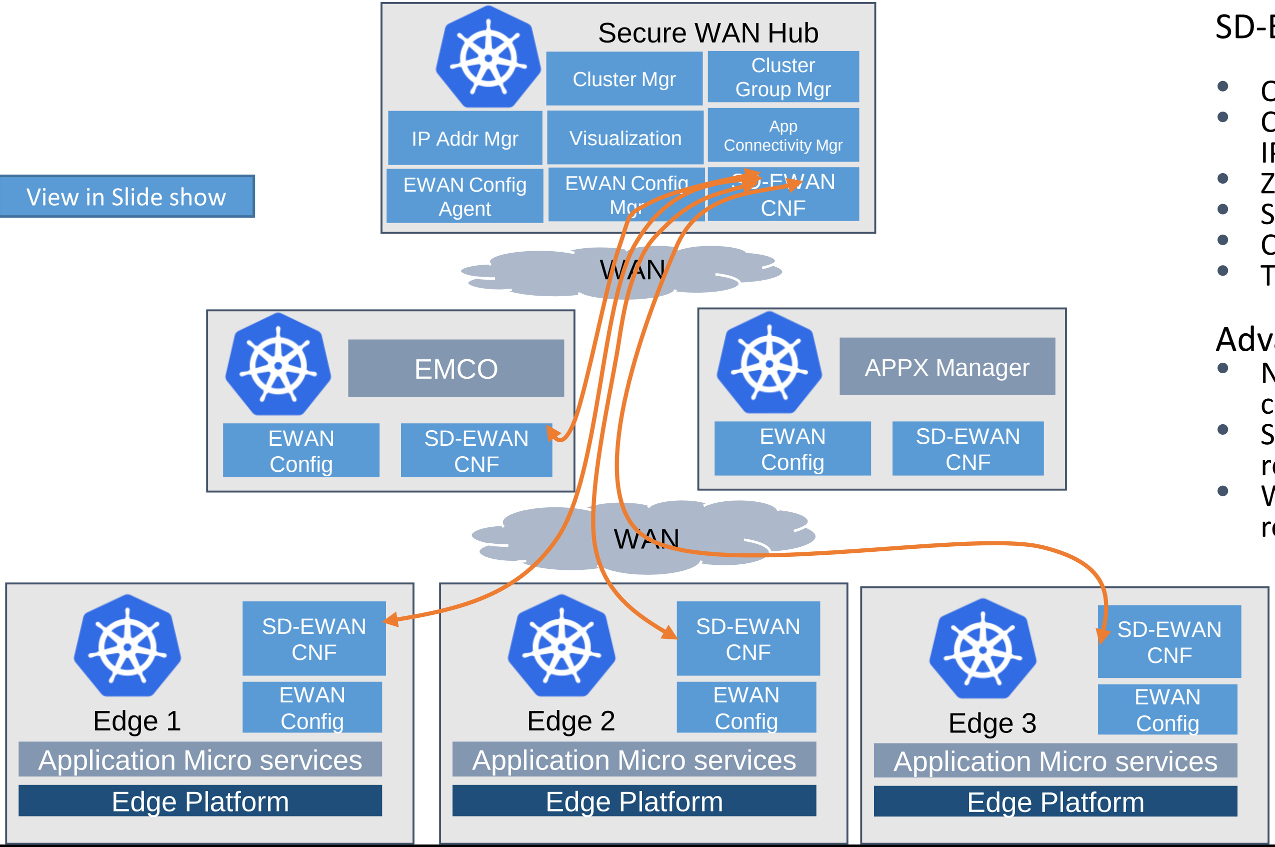
- No changes to application Micro services and configuring Edges
- Supporting both green field and brownfield requirements
- Work with third party SD-WAN VNFs (future roadmap)

Refer

Repo: https://gerrit.akraino.org/r/admin/repos/icn/sdwan

Software Defined Edge WAN (SDEWAN)

View in Slide show



SD-EWAN

- Open WRT based SE-DWAN CNFS
- Cloud Native based SD-EWAN controller and IPSec controller
- Zero touch automation
- Solution to all Edge Challenges identified
- Centralization controller for configuration
- Traffic Hub for sanitization

