Application/Service orchestration on K8s based sites

A foundation for Multi Edge & Cloud Orchestration

Srinivasa Addepalli

Contact: Srinivasa.r.addepalli@intel.com

Agenda

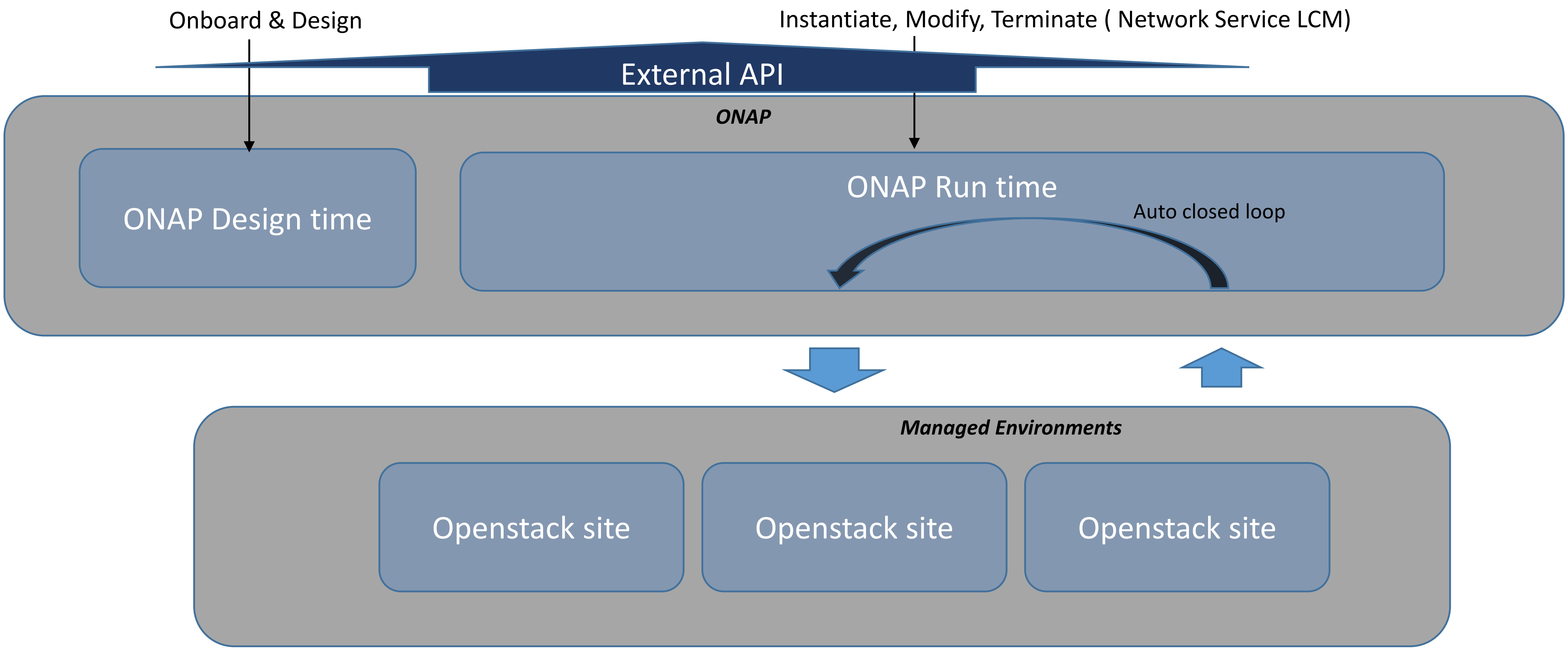
ONAP introduction

ONAP R4/R5 – K8s support

ONAP R6/R7 - Roadmap

Details

ONAP Overview

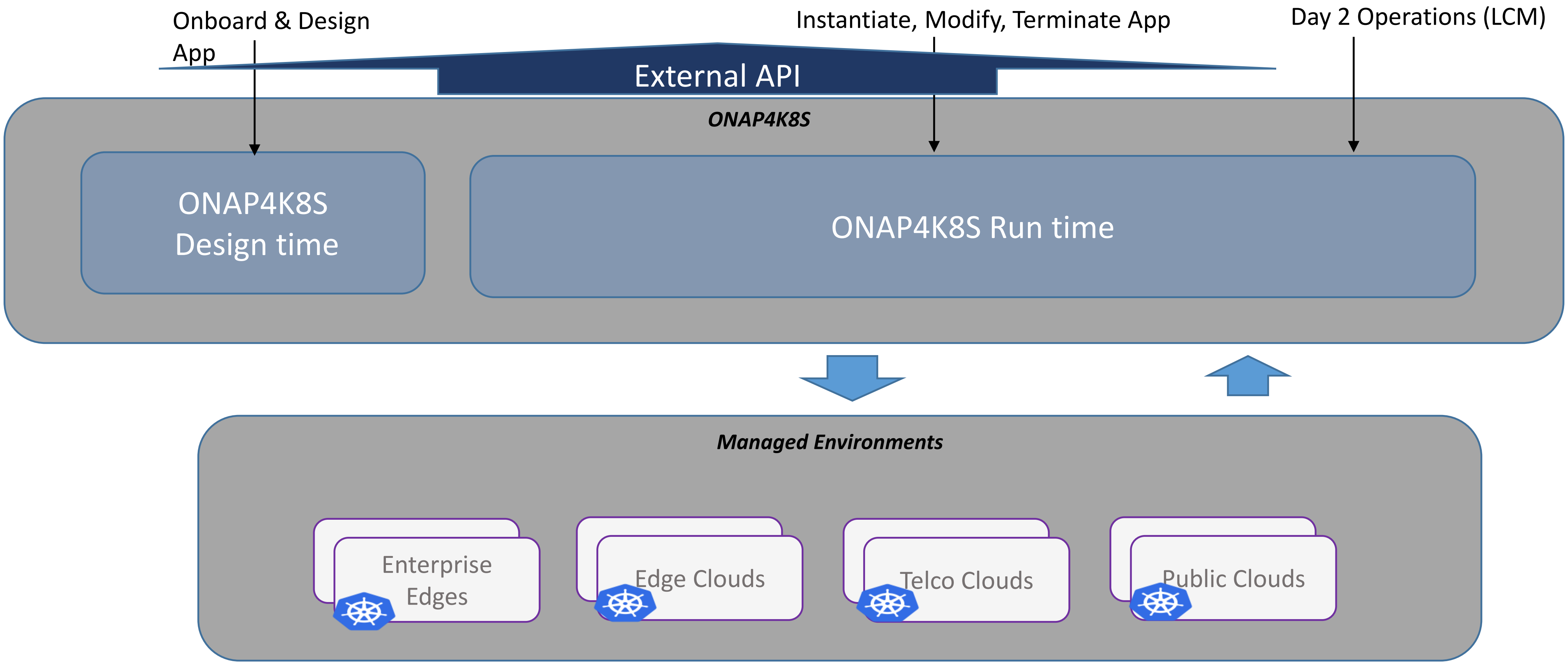


Automation platform for managing network services and VNFs across Openstack sites

ONAP - Intel Journey and contributions

R1 – Amsterdam	R2 – Beijing	R3 – Casablanca	R4 – Dublin	R5 –El Alto
Successful merge of ECOMP and Open-O in ONAP Established Structure Validated with vFirewall, vDNS, vCPE, VoLTE	Containerized S3P introduced **HPA introduced** Change Management & Scaling foundations	5G work started CCVPN use case Increased standard alignment Started Control loop subcommittee **HPA Matured** **CA key protection using PKCS11/TPM**	**Support for Kubernetes based sites introduced : Support for VNFs and CNFs on Kubernetes sites – vFirewall and EdgeXFoundry use cases** Footprint optimizations Model driven closed loop **Introduction of vIPSEC use case**	Reduce technical debt Security by design **ONAP4K8s (standalone)** **ISTIO security for ONAP4K8s**

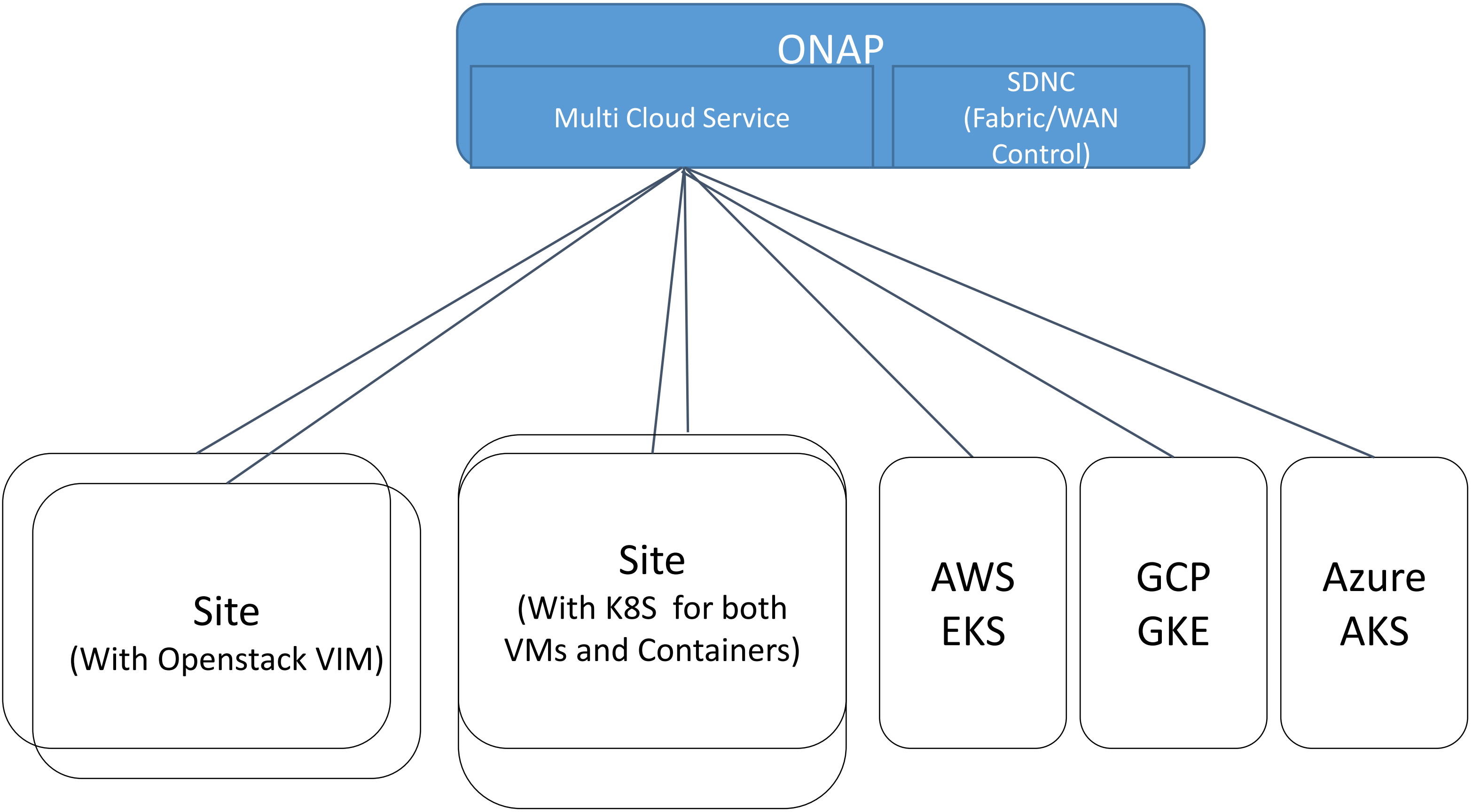
ONAP4K8S – R5 Review



A platform for managing both applications and network functions across edges and clouds

CNF/VNF support via K8s in ONAP R4/R5

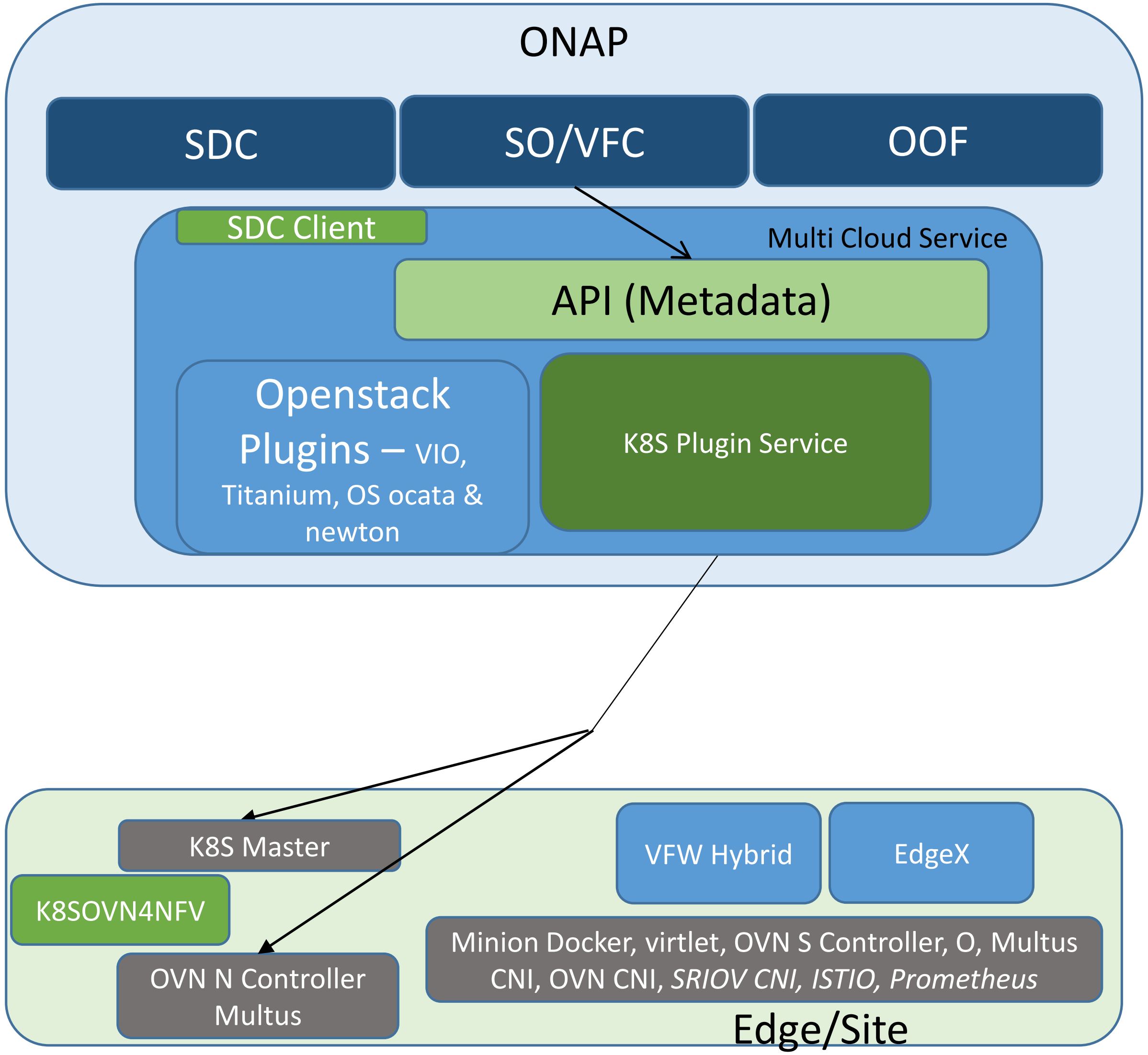
ONAP – Support for K8S based Sites


ONAP
Multi Cloud Service
SDNC (Fabric/WAN Control)
Site (With Openstack VIM)
Site (With K8S for both VMs and Containers)
AWS EKS
GCP GKE
Azure AKS

- Current support as in R3: Openstack based remote Clouds, Support multiple Openstack variations – Windriver Titanium, VMWare VIO, Native Newton, Ocata. Only VM based VNFs.

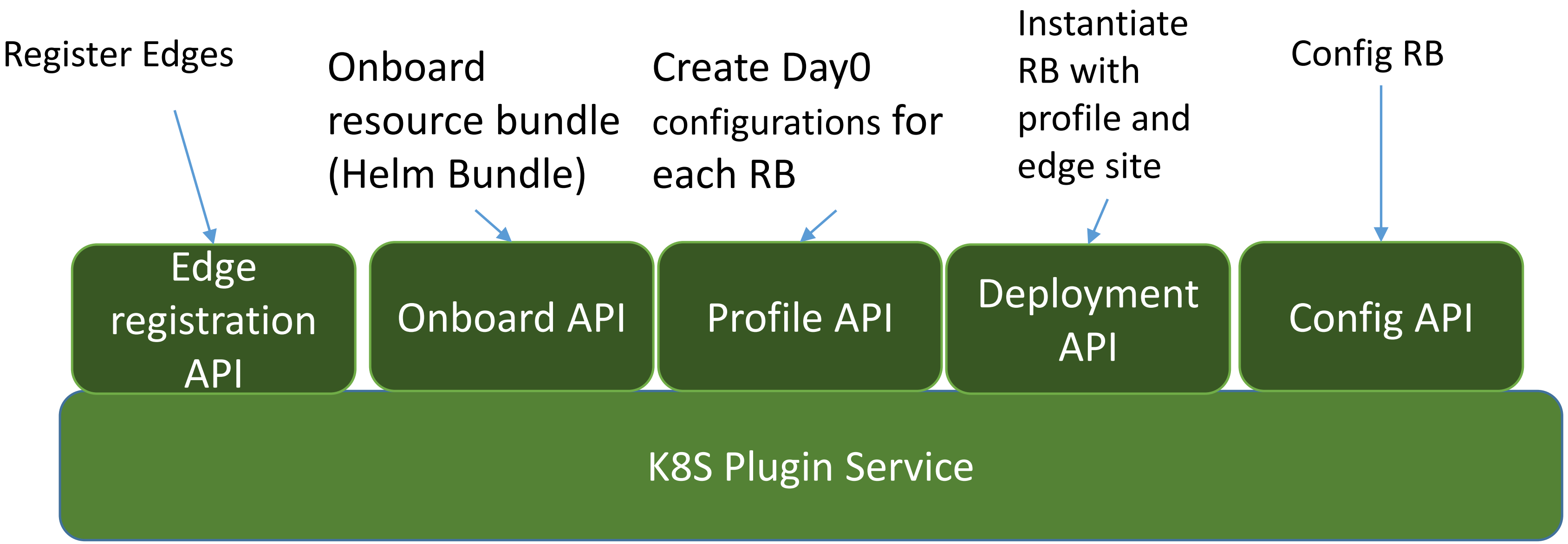
- Goals for R4
  - Support containerized workloads
  - Support containerized VNFs
  - Support both VMs and containers on same compute nodes. (Bare-metal deployment)
  - Support for multiple virtual networks
  - Support for dynamic creation of Virtual networks
  - Support public cloud CaaS such as AWS EKS, GCP GKE and Azure AKS (Only containers, not VMs)

ONAP – K8S Support

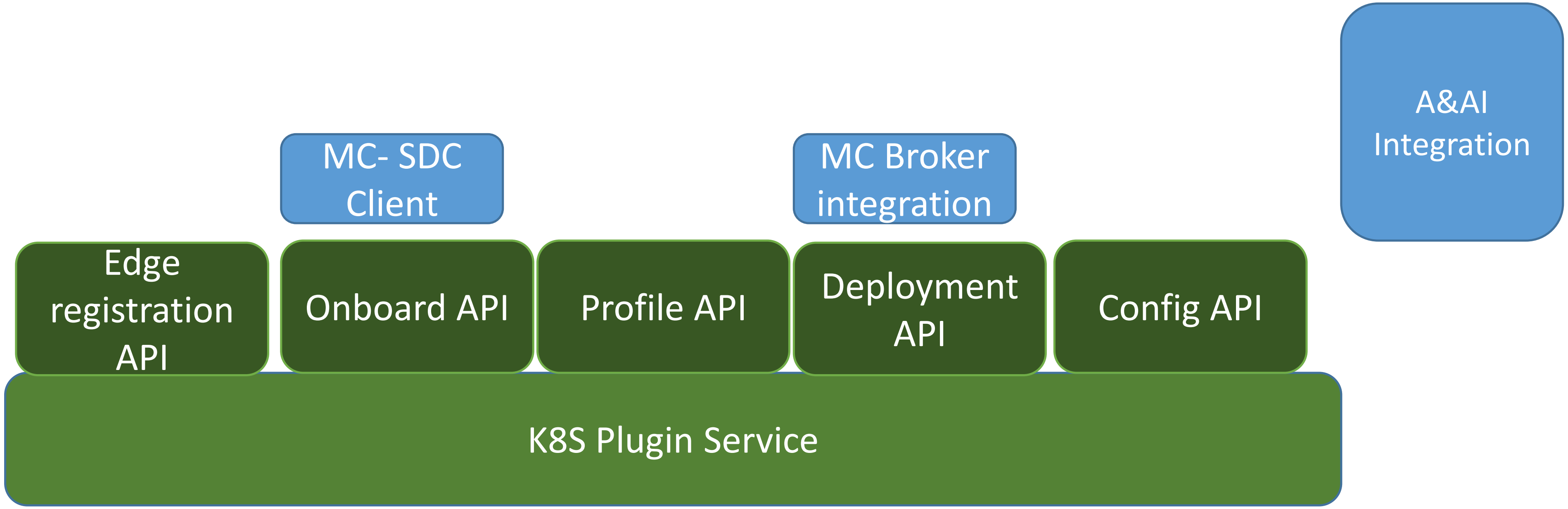

ONAP
SDC
SO/VFC
OOF
SDC Client
Multi Cloud Service
API (Metadata)
Openstack Plugins – VIO, Titanium, OS ocata & newton
K8S Plugin Service
K8S Master
K8SOVN4NFV
OVN N Controller Multus
VFW Hybrid
EdgeX
Minion Docker, virtlet, OVN S Controller, O, Multus CNI, OVN CNI, *SRIOV CNI, ISTIO, Prometheus*
Edge/Site

1. Uniform API across cloud technologies (HEAT, K8S, Azure etc..)
2. K8S Multi-Cloud Service plugin
   - Support for deployment and services.
   - K8S yaml artifacts
   - Networking – OVN, flannel and Multus
   - Mongo DB for storing config/RBs, etcd for Day 2 configuration
3. Kubernetes Deployment (KuD)
   - Installation of software & configuration to make K8S based sites.
   - Additional of virtlet, Multus, OVN and flannel.
4. OVN-for-K8s-NFV (OPNFV project, visualized as part of ONAP work)
   - Support for multiple virtual networks
   - Support for dynamic creation/deletion of virtual networks
5. ONAP Integration
   - SDC for onboarding VNF/App with Helm artifacts
   - Distribution of Helm artifacts to MC.
   - SO based instantiation
   - Two modes - Self contained and with rest of ONAP

K8S Plugin Service as independent manager – Modular design


Register Edges
Onboard resource bundle (Helm Bundle)
Create Day0 configurations for each RB
Instantiate RB with profile and edge site
Config RB
Edge registration API
Onboard API
Profile API
Deployment API
Config API
K8S Plugin Service

K8S Plugin Service with rest of ONAP



MC- SDC Client

MC Broker integration

A&AI Integration

Edge registration API

Onboard API

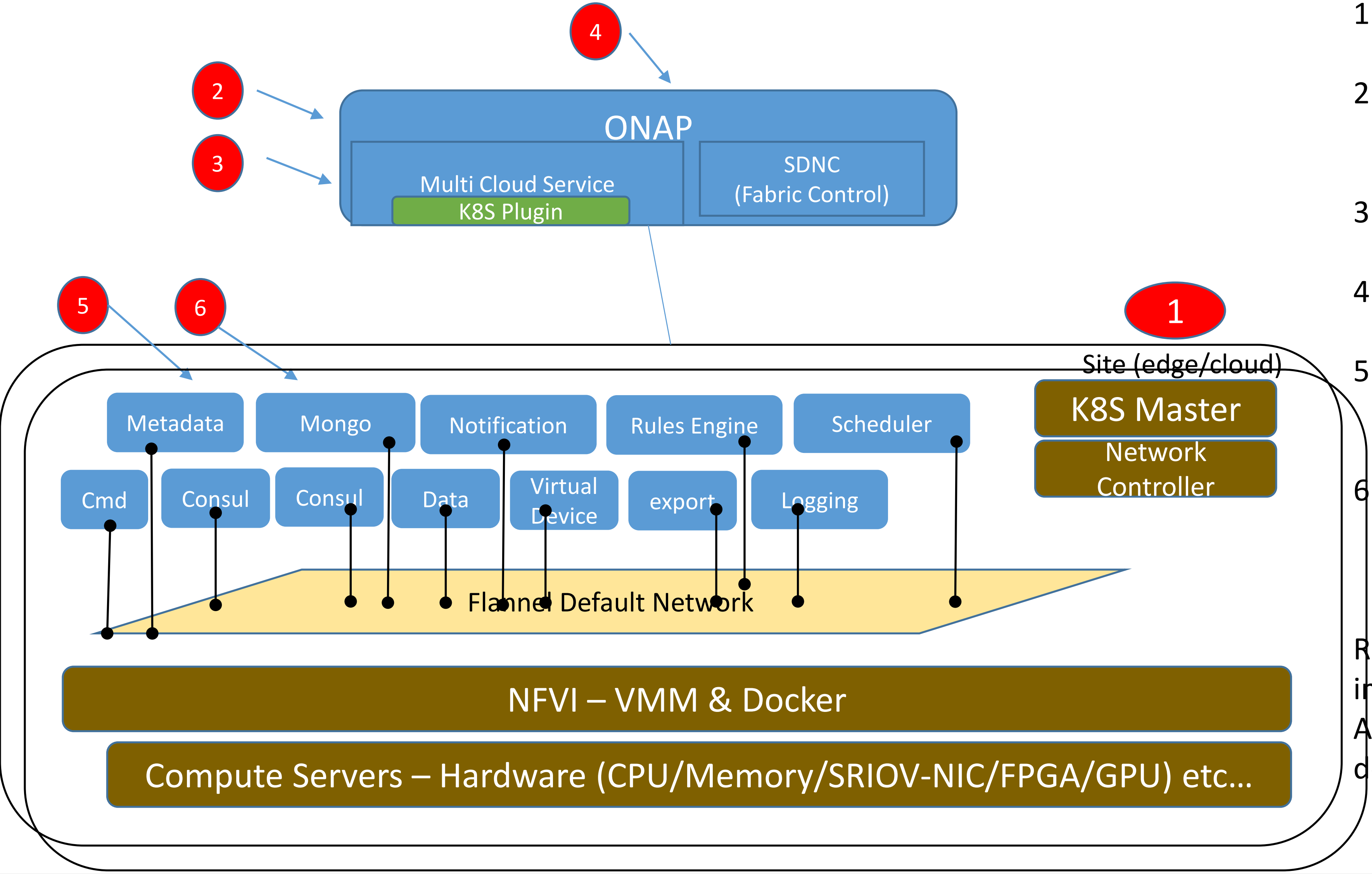
Profile API

Deployment API

Config API

K8S Plugin Service

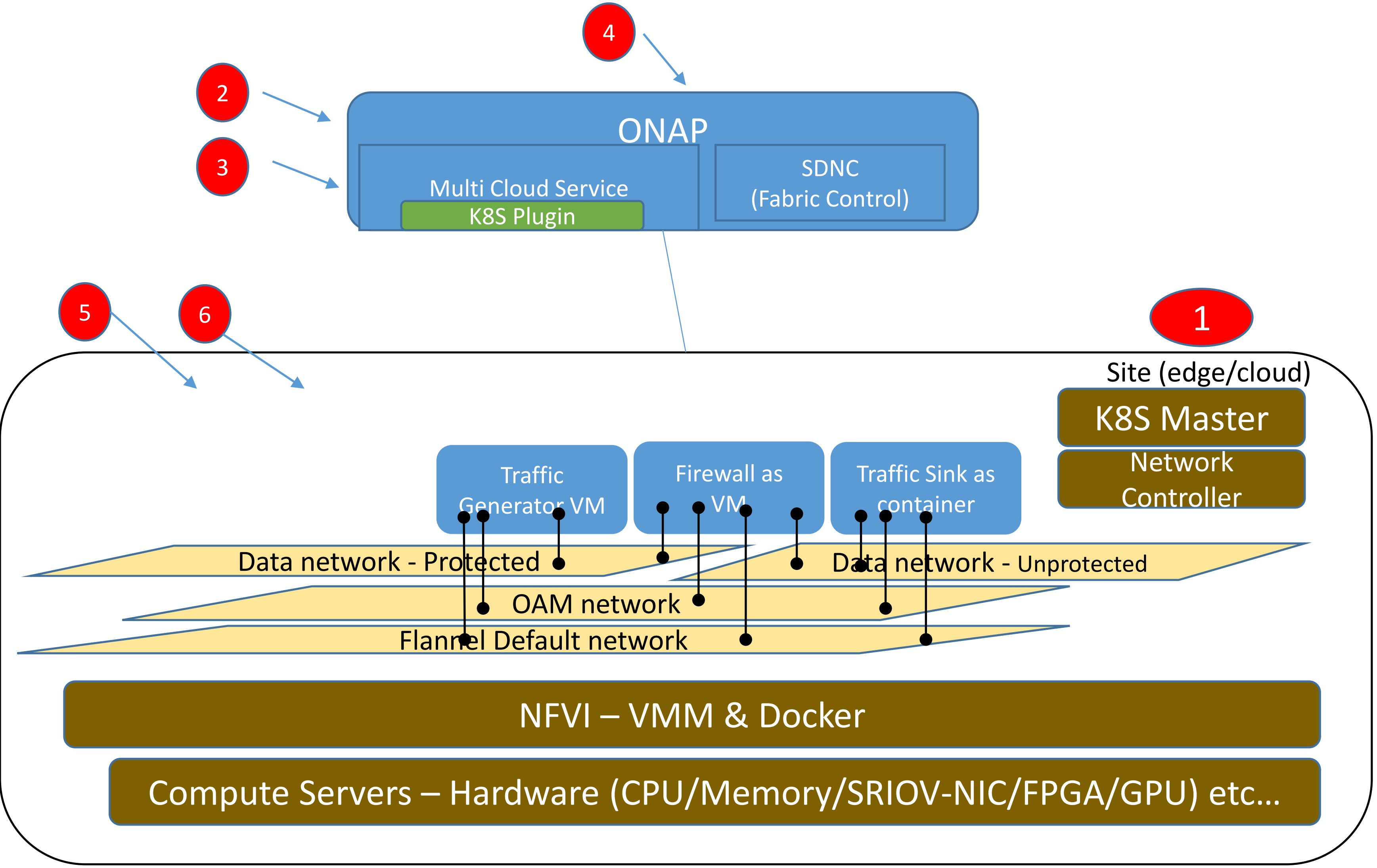
R4 Scenarios – EdgeX deployment



1. One time: Prepare K8S based site using KUD (if it does not exist)
2. One time: Register the K8S Site in ONAP by adding Kubeconfig file in ONAP (if the site is not added earilier)
3. EdgeX onboarding: EdgeX deployment and service helm charts in SDC
4. Instantiate EdgeX (by calling SO API) via postman or via VID GUI
5. Check if all EdgeX containers are successful brought up on the site (using K8S utilities on the site)
6. Basic EdgeX testing to ensure that functionality also works
   - Use consul dashboard to check the services and their status

Repeat step 4 to 6 by bringing second instance of EdgeX on a different namespace. Also, work with Edgex team to automate deployment verification

vFirewall scenario (as VMs and containers – Hybrid)

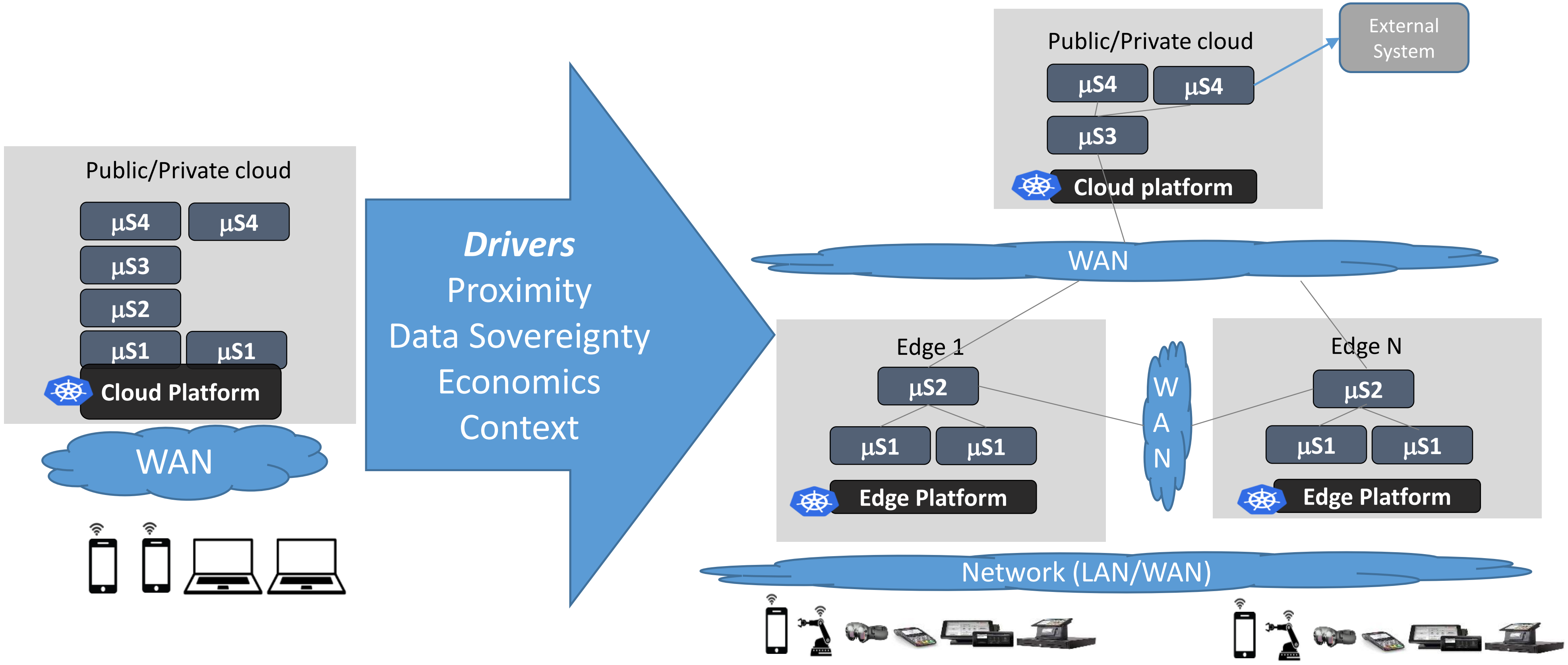


1. One time: Prepare K8S based site using KRD (if it does not exist)
2. One time: Register the K8S Site in ONAP by adding Kubeconfig file in ONAP (if the site is not added earilier)
3. vFirewall onboarding: Create deployment and service yaml
4. Instantiate vFirewall using SO API (or VID GUI)
5. Check if firewall is successfully brought up on the site (using tools) and also ensure that three additional virtual networks are created. Also ensure that firewall belongs in all data networks. Ensure that generator and sink belong to different data networks.
6. Basic firewall testing to ensure that functionality also works
   - Check the sink dashboard to ensure that right packet streams are received by sink.

ONAP
OPEN NETWORK AUTOMATION PLATFORM

ONAP R6 and R7

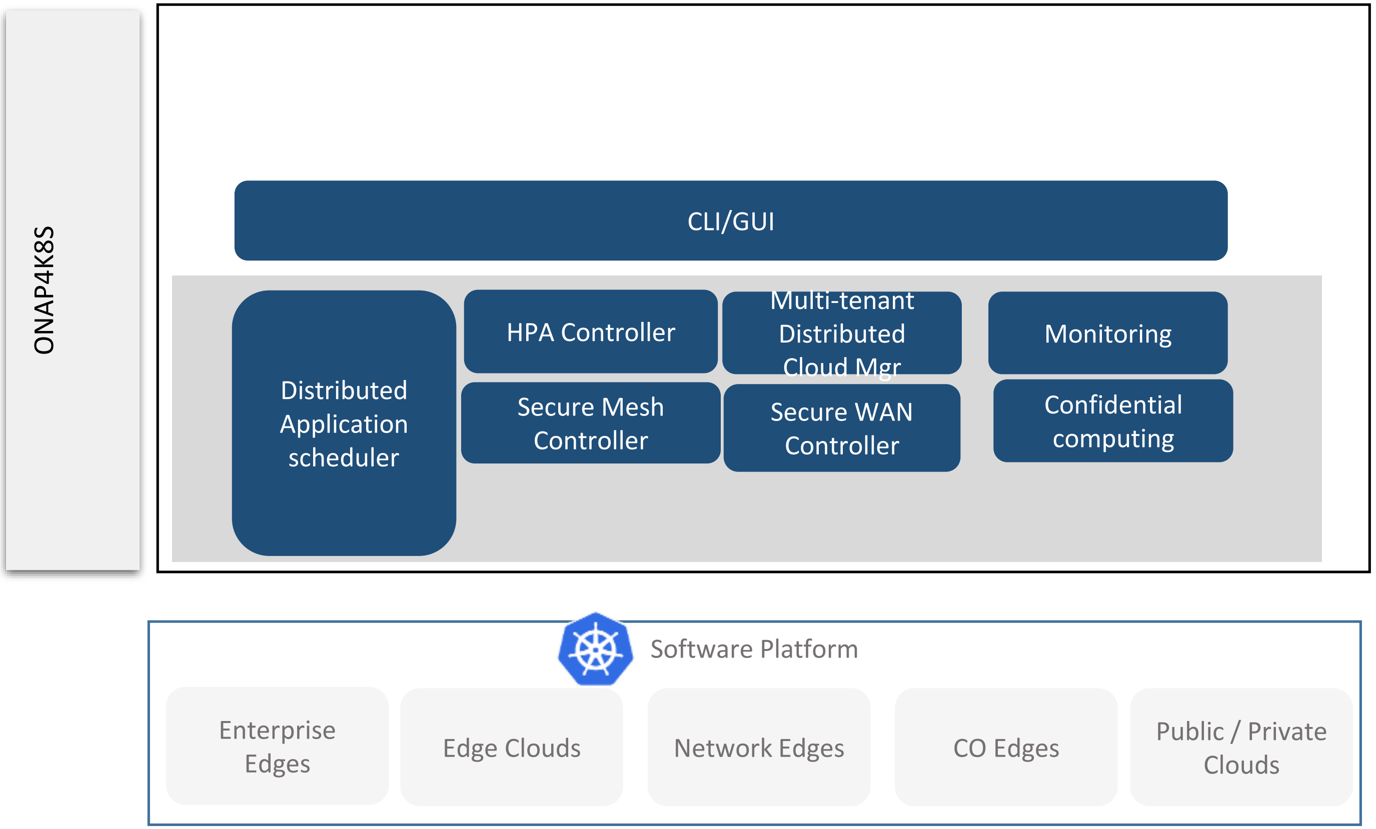
Towards ONAP4K8S R6 and beyond
Application Transformation – Centralized to Geo-Distributed



requires >20 manual operations on each edge. Think about the effort with multiple edges!!!!

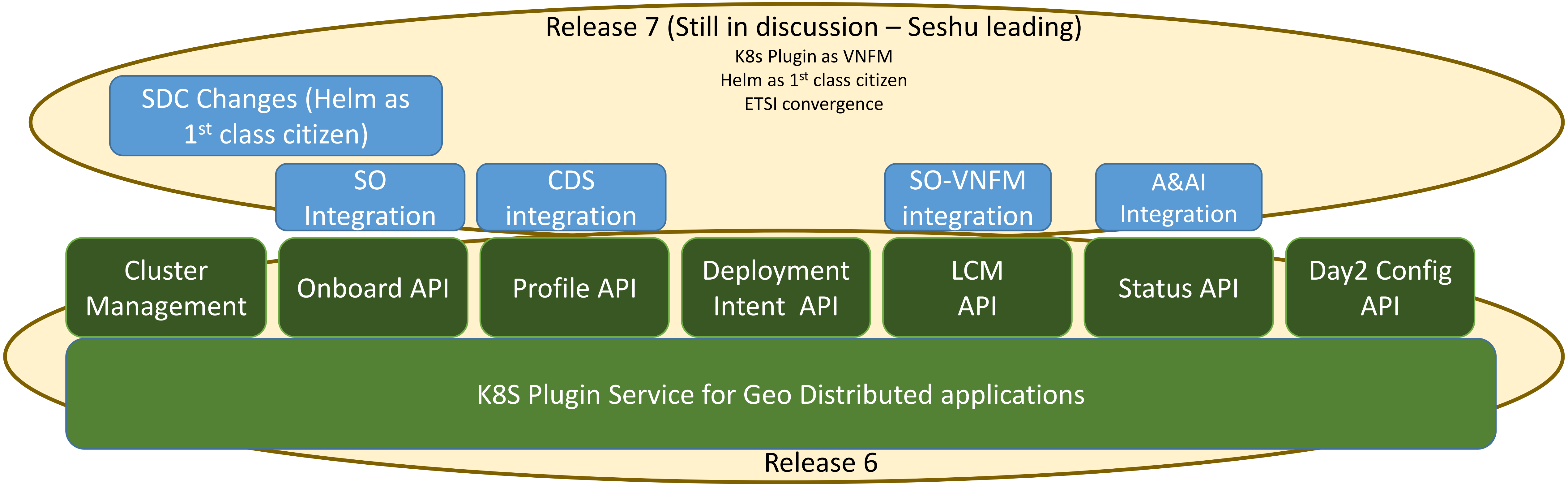
Need for Multi Edge/Cloud Orchestrator

Solution ingredient: ONAP4K8S for Distributed Applications



- Generic and flexible distributed **application scheduler** with extensible placement and action controllers
- **Hardware Platform Aware** scheduling with Auto discovery of platform capabilities
- Auto configuration of **service meshes (e.g., ISTIO) and security policies** (e.g. firewall & NAT) with workload LCM.
- **Secure WAN** cloud across edge groups
- Protect Application IP via confidential computing
- Monitor distributed application performance, accesses

R6 and R7 Plan



Release 7 (Still in discussion – Seshu leading)

K8s Plugin as VNFM

Helm as 1st class citizen

ETSI convergence

SDC Changes (Helm as 1st class citizen)

SO Integration

CDS integration

SO-VNFM integration

A&AI Integration

Cluster Management

Onboard API

Profile API

Deployment Intent API

LCM API

Status API

Day2 Config API

K8S Plugin Service for Geo Distributed applications

Release 6

Details

What is ONAP4K8S?

Is Multi Site Orchestrator

Independent package by itself (Being done in ONAP project, but independent of rest of ONAP)

Targeting Enterprise, MSP, IOT markets

Supporting deployment of both applications and network functions

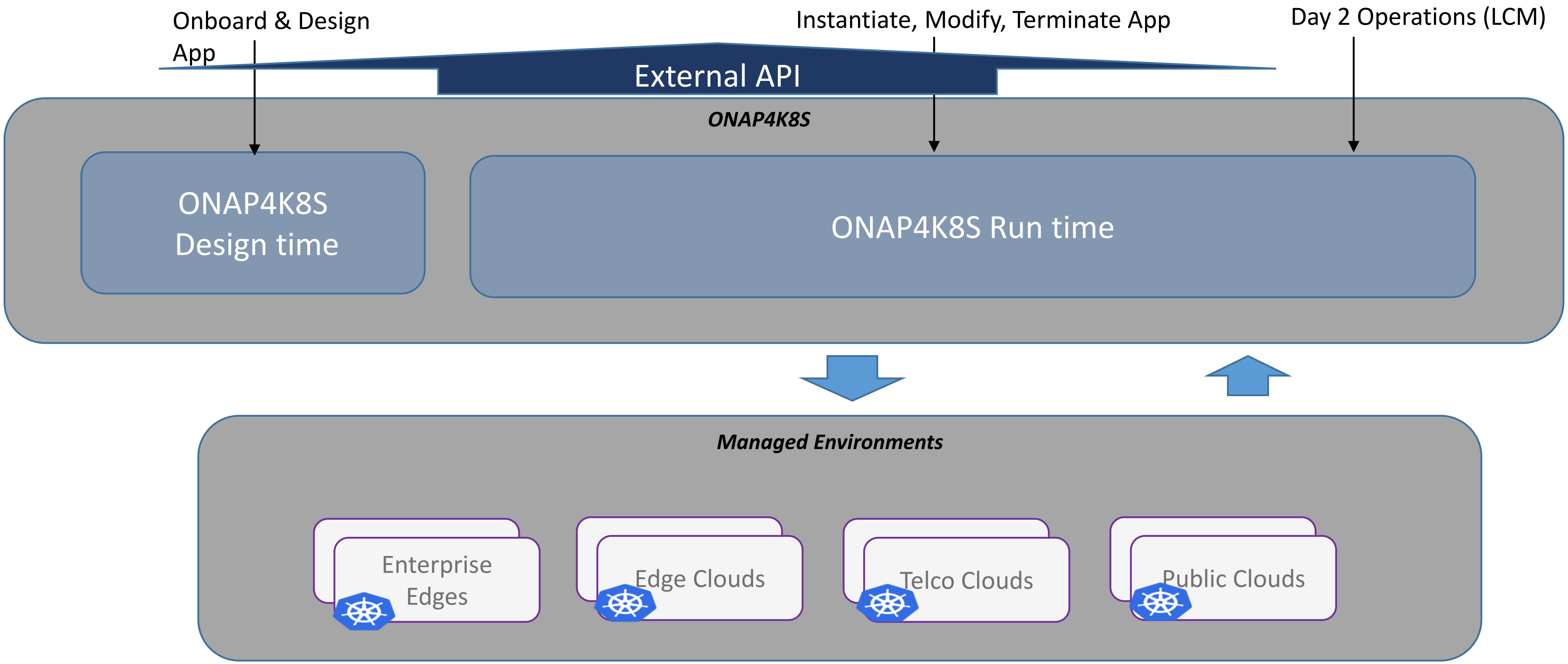
Supporting workload types - VMs, Containers, VNFs and CNFs

Lightweight & high performance

Micro-service based architecture

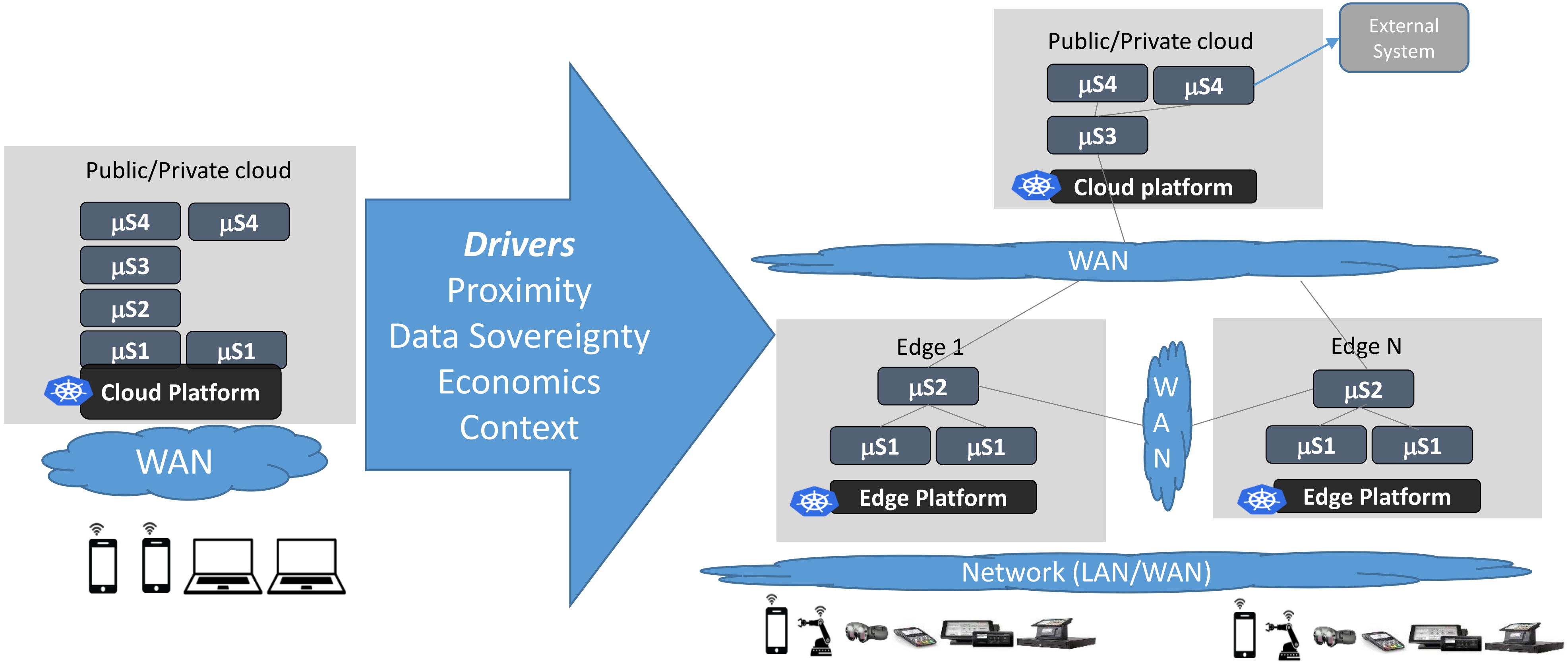
Though it is being done in ONAP community, the entire code for ONAP4K8s is developed from scratch and does not need to include legacy ONAP if the remote sites are K8s based.

ONAP4K8S – R5 Review



A platform for managing both applications and network functions across edges and clouds

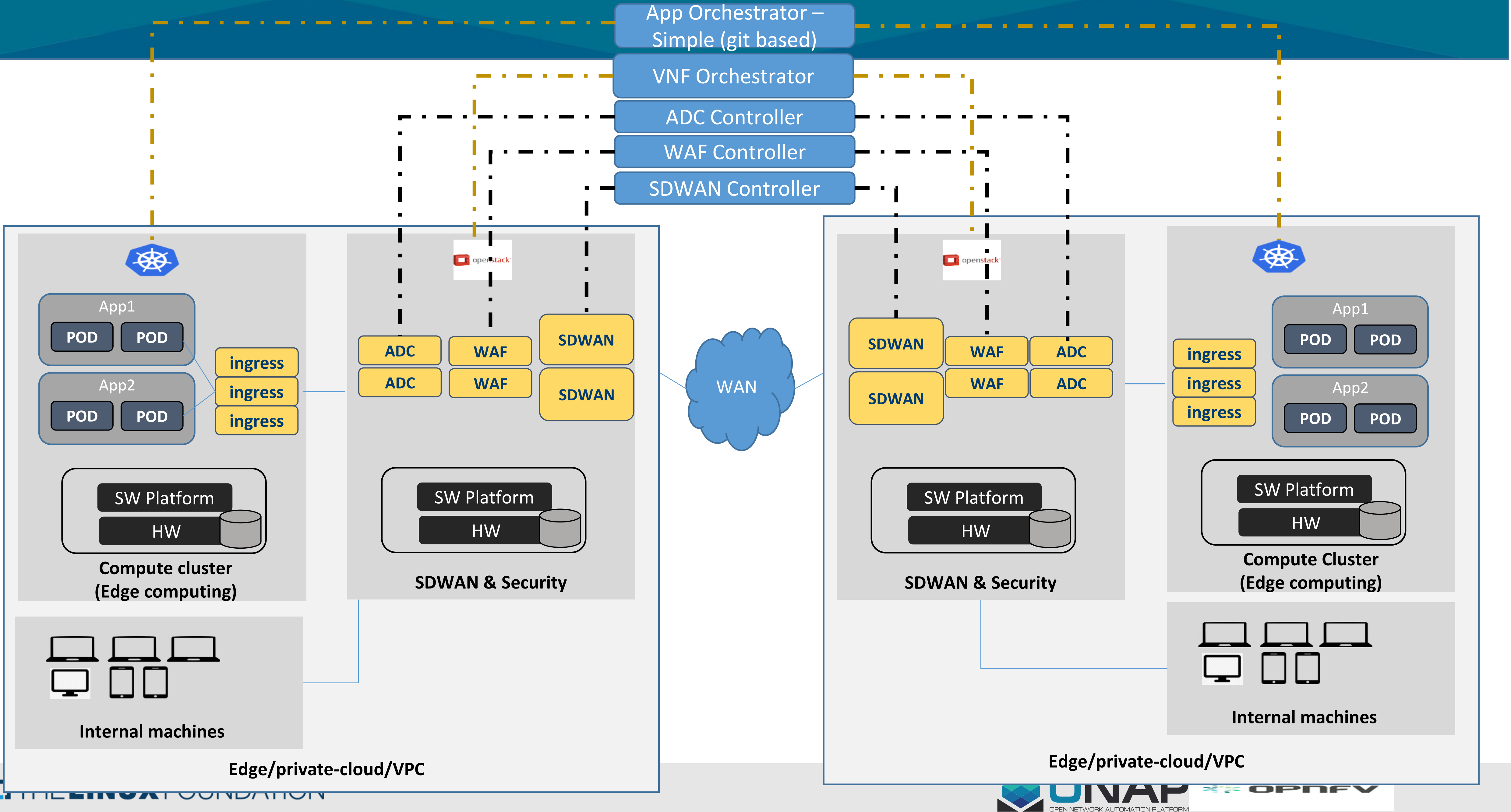
Towards ONAP4K8S R6 and beyond
Application Transformation – Centralized to Geo-Distributed



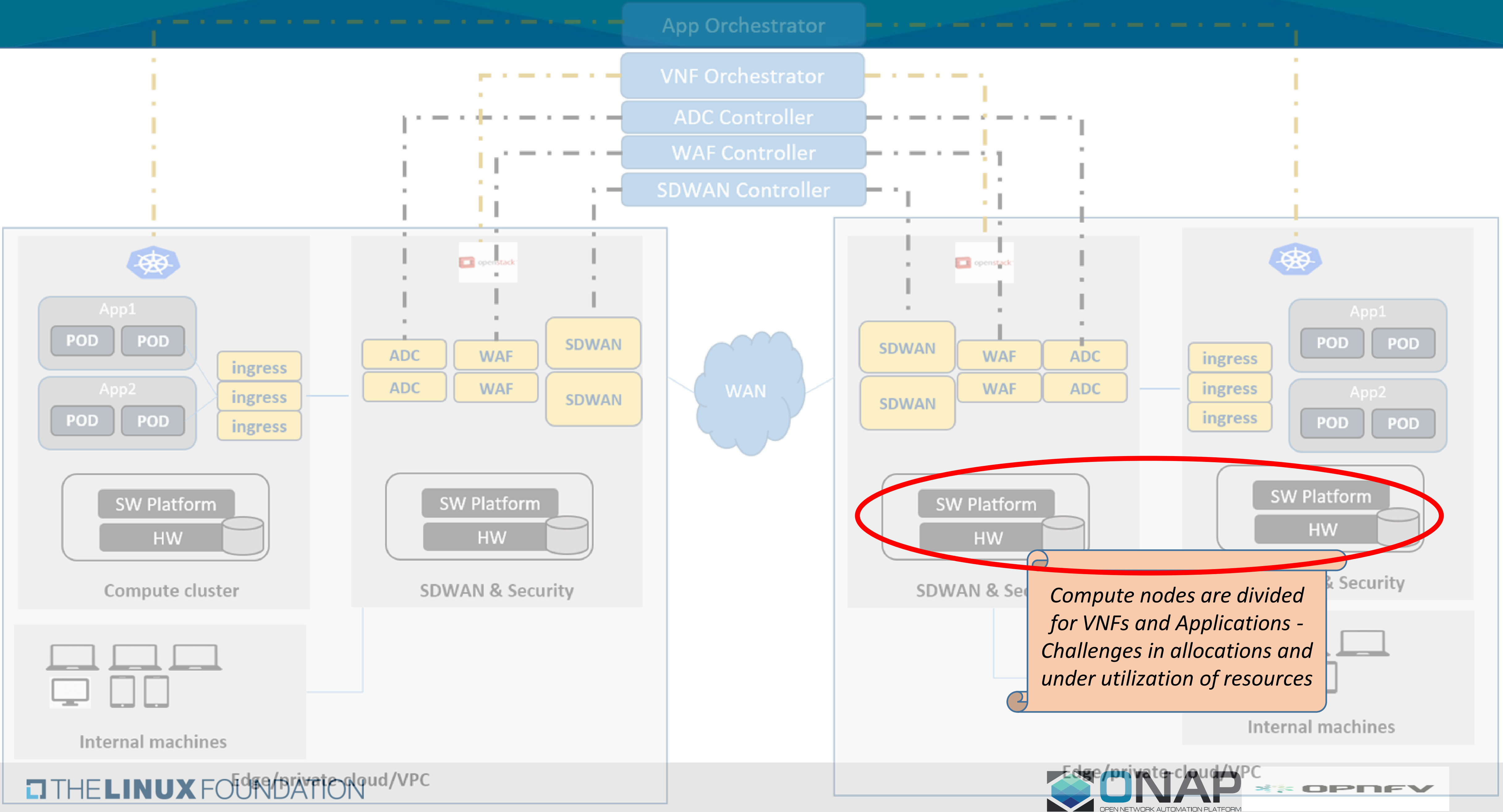
requires >20 manual operations on each edge. Think about the effort with multiple edges!!!!

Need for Multi Edge/Cloud Orchestrator

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

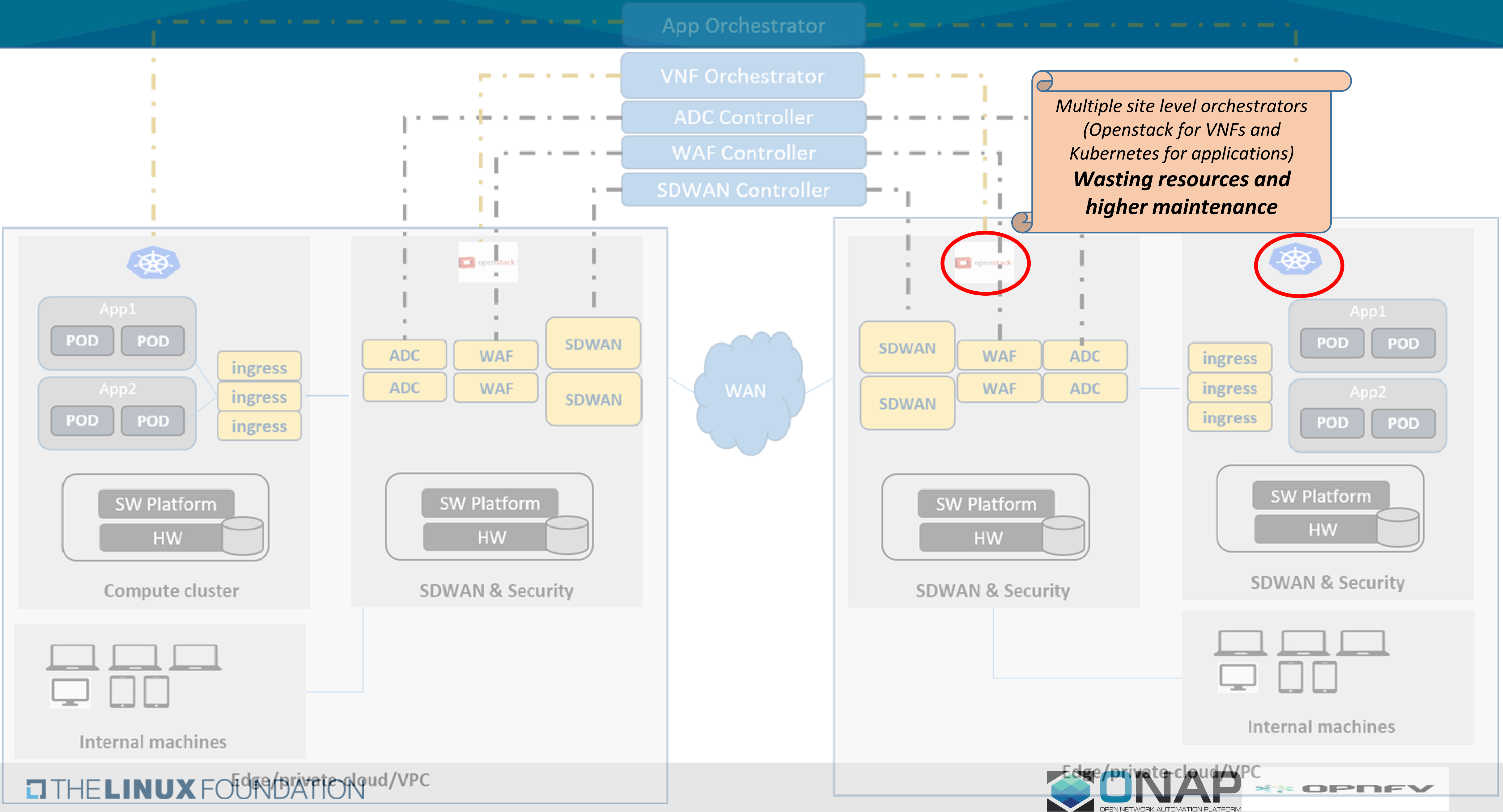


Challenge : Under utilization of resources



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

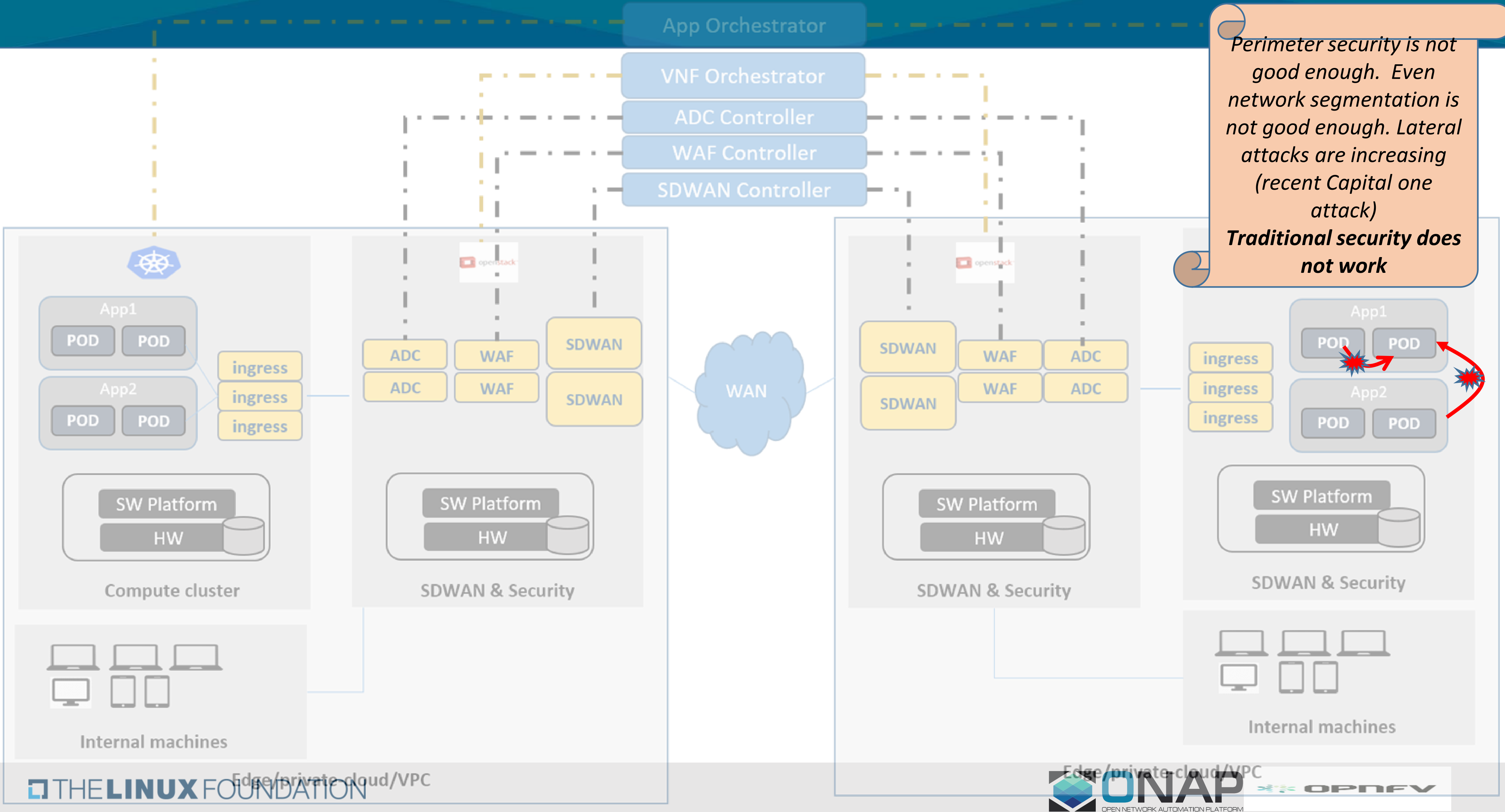
Challenge: Multiple Site level orchestrators leading to wasting of resources



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

Wasting resources and higher maintenance

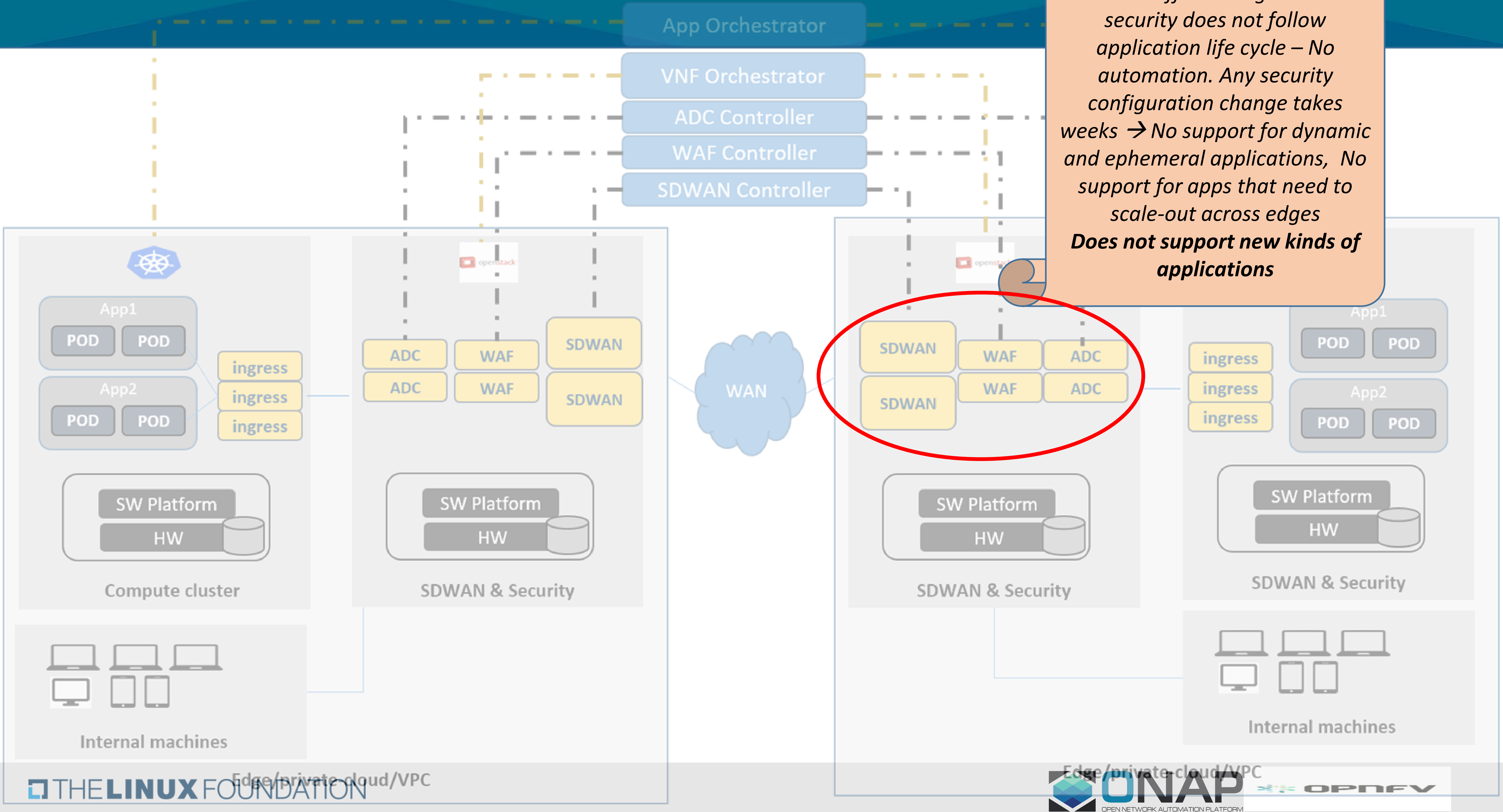
Challenge : Lack of E-W traffic Security



Perimeter security is not good enough. Even network segmentation is not good enough. Lateral attacks are increasing (recent Capital one attack)

Traditional security does not work

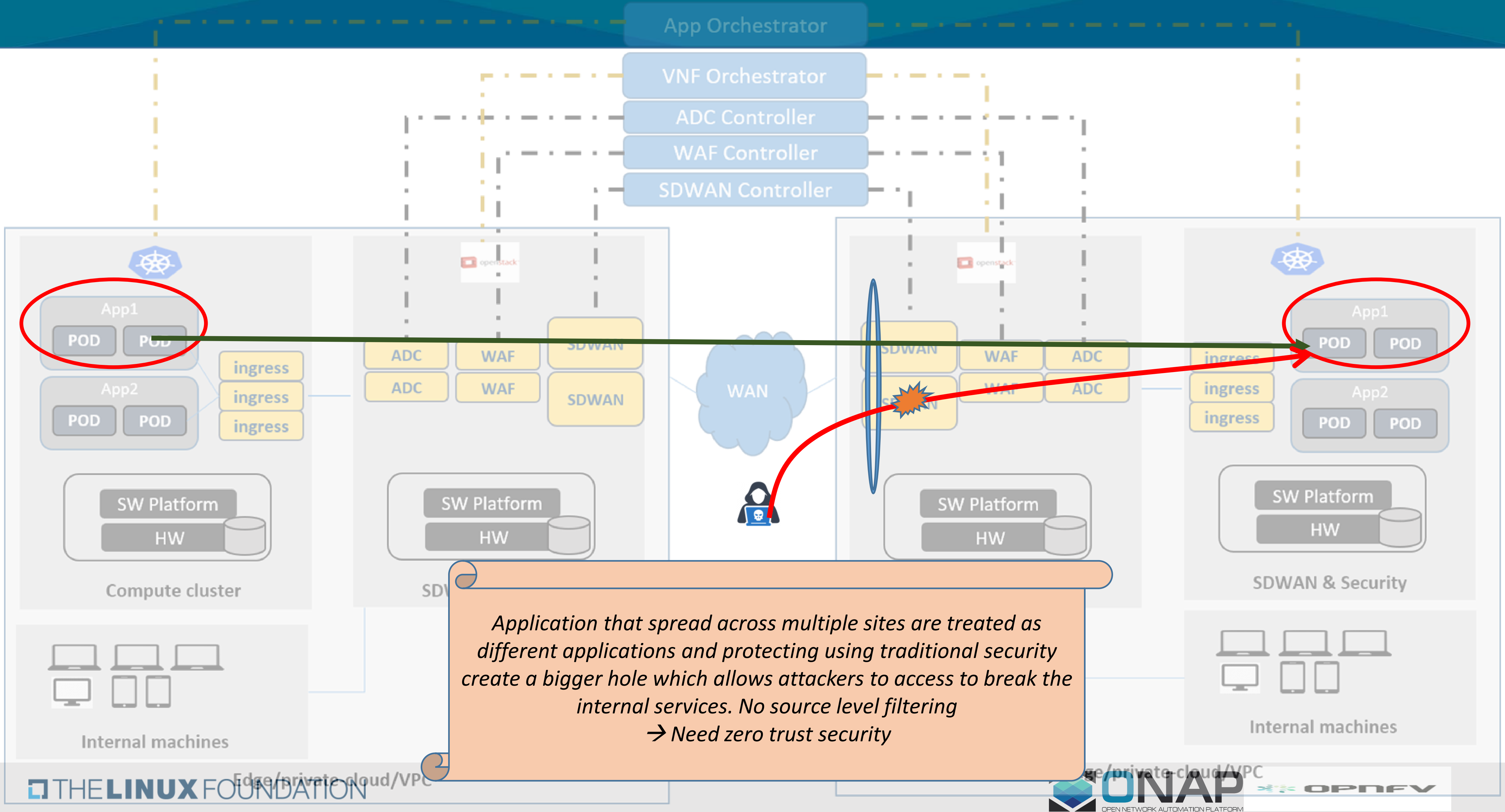
Challenge : Insufficient support for dynamic applications



Current traffic management and security does not follow application life cycle – No automation. Any security configuration change takes weeks → No support for dynamic and ephemeral applications, No support for apps that need to scale-out across edges

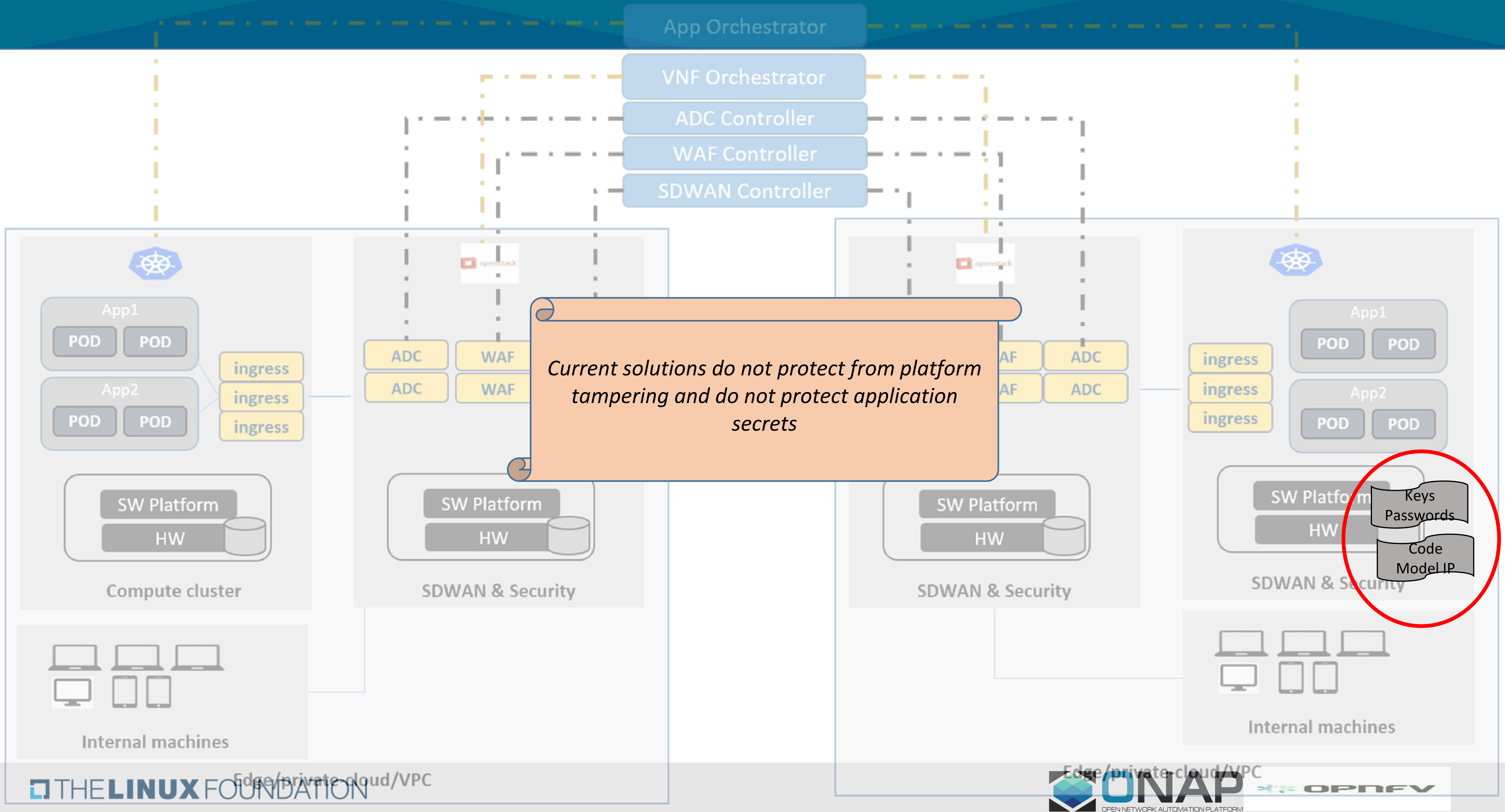
Does not support new kinds of applications

Challenge: Lack of Zero Trust Security



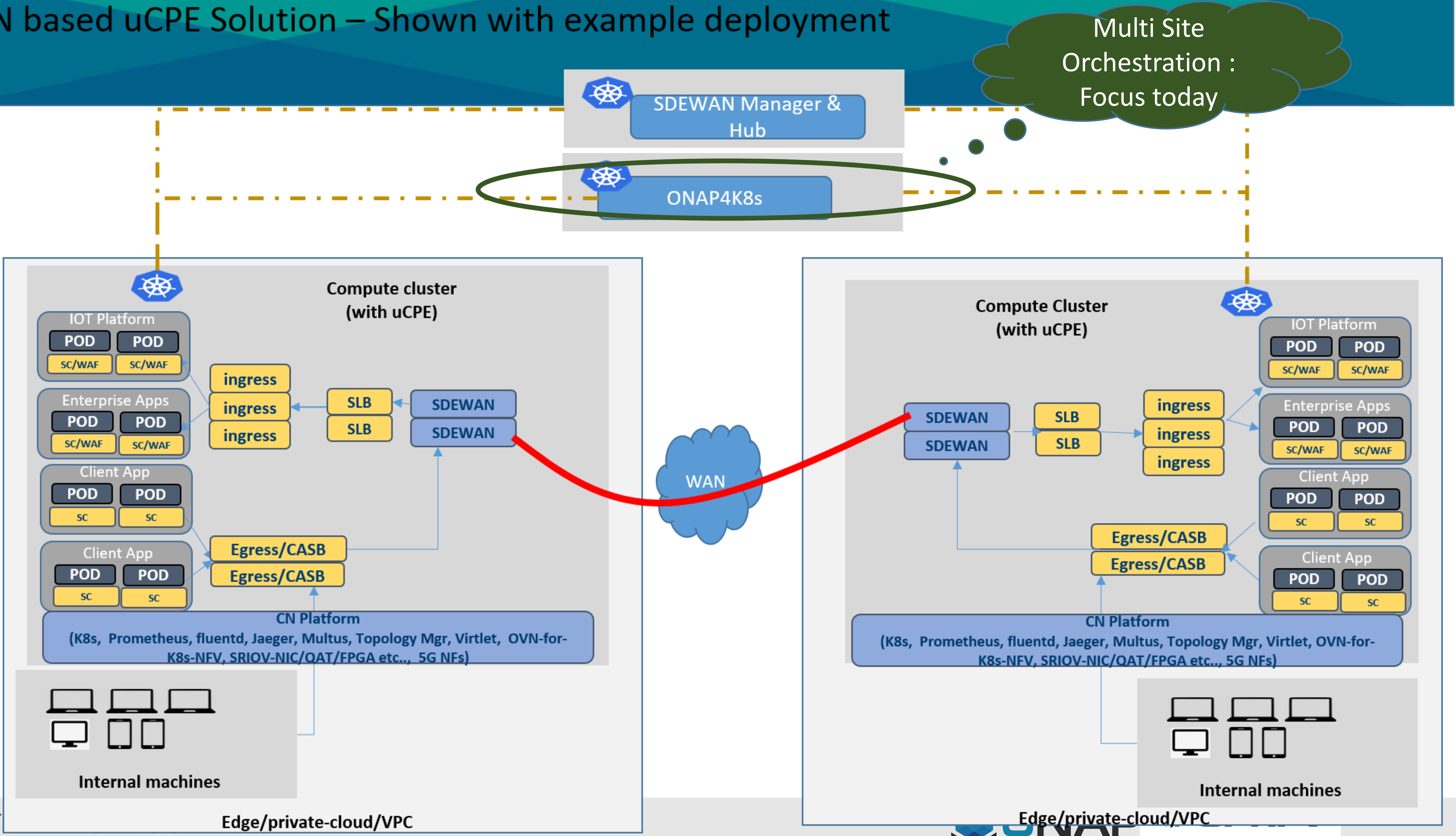
Application that spread across multiple sites are treated as different applications and protecting using traditional security create a bigger hole which allows attackers to access to break the internal services. No source level filtering
→ Need zero trust security

Challenge: Inadequate security for secrets, keys and IP

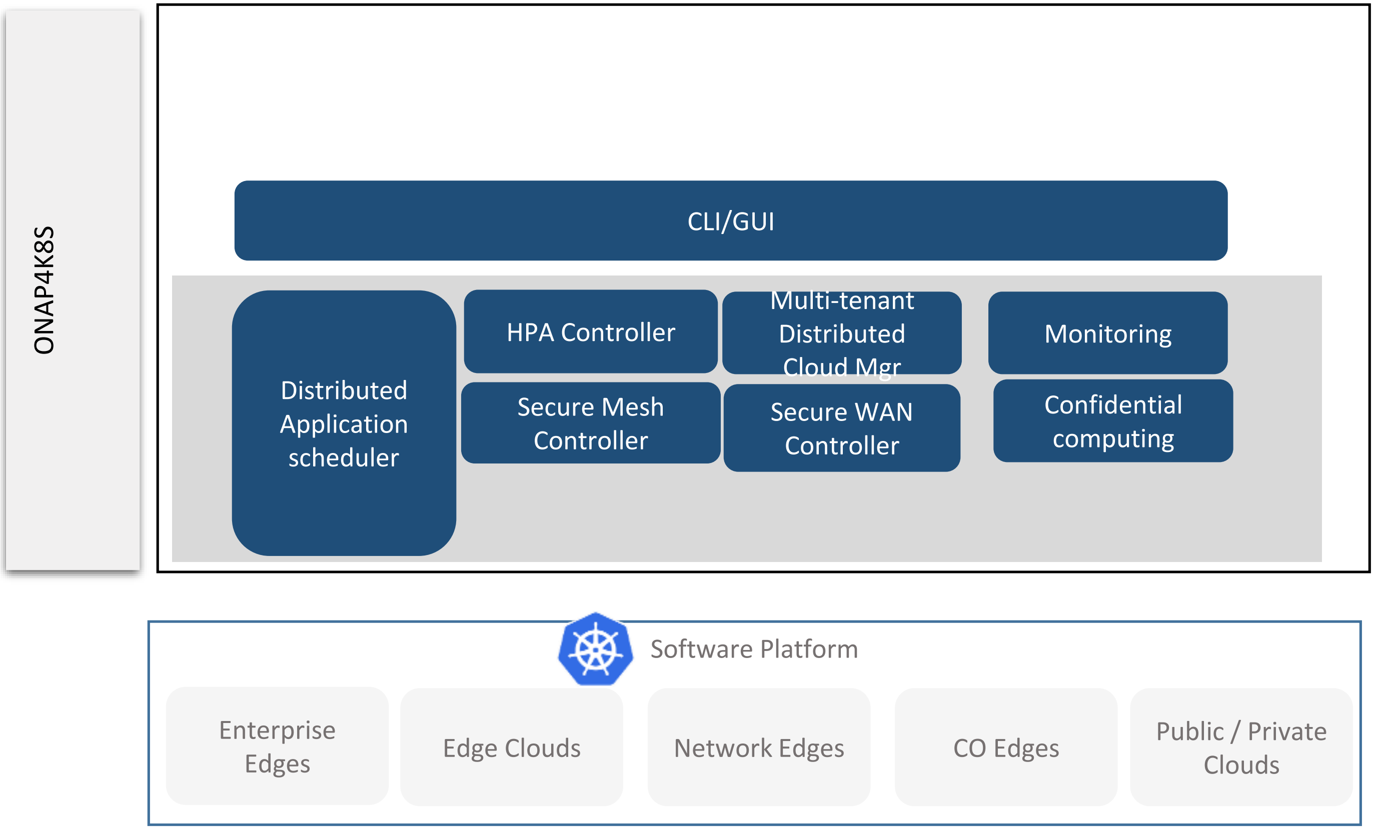


Current solutions do not protect from platform tampering and do not protect application secrets

ICN based uCPE Solution – Shown with example deployment

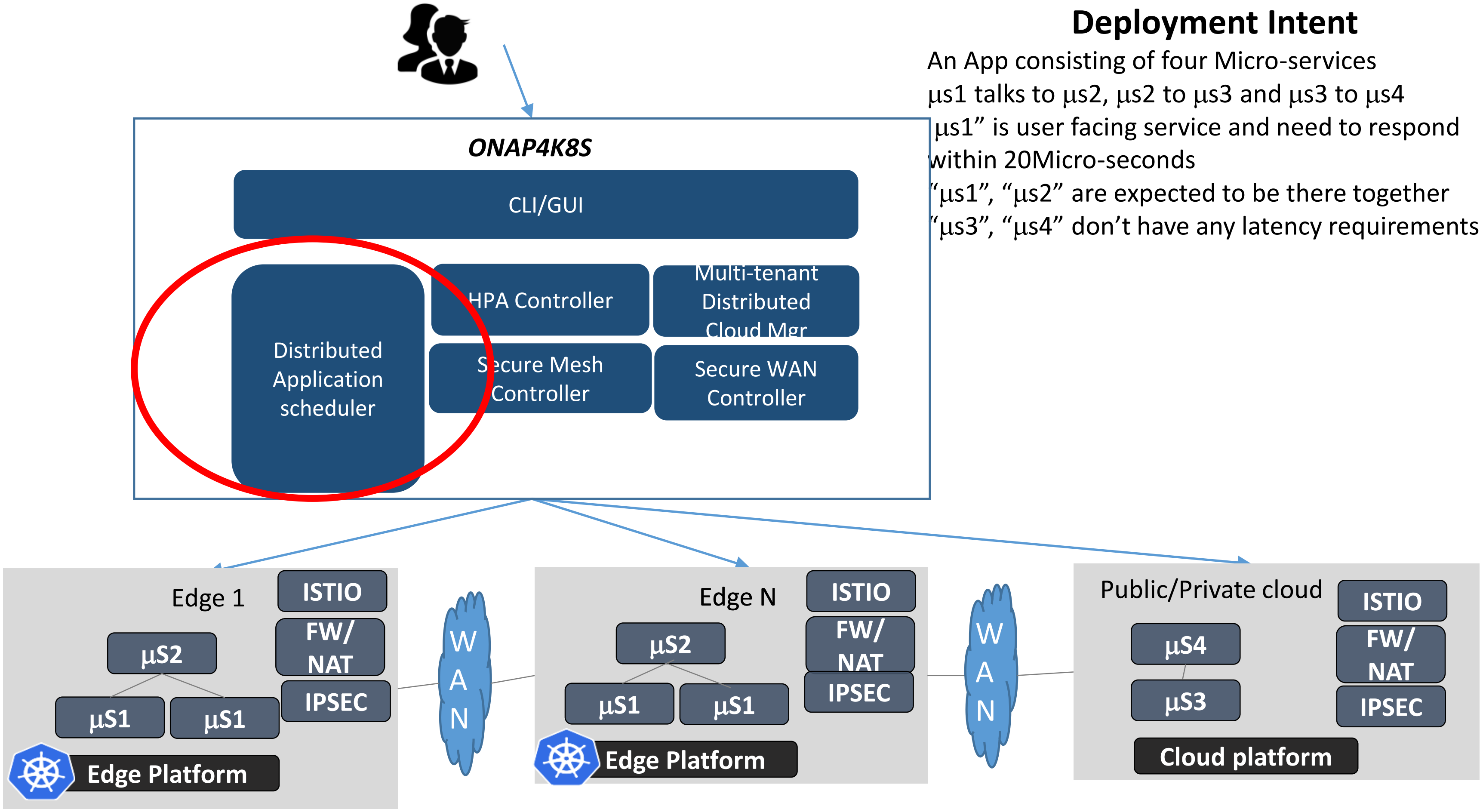


Solution ingredient: ONAP4K8S for Distributed Applications



- Generic and flexible distributed **application scheduler** with extensible placement and action controllers
- **Hardware Platform Aware** scheduling with Auto discovery of platform capabilities
- Auto configuration of **service meshes (e.g., ISTIO) and security policies** (e.g. firewall & NAT) with workload LCM.
- **Secure WAN** cloud across edge groups
- Protect Application IP via confidential computing
- Monitor distributed application performance, accesses

Distributed Application Scheduler



Deployment Intent

An App consisting of four Micro-services

μs1 talks to μs2, μs2 to μs3 and μs3 to μs4

μs1” is user facing service and need to respond within 20Micro-seconds

“μs1”, “μs2” are expected to be there together

“μs3”, “μs4” don’t have any latency requirements

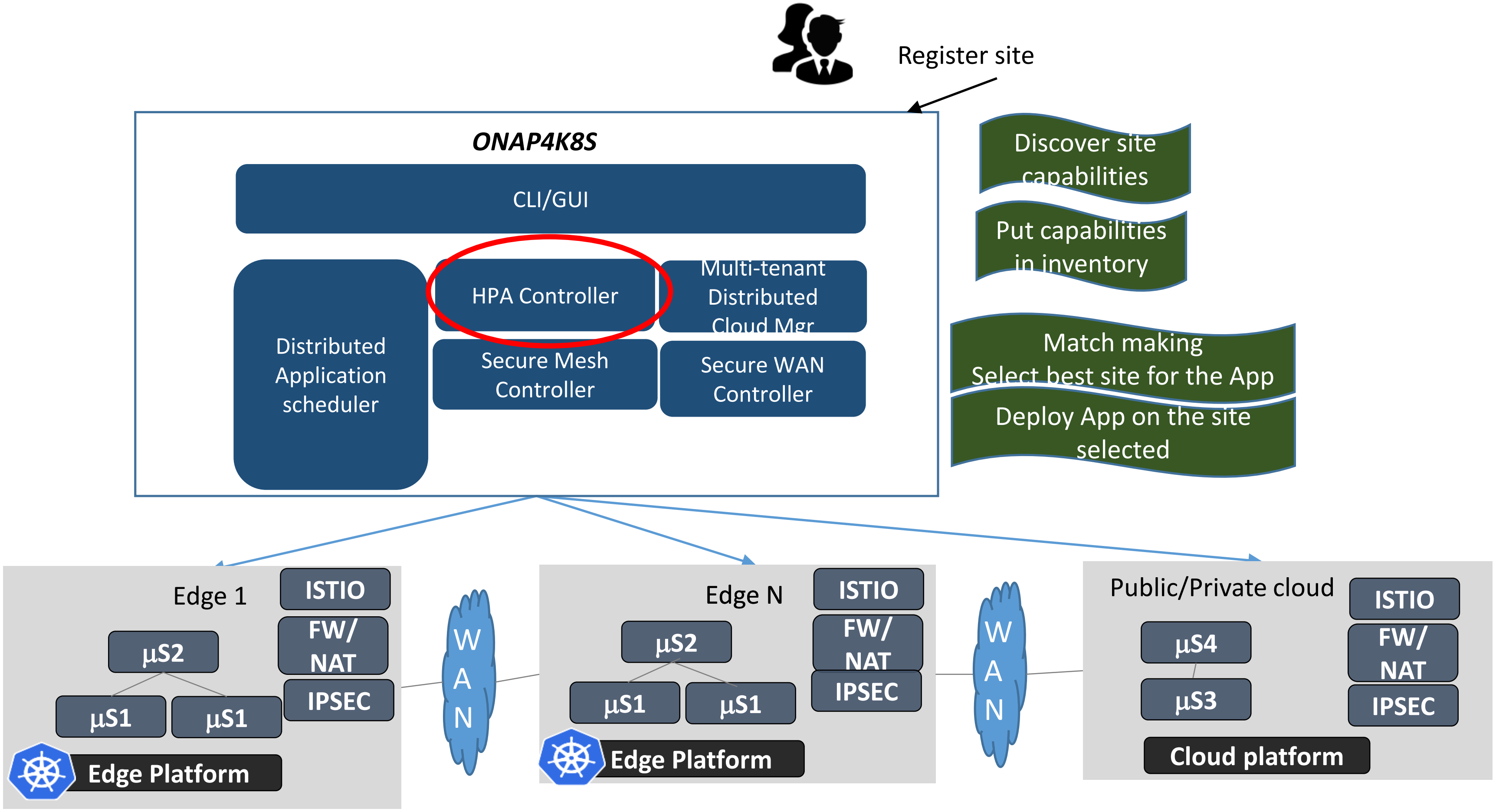
Why?

- Geo replication
- Geo Distribution

New Edges locations -> No manual intervention

Not only for scheduling for apps, but also VNFs/CNFs.

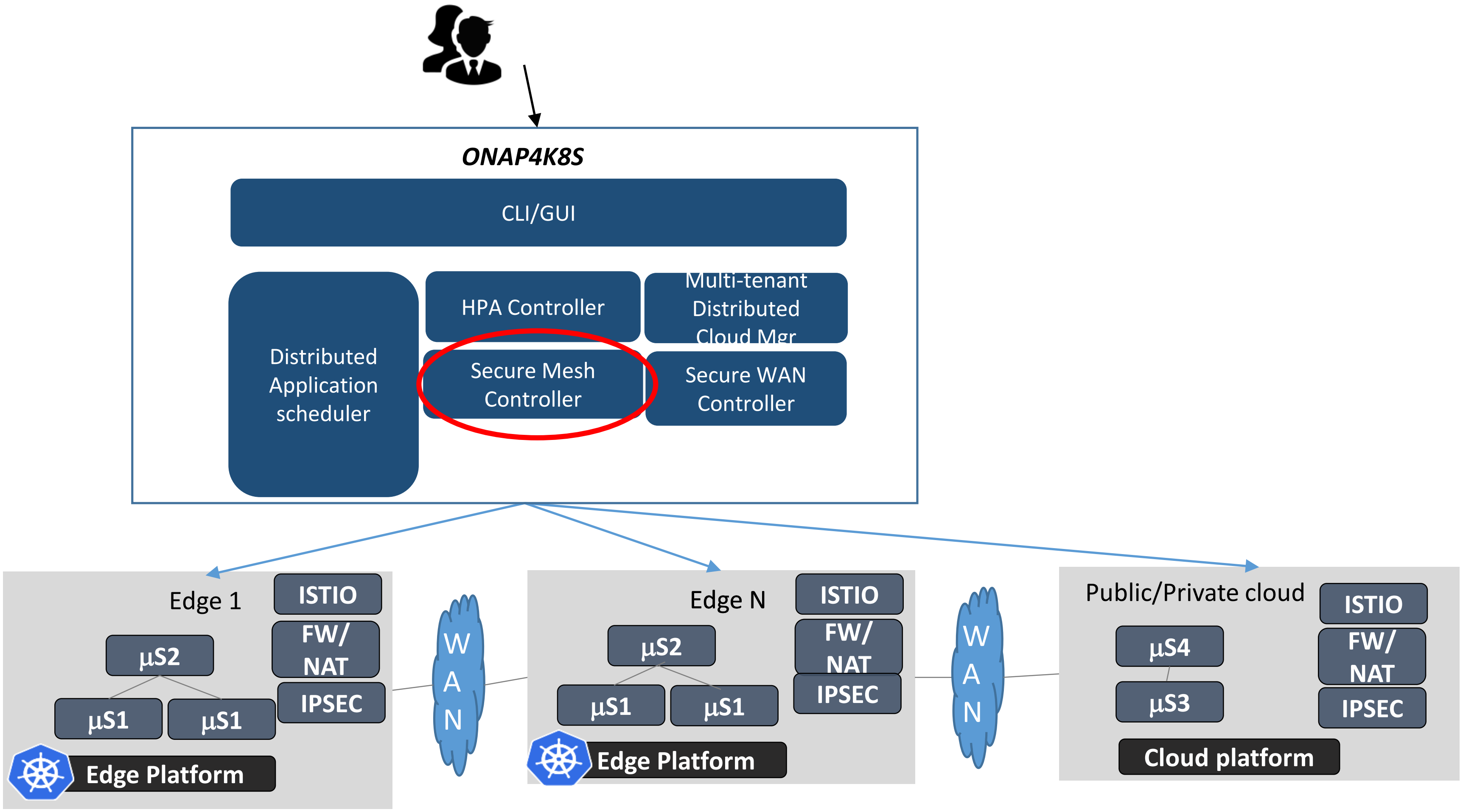
HPA Controller



Why?

Selecting right edge and flavor based on Edge/Cloud capabilities and Micro-service requirements

Secure Mesh Controller



Why?

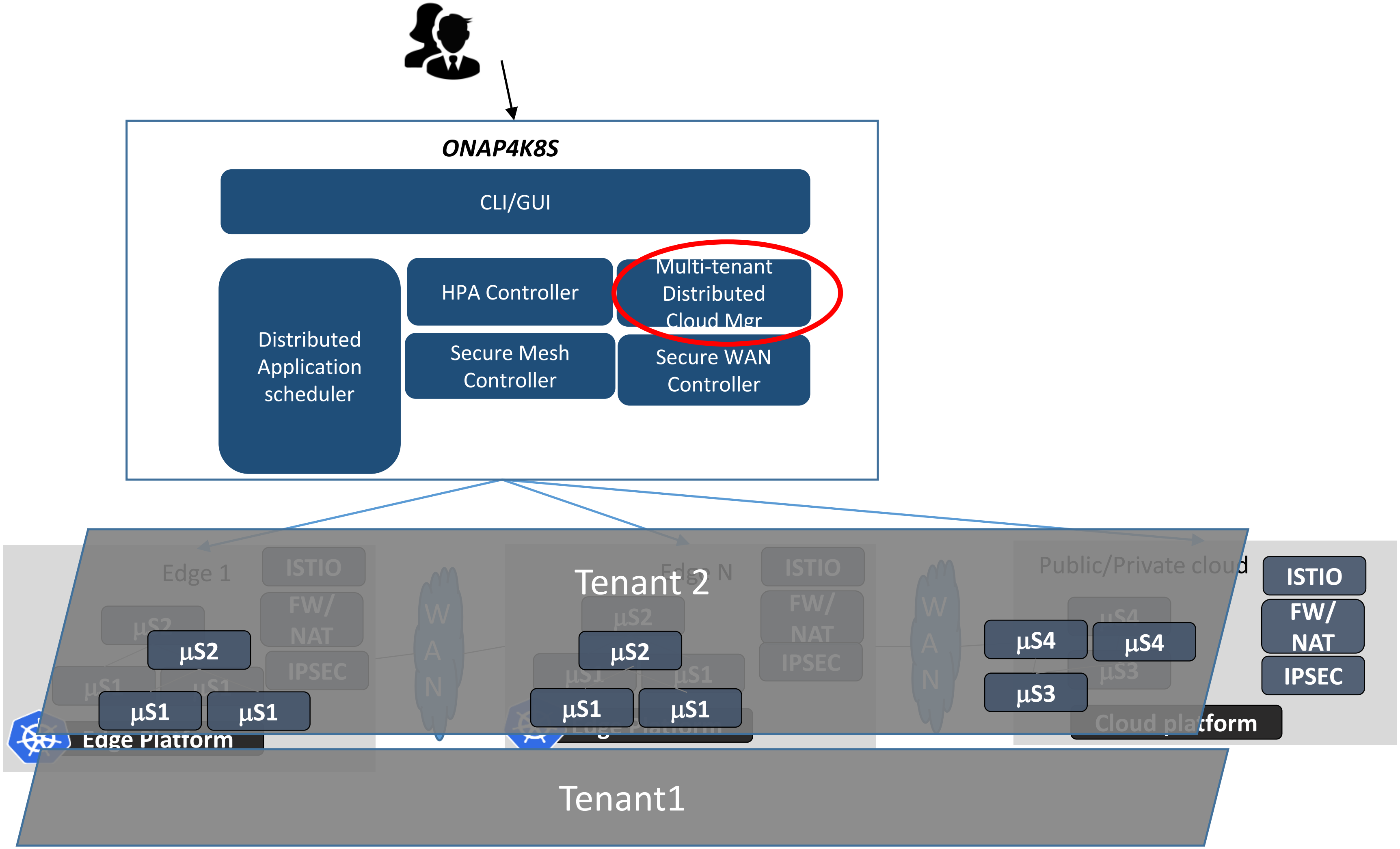
To enable secure communication among microservices in different locations

To enable connectivity with users

How:

- **Programming ISTIO egress/ingress**
- **Auto NAT and FW configuration**
- **Programming DNS entries (e.g Route 53)**

Multi-Tenant Distributed Cloud Manager



Why?

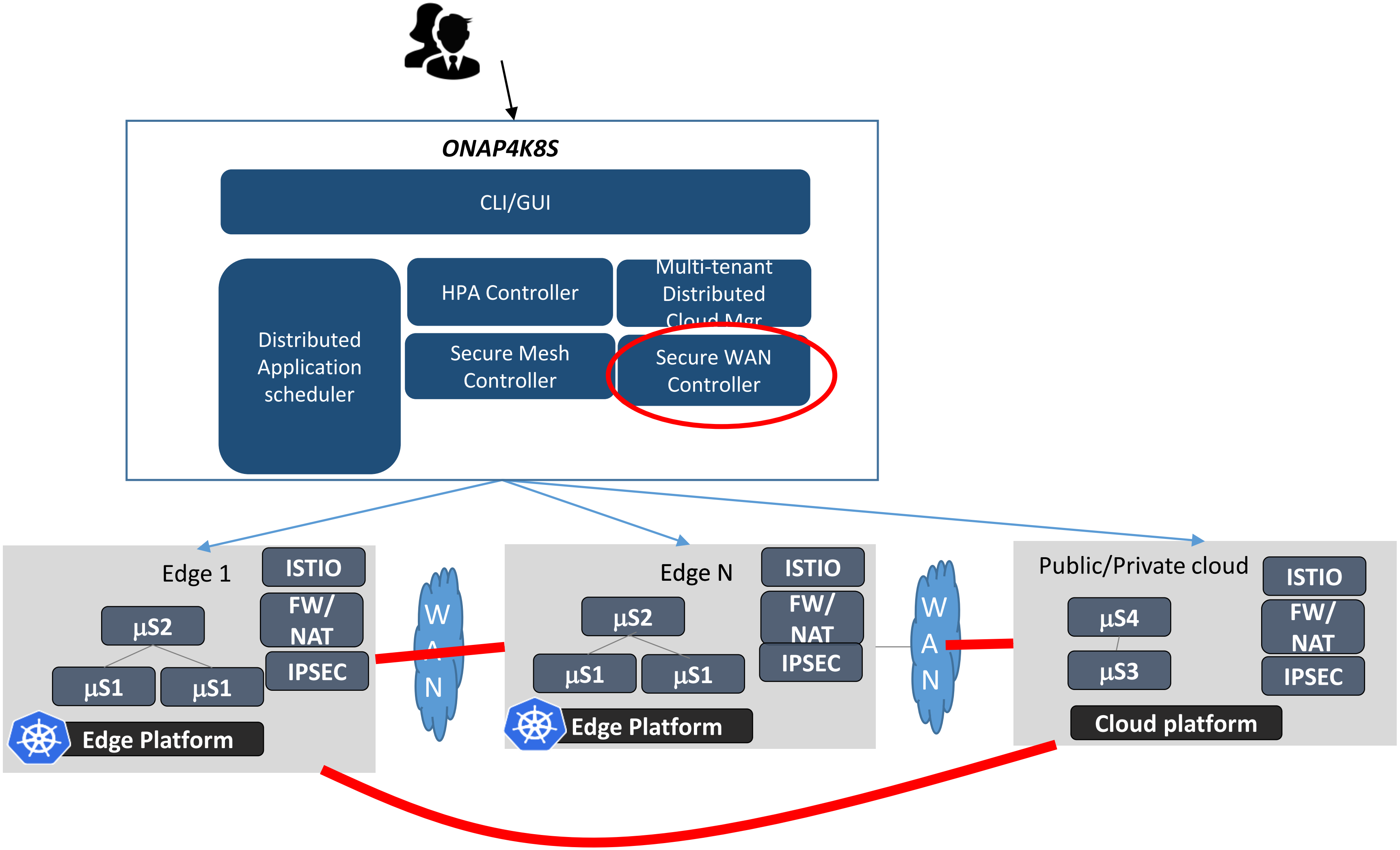
Easy creation of tenants across multiple edges using one user operation

How:

- **Creating namespaces**
- **Users**
- **Roles**
- **Permissions**
- **ISTIO control plane**
- **Quotas**

across multiple sites

Secure WAN Controller



Why?

To secure connect edges
No static public IP address

How:

- **Auto configuration of IPSEC functionality of Edge platform.**
- **Support for tunnel mesh and Hub-and-spoke**

ONAP4K8S – Summary

ONAP4K8S is not just for Telcos, but also for Enterprises

Intel is leading the effort in the community

Feedback from many Enterprises, Telcos, MSPs

Started to see contributions from Orange, Tech Mahindra, Aarna, Samsung.

Status:

- One release is made
- Distributed Application Orchestration is in planning
- Plan to complete majority of development in 2020

ONAP

OPEN NETWORK AUTOMATION PLATFORM

Q&A

Edge Platform Requirements

View in slide show mode

Co-existence of multiple deployment types
(VNFs, CNFs, VMs, Containers and functions)

Advanced Networking support
(Multiple networks, Provider networks, Dynamic Route/network creation, Service function chaining)

Soft and Strict Multi-tenancy

AI based Predictive placement
(Collection using Prometheus, Training and inferencing framework)

Slicing in each tenant
(QoS On per Slice basis, VLAN networks for slices, VNFs/CNFs/VMs/PODs on per slice basis or slice configuration facility on shared VNFs/CNFs)

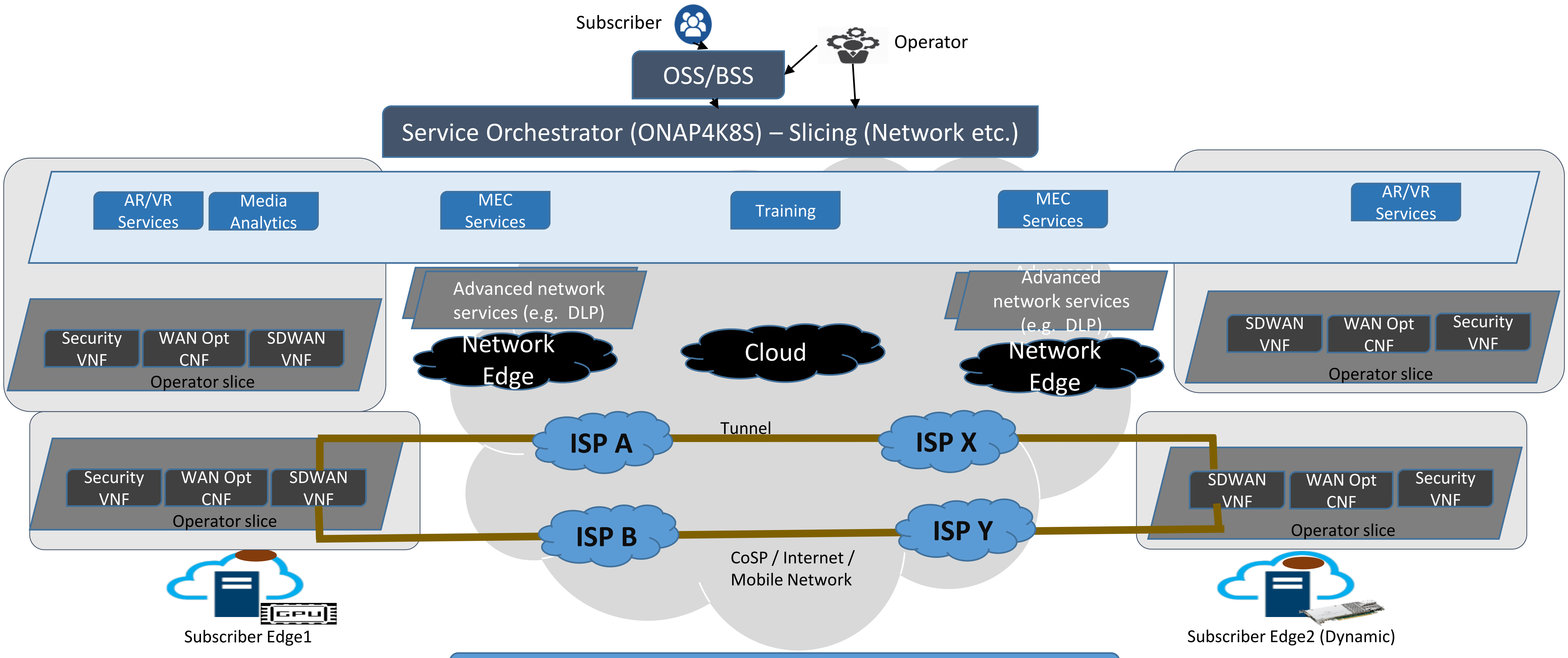
Service Mesh for Micro-services
(Acceleration using Cilium' Kernel bypass among service mesh side cars - e.g. Envoys; and others)

Programmable CNI
(to allow SFC and avoid multiple protocol layers)

Security Orchestration
(Key orchestration for securing private keys of CA and user certificates)

Managed SDWAN and Compute use case

View in slide show mode



Subscriber

Operator

OSS/BSS

Service Orchestrator (ONAP4K8S) – Slicing (Network etc.)

AR/VR Services

Media Analytics

MEC Services

Training

MEC Services

AR/VR Services

Advanced network services (e.g. DLP)

Advanced network services (e.g. DLP)

Network Edge

Cloud

Network Edge

Security VNF

WAN Opt CNF

SDWAN VNF

Operator slice

SDWAN VNF

WAN Opt CNF

Security VNF

Operator slice

Security VNF

WAN Opt CNF

SDWAN VNF

Operator slice

SDWAN VNF

WAN Opt CNF

Security VNF

Operator slice

ISP A

Tunnel

ISP X

ISP B

ISP Y

CoSP / Internet / Mobile Network

Subscriber Edge1

Subscriber Edge2 (Dynamic)

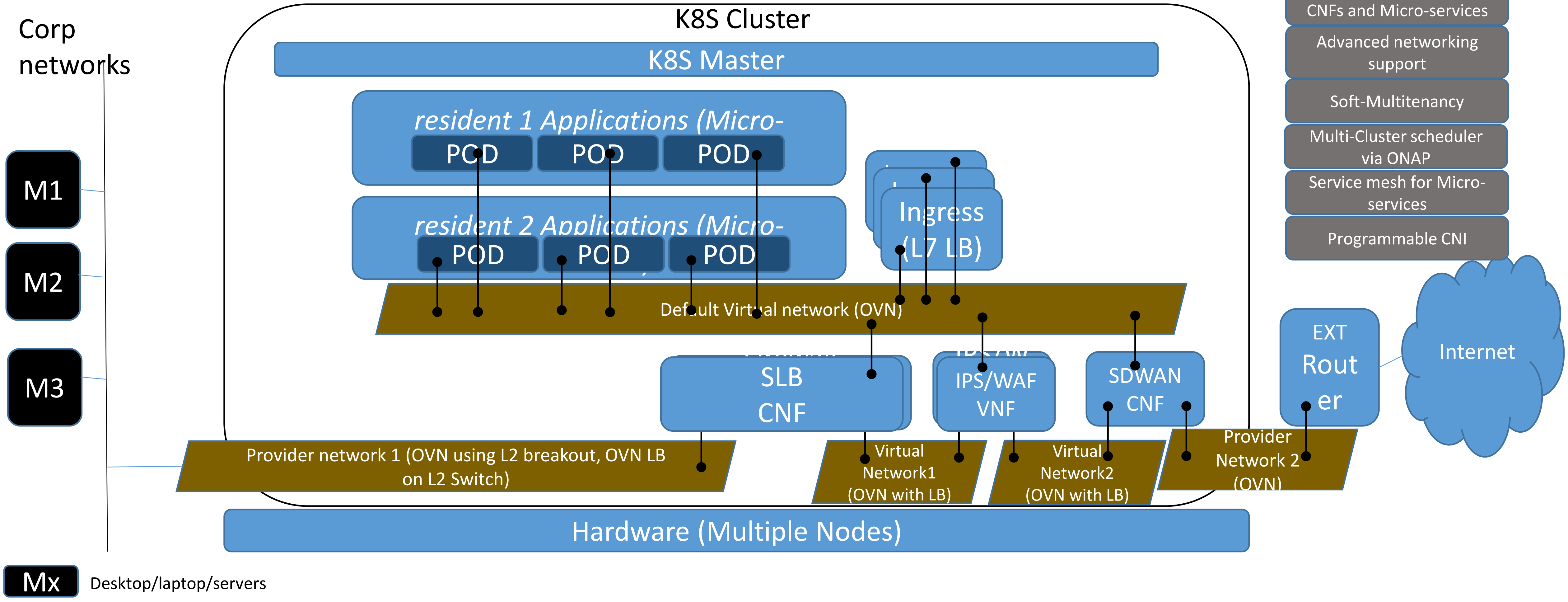
May be tested with dummy apps and OpenWRT, if no real apps from community

● Kubernetes

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

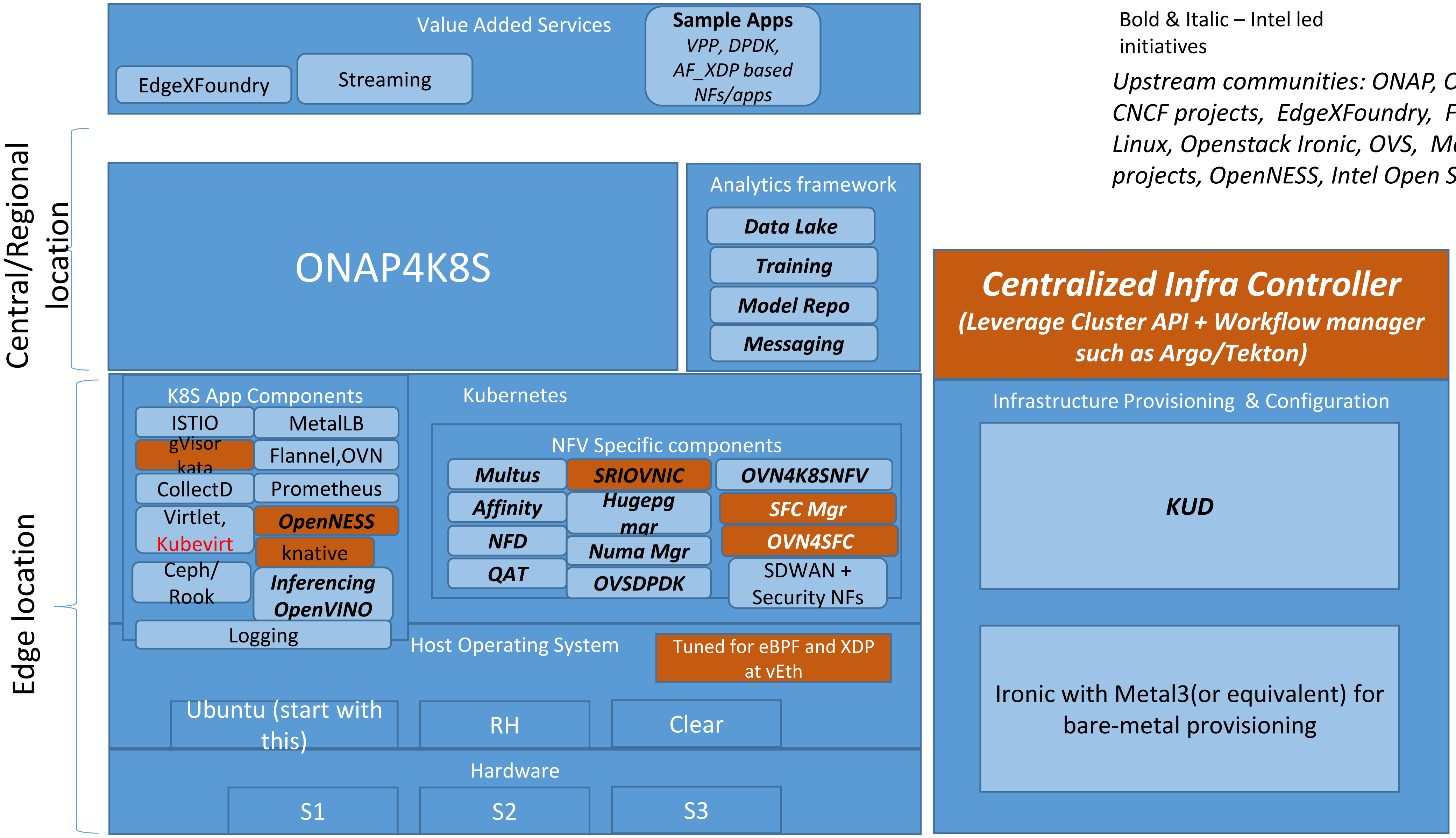
What it proves

- Coexistence of VNFs, CNFs and Micro-services
- Advanced networking support
- Soft-Multitenancy
- Multi-Cluster scheduler via ONAP
- Service mesh for Micro-services
- Programmable CNI



Corp networks
M1
M2
M3
K8S Cluster
K8S Master
resident 1 Applications (Micro-
POD
POD
POD
resident 2 Applications (Micro-
POD
POD
POD
Ingress (L7 LB)
Default Virtual network (OVN)
SLB CNF
IPS/WAF VNF
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

Cloud Native App & NFV Stack – BICN (Potential to use CNF test bed)



Value Added Services

EdgeXFoundry

Streaming

Sample Apps
VPP, DPDK, AF_XDP based NFs/apps

Central/Regional location

ONAP4K8S

Analytics framework

Data Lake

Training

Model Repo

Messaging

Centralized Infra Controller

(Leverage Cluster API + Workflow manager such as Argo/Tekton)

Infrastructure Provisioning & Configuration

KUD

Ironic with Metal3(or equivalent) for bare-metal provisioning

Edge location

Kubernetes

K8S App Components

ISTIO

MetalLB

gVisor kata

Flannel,OVN

CollectD

Prometheus

Virtlet, Kubevirt

OpenNESS

knative

Ceph/ Rook

Inferencing OpenVINO

Logging

NFV Specific components

Multus

SRIOVNIC

OVN4K8SNFV

Affinity

Hugepg mgr

SFC Mgr

NFD

Numa Mgr

OVN4SFC

QAT

OVSDPDK

SDWAN + Security NFs

Host Operating System

Tuned for eBPF and XDP at vEth

Ubuntu (start with this)

RH

Clear

Hardware

S1

S2

S3

Bold & Italic – Intel led initiatives

Upstream communities: ONAP, OPNFV, Many CNCF projects, EdgeXFoundry, FD.IO, DPDK, Linux, Openstack Ironic, OVS, Many ASF projects, OpenNESS, Intel Open Source