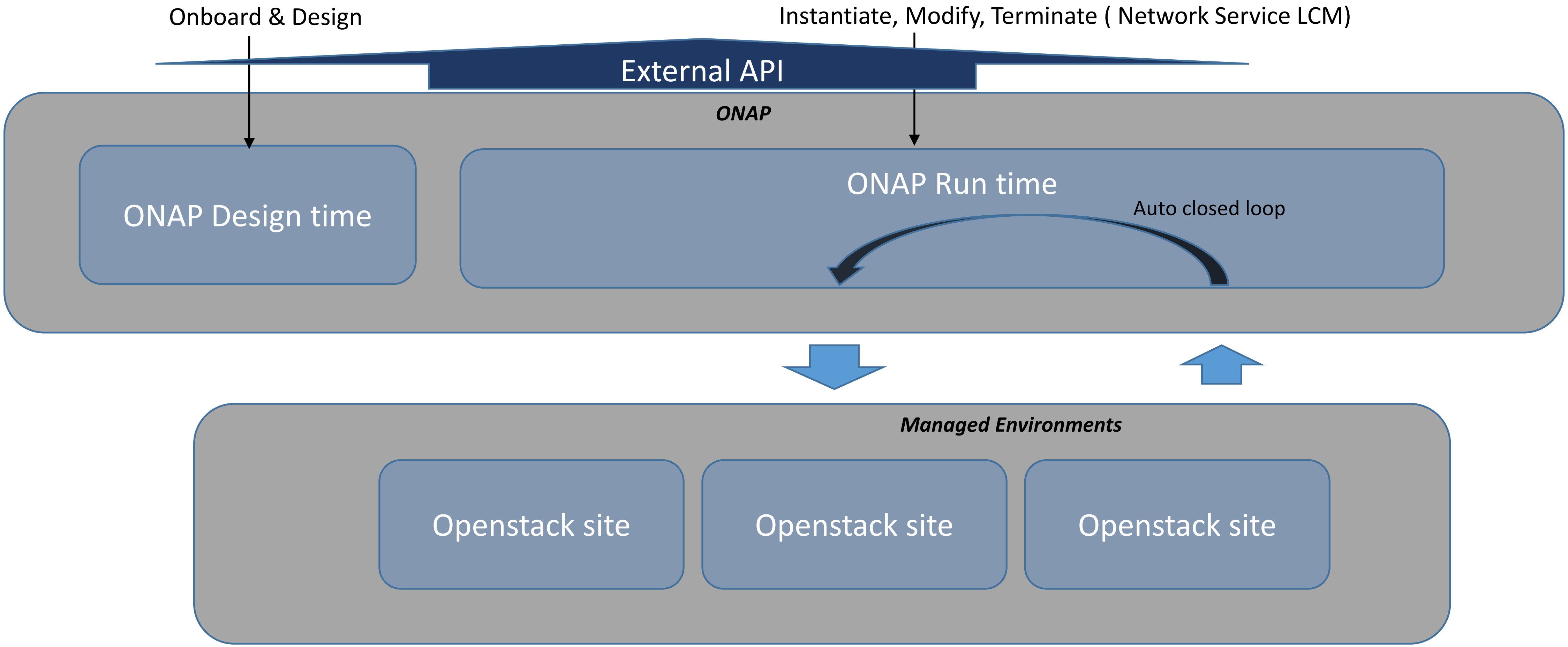
Multi Site Orchestration with ONAP4K8s (ONAP-for-K8s)

A foundation for Multi Edge & Cloud Orchestration

Srinivasa Addepalli

Contact: Srinivasa.r.addepalli@intel.com

ONAP Overview (till R4)



Onboard & Design

Instantiate, Modify, Terminate (Network Service LCM)

External API

ONAP

ONAP Design time

ONAP Run time

Auto closed loop

Managed Environments

Openstack site

Openstack site

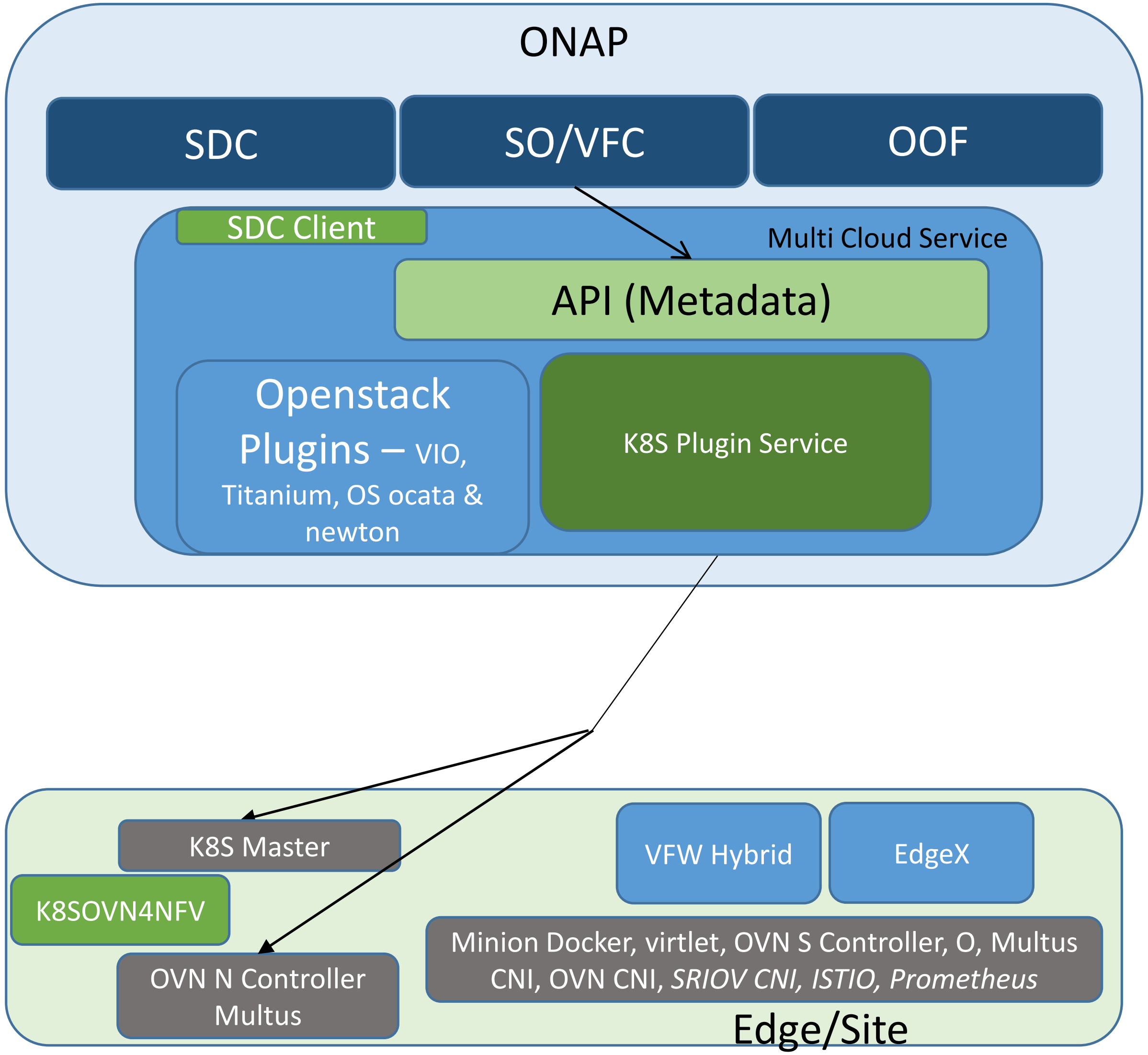
Openstack site

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** **ISTIO security for ONAP4K8S**

ONAP – K8S Support (till R5)


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. K8S-OVN4NFV (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 containers and with rest of ONAP

R6 and beyond: The “Edge Application” Challenge

- Large Number of Edge Clouds
- Network edge clouds could be in the order of 10,000
- On-Premise edge clouds could be in the order of 100,000
- Large Number of App Instances
- CNF, IoT, Analytics, and Cloud Native apps in the order of 1,000
- With network slicing the instances in the order of 10,000
- Dynamic Changes
- Network slicing, configuration changes, cloud migration, CI/CD, etc.
- Termination will also be critical as edge resources are precious
- Complex Geo Distributed Apps
- Geo distributed apps to span multiple clouds: edge, core, public etc.
- Complex networking & security configs for geo-distributed apps

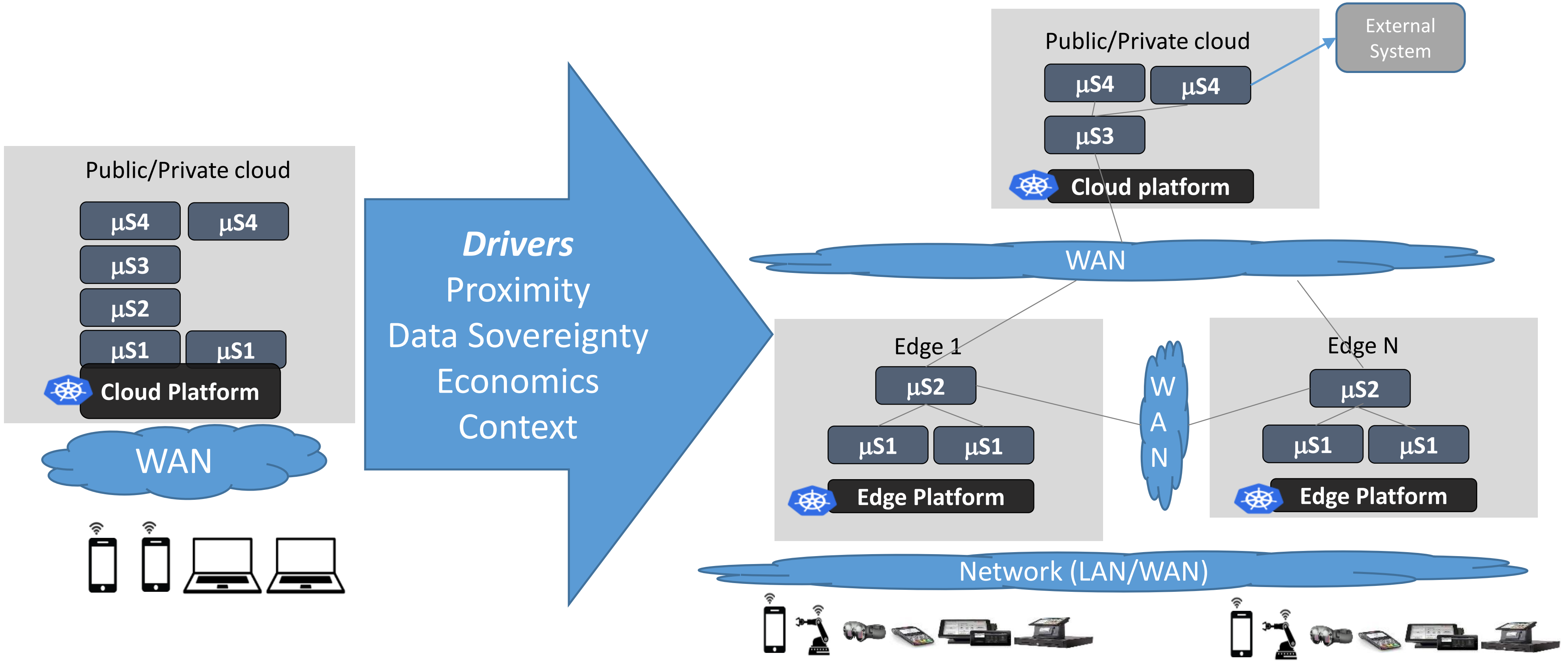
Operational Implications

- Overwhelming diversity of element managers and GUI screens, as well as vendor installers
- Confounding ad-hoc automation, and impractical manual break/ fix methodologies

Operational Requirements

- App-agnostic, full automation of orchestration and lifecycle management
- Monitoring, analytics, and real-time “policy driven” closed loop automation