Advantages

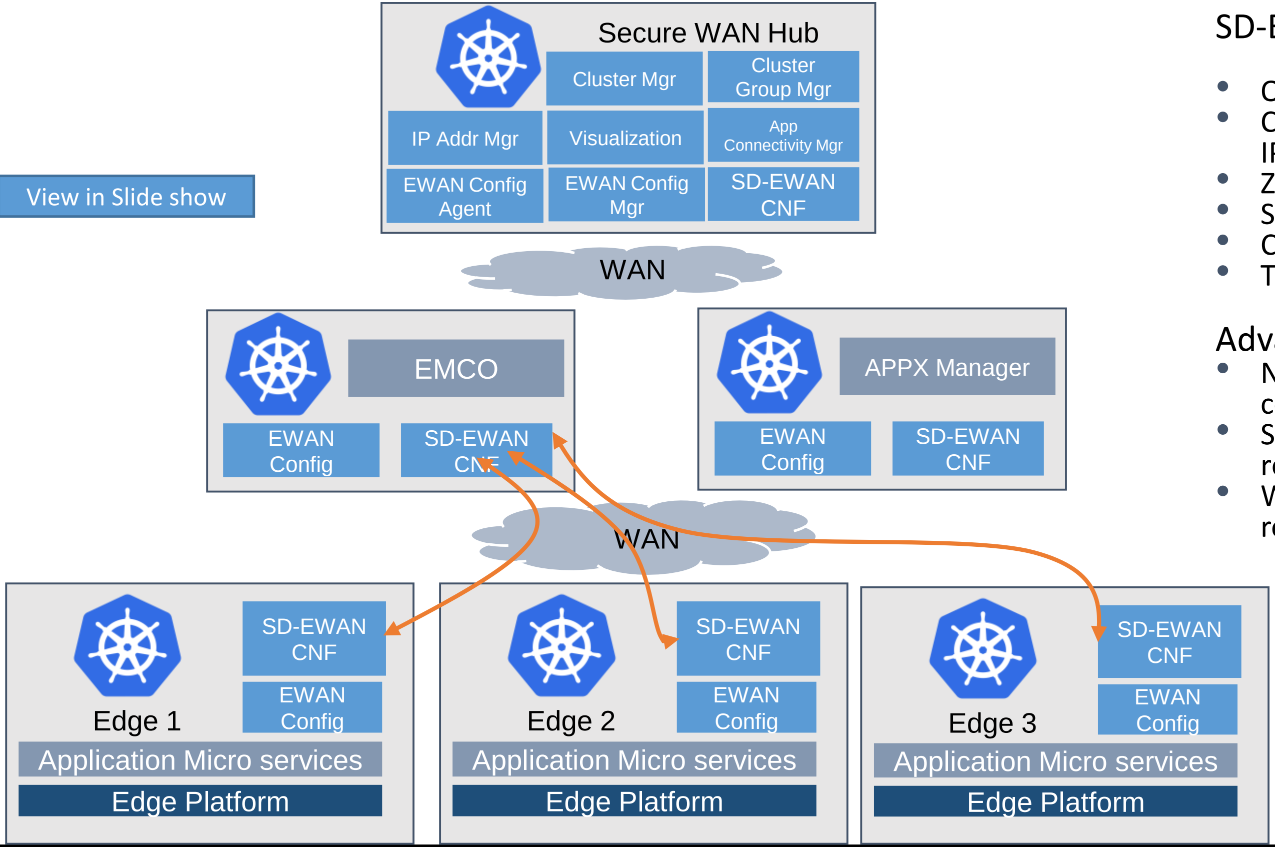
- No changes to application Micro services and configuring Edges
- Supporting both green field and brownfield requirements
- Work with third party SD-WAN VNFs (future roadmap)

Refer

Repo:
https://gerrit.akraino.org/r/admin/repos/icn/sdwan

Software Defined Edge WAN (SDEWAN)

View in Slide show



SD-EWAN

- Open WRT based SE-DWAN CNFS
- Cloud Native based SD-EWAN controller and IPSec controller
- Zero touch automation
- Solution to all Edge Challenges identified
- Centralization controller for configuration
- Traffic Hub for sanitization

Advantages

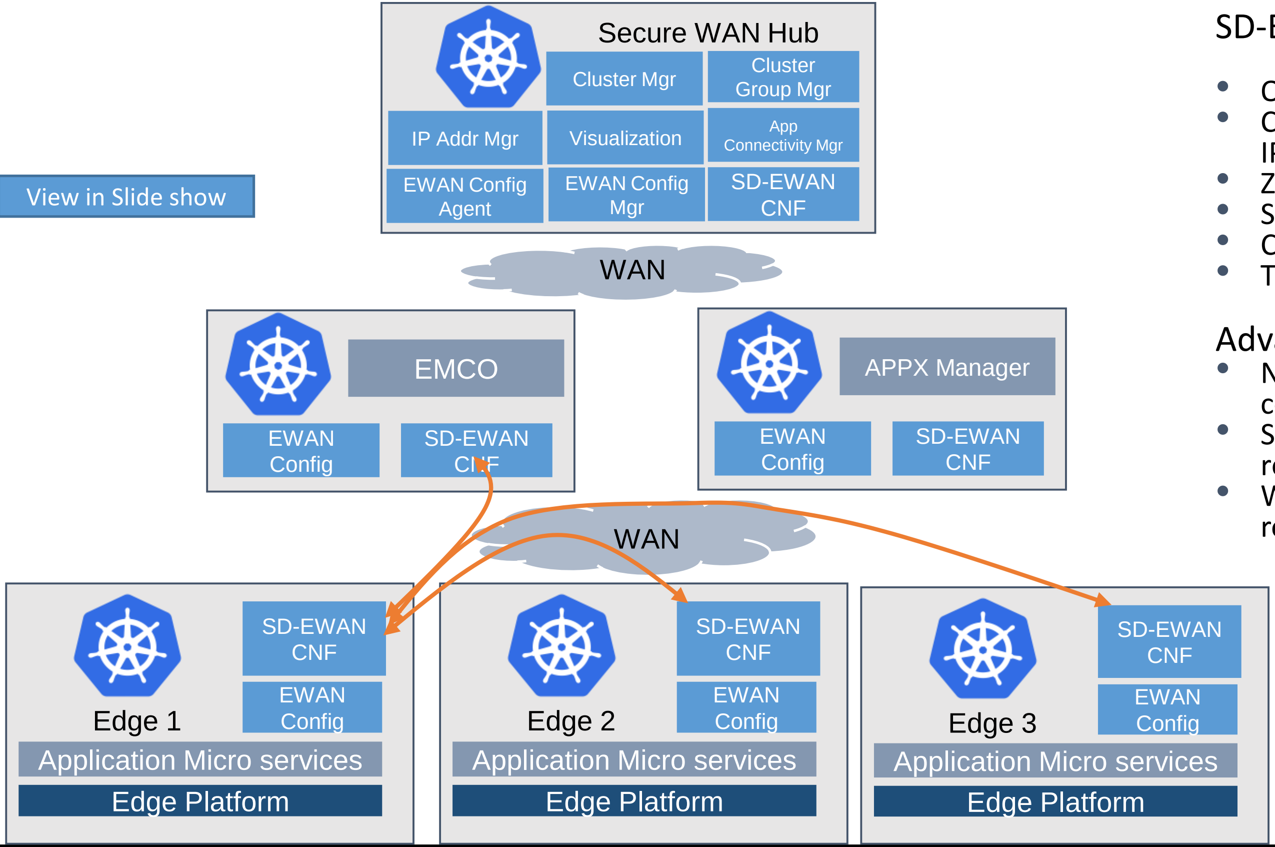
- No changes to application Micro services and configuring Edges
- Supporting both green field and brownfield requirements
- Work with third party SD-WAN VNFs (future roadmap)

Refer

Repo: https://gerrit.akraino.org/r/admin/repos/icn/sdwan

Software Defined Edge WAN (SDEWAN)

View in Slide show



SD-EWAN

- Open WRT based SE-DWAN CNFS
- Cloud Native based SD-EWAN controller and IPSec controller
- Zero touch automation
- Solution to all Edge Challenges identified
- Centralization controller for configuration
- Traffic Hub for sanitization

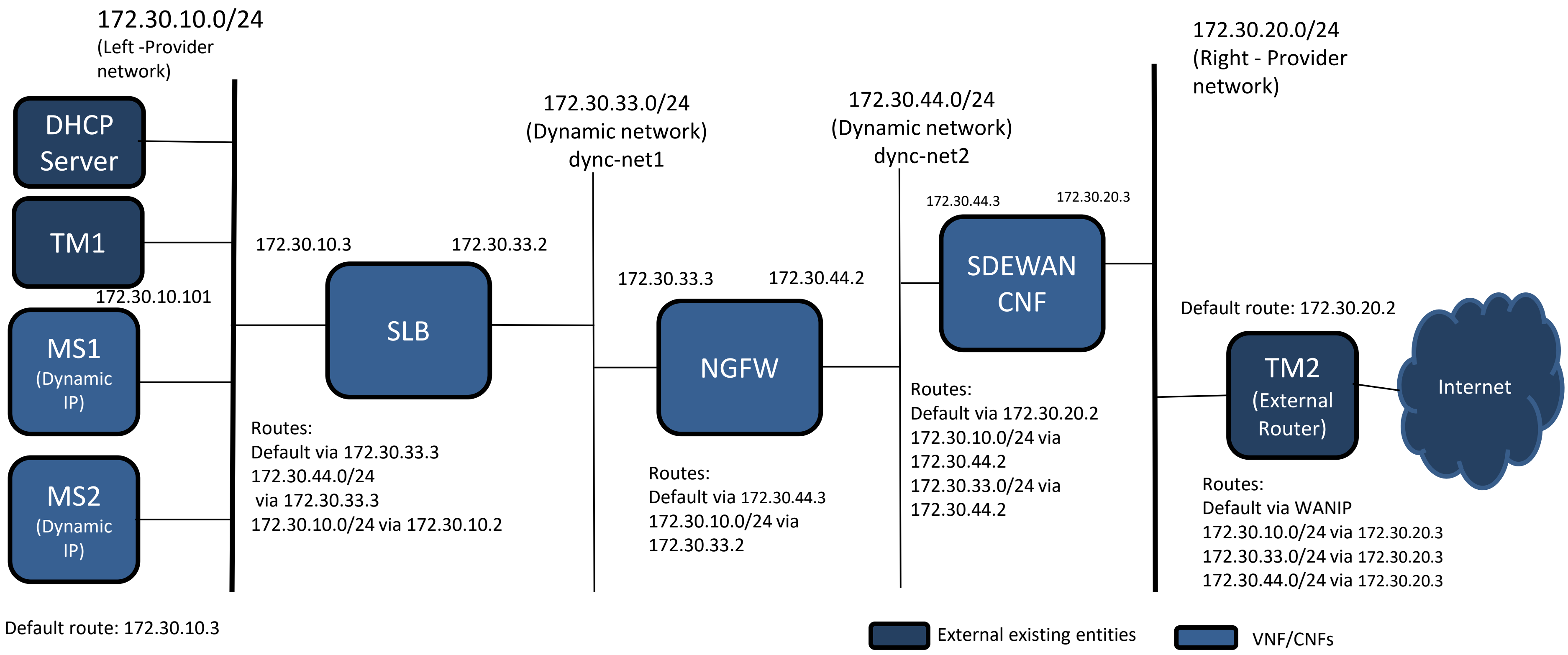
Advantages

- No changes to application Micro services and configuring Edges
- Supporting both green field and brownfield requirements
- Work with third party SD-WAN VNFs (future roadmap)

Refer

Repo: https://gerrit.akraino.org/r/admin/repos/icn/sdwan

Test scenario – to comprehend multiple deployment variations



172.30.10.0/24
(Left -Provider network)

DHCP Server

TM1

172.30.10.101

MS1 (Dynamic IP)

MS2 (Dynamic IP)

Default route: 172.30.10.3

172.30.10.3

172.30.33.2

SLB

Routes:
Default via 172.30.33.3
172.30.44.0/24
via 172.30.33.3
172.30.10.0/24 via 172.30.10.2

172.30.33.0/24
(Dynamic network)
dync-net1

172.30.33.3

172.30.44.2

NGFW

Routes:
Default via 172.30.44.3
172.30.10.0/24 via
172.30.33.2

172.30.44.0/24
(Dynamic network)
dync-net2

172.30.44.3

172.30.20.3

SDEWAN CNF

Routes:
Default via 172.30.20.2
172.30.10.0/24 via
172.30.44.2
172.30.33.0/24 via
172.30.44.2

172.30.20.0/24
(Right - Provider network)

Default route: 172.30.20.2

TM2 (External Router)

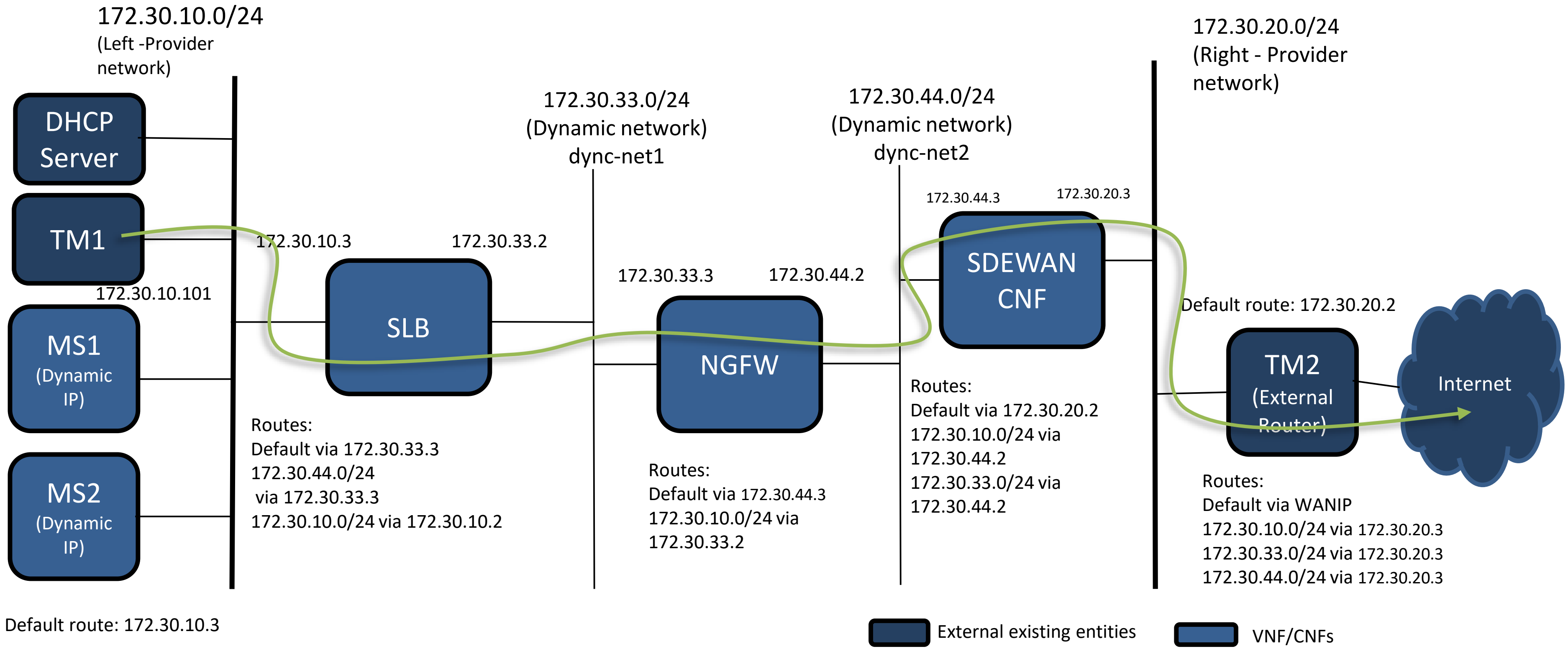
Internet

Routes:
Default via WANIP
172.30.10.0/24 via 172.30.20.3
172.30.33.0/24 via 172.30.20.3
172.30.44.0/24 via 172.30.20.3

▬ External existing entities

▬ VNF/CNFs

Traffic from external entities with sfc



Default route: 172.30.20.2

Routes:
Default via 172.30.33.3
172.30.44.0/24
via 172.30.33.3
172.30.10.0/24 via 172.30.10.2

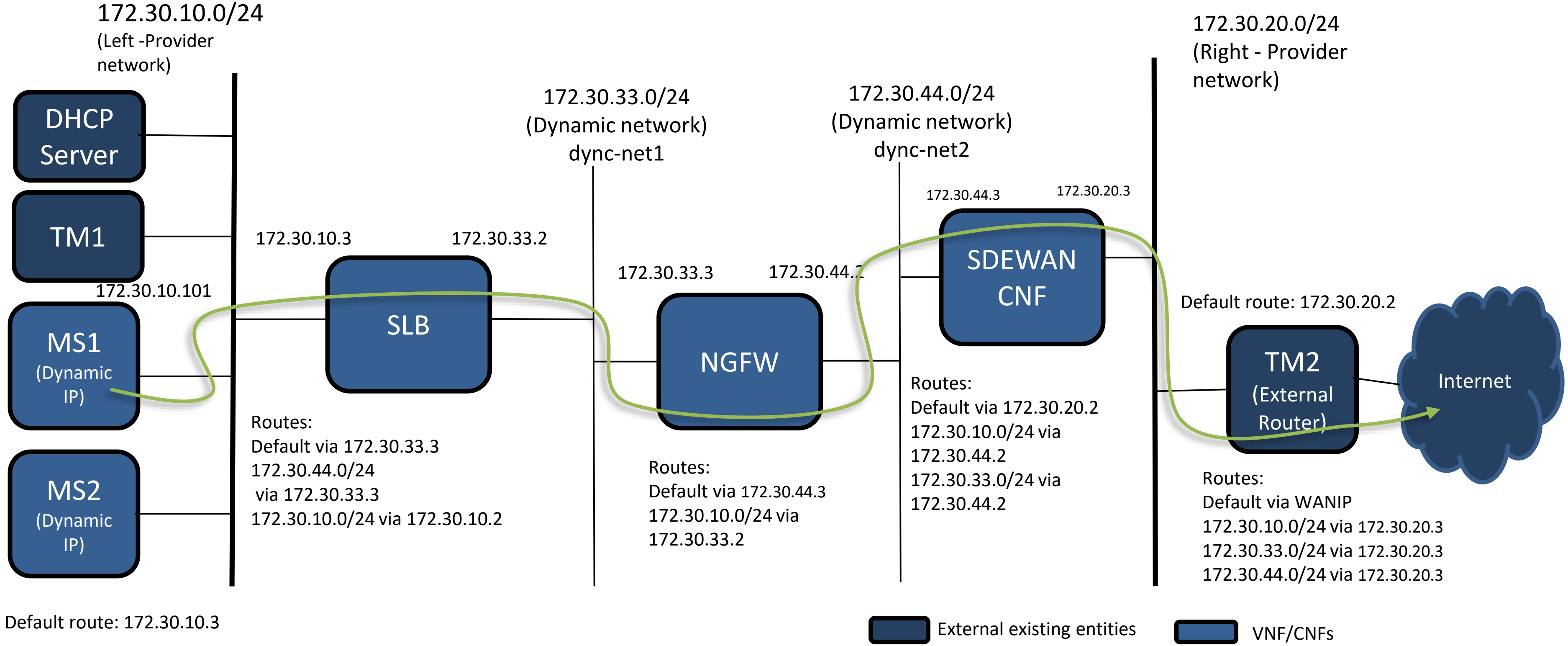
Routes:
Default via 172.30.44.3
172.30.10.0/24 via
172.30.33.2

Routes:
Default via 172.30.20.2
172.30.10.0/24 via
172.30.44.2
172.30.33.0/24 via
172.30.44.2

Routes:
Default via WANIP
172.30.10.0/24 via 172.30.20.3
172.30.33.0/24 via 172.30.20.3
172.30.44.0/24 via 172.30.20.3

Default route: 172.30.10.3

Traffic from pod within the cluster with sfc



172.30.10.0/24
(Left -Provider
network)

DHCP
Server

TM1

172.30.10.101

MS1
(Dynamic
IP)

MS2
(Dynamic
IP)

Default route: 172.30.10.3

172.30.10.3

172.30.33.2

SLB

Routes:
Default via 172.30.33.3
172.30.44.0/24
via 172.30.33.3
172.30.10.0/24 via 172.30.10.2

172.30.33.0/24
(Dynamic network)
dync-net1

172.30.33.3

172.30.44.2

NGFW

Routes:
Default via 172.30.44.3
172.30.10.0/24 via
172.30.33.2

172.30.44.0/24
(Dynamic network)
dync-net2

172.30.44.3

172.30.20.3

SDEWAN
CNF

Routes:
Default via 172.30.20.2
172.30.10.0/24 via
172.30.44.2
172.30.33.0/24 via
172.30.44.2

172.30.20.0/24
(Right - Provider
network)

Default route: 172.30.20.2

TM2
(External
Router)

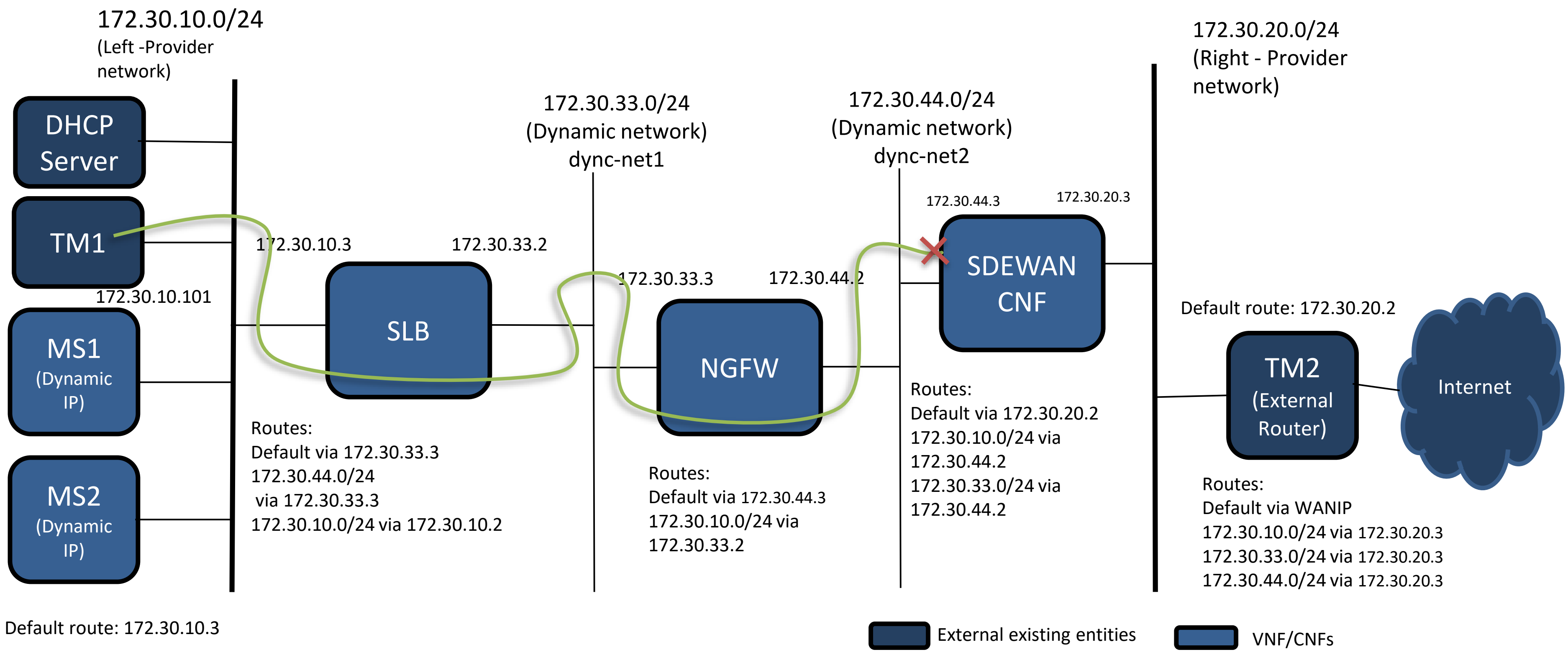
Internet

Routes:
Default via WANIP
172.30.10.0/24 via 172.30.20.3
172.30.33.0/24 via 172.30.20.3
172.30.44.0/24 via 172.30.20.3

■ External existing entities

■ VNF/CNFs

Traffic from external entities – Firewall icmp reject



172.30.10.0/24
(Left -Provider
network)

172.30.33.0/24
(Dynamic network)
dync-net1

172.30.44.0/24
(Dynamic network)
dync-net2

172.30.20.0/24
(Right - Provider
network)

DHCP Server

TM1

172.30.10.101

MS1
(Dynamic IP)

MS2
(Dynamic IP)

Default route: 172.30.10.3

SLB

172.30.10.3

172.30.33.2

Routes:
Default via 172.30.33.3
172.30.44.0/24
via 172.30.33.3
172.30.10.0/24 via 172.30.10.2

NGFW

172.30.33.3

172.30.44.2

Routes:
Default via 172.30.44.3
172.30.10.0/24 via
172.30.33.2

SDEWAN CNF

172.30.44.3

172.30.20.3

Routes:
Default via 172.30.20.2
172.30.10.0/24 via
172.30.44.2
172.30.33.0/24 via
172.30.44.2

Default route: 172.30.20.2

TM2
(External Router)

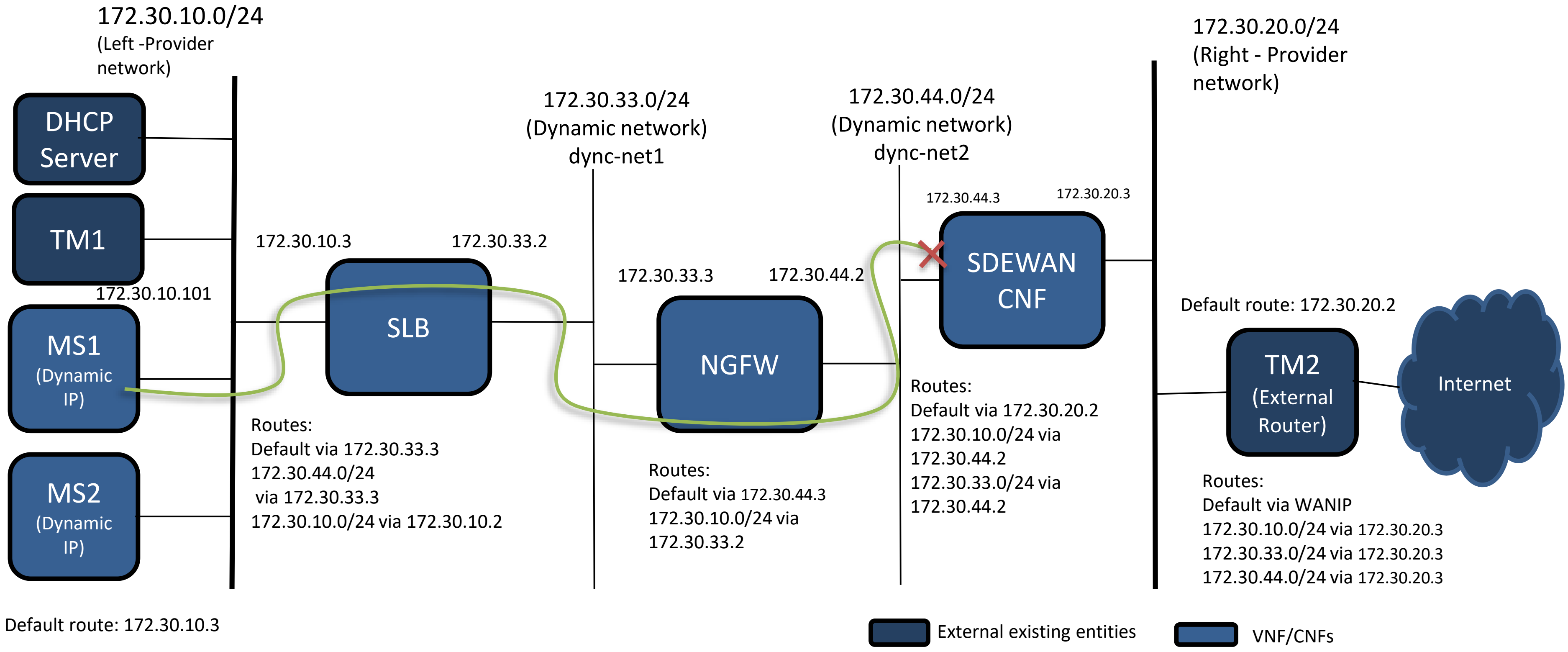
Internet

Routes:
Default via WANIP
172.30.10.0/24 via 172.30.20.3
172.30.33.0/24 via 172.30.20.3
172.30.44.0/24 via 172.30.20.3

■ External existing entities

■ VNF/CNFs

Traffic from pod within the cluster – Firewall icmp reject



172.30.10.0/24
(Left -Provider
network)

172.30.33.0/24
(Dynamic network)
dync-net1

172.30.44.0/24
(Dynamic network)
dync-net2

172.30.20.0/24
(Right - Provider
network)

DHCP Server

TM1

172.30.10.101

MS1
(Dynamic IP)

MS2
(Dynamic IP)

Default route: 172.30.10.3

172.30.10.3

172.30.33.2

SLB

Routes:
Default via 172.30.33.3
172.30.44.0/24
via 172.30.33.3
172.30.10.0/24 via 172.30.10.2

172.30.33.3

172.30.44.2

NGFW

Routes:
Default via 172.30.44.3
172.30.10.0/24 via
172.30.33.2

172.30.44.3

172.30.20.3

SDEWAN CNF

Routes:
Default via 172.30.20.2
172.30.10.0/24 via
172.30.44.2
172.30.33.0/24 via
172.30.44.2

Default route: 172.30.20.2

TM2
(External
Router)

Internet

Routes:
Default via WANIP
172.30.10.0/24 via 172.30.20.3
172.30.33.0/24 via 172.30.20.3
172.30.44.0/24 via 172.30.20.3

▬ External existing entities

▬ VNF/CNFs

OVN4NFV Status

Current

- Dynamic Network Creation
- VLAN Provider Network Support – Controller and Agent
- Direct Provider Network Support – Controller and Agent
- SFC feature – Controller and Agent
- Kubespray default primary network plugin
- Tested with sdewan CNFs and SDEWAN Controller
- Tested with vDPA and OVS offloading as a PoC

Link to Repo:
https://github.com/opnfv/ovn4nfv-k8s-plugin
https://github.com/kubernetes-sigs/kubespray/blob/master/docs/ovn4nfv.md

Upcoming features in OVN4NFV

Work In Progress

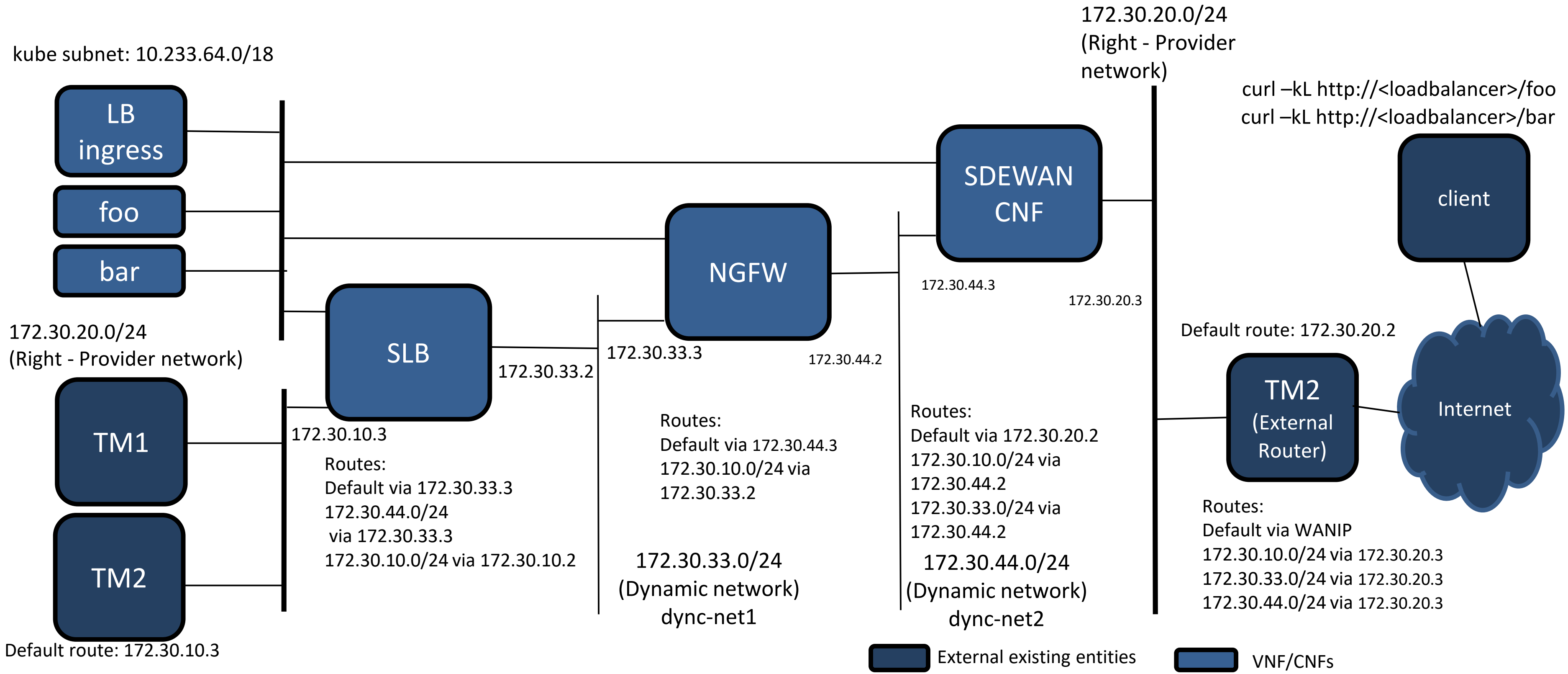
- Kubespray Centos CI/CD, SFC advance testing
- Using OVN Load balancer for Kubernetes service(without kube-proxy)
- SRIOV NIC as primary network interfaces
- SFC support with OVN load balancer support for NF Elasticity
- Network policy with OVS
- Proxy less service mesh with OVN & Ipsec in network namespace
- IPv6 support
- Traffic interception method with 5G UPF

OVN4NFV – SFC

By:
Kuralamudhan Ramakrishnan(kuralamudhan.ramakrishnan@intel.com) | August 11th, 2020

Acknowledgement:
Srinivasa Addepalli, Ritu Sood

Test scenario – to comprehend multiple deployment variations



kube subnet: 10.233.64.0/18

LB ingress

foo

bar

172.30.20.0/24
(Right - Provider network)

TM1

TM2

Default route: 172.30.10.3

SLB

172.30.10.3

Routes:
Default via 172.30.33.3
172.30.44.0/24
via 172.30.33.3
172.30.10.0/24 via 172.30.10.2

172.30.33.2

172.30.33.3

NGFW

172.30.44.2

Routes:
Default via 172.30.44.3
172.30.10.0/24 via
172.30.33.2

172.30.33.0/24
(Dynamic network)
dync-net1

SDEWAN CNF

172.30.44.3

172.30.20.3

Routes:
Default via 172.30.20.2
172.30.10.0/24 via
172.30.44.2
172.30.33.0/24 via
172.30.44.2

172.30.44.0/24
(Dynamic network)
dync-net2

172.30.20.0/24
(Right - Provider network)

curl –kL http://<loadbalancer>/foo
curl –kL http://<loadbalancer>/bar

client

Default route: 172.30.20.2

TM2 (External Router)

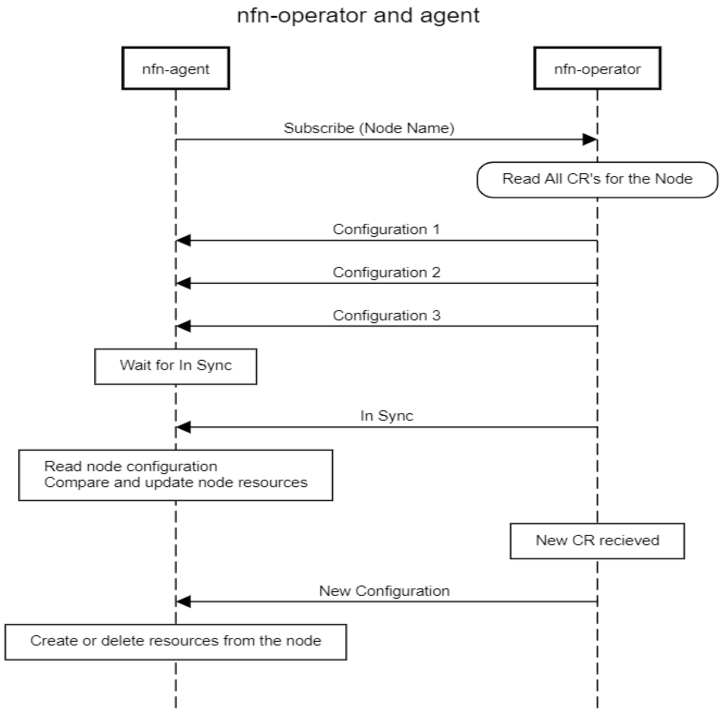
Internet

Routes:
Default via WANIP
172.30.10.0/24 via 172.30.20.3
172.30.33.0/24 via 172.30.20.3
172.30.44.0/24 via 172.30.20.3

■ External existing entities

■ VNF/CNFs

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

Open vSwitch

OVN4NFV – SFC

By:
Kuralamudhan Ramakrishnan(kuralamudhan.ramakrishnan@intel.com) | August 11th, 2020

Acknowledgement:
Srinivasa Addepalli, Ritu Sood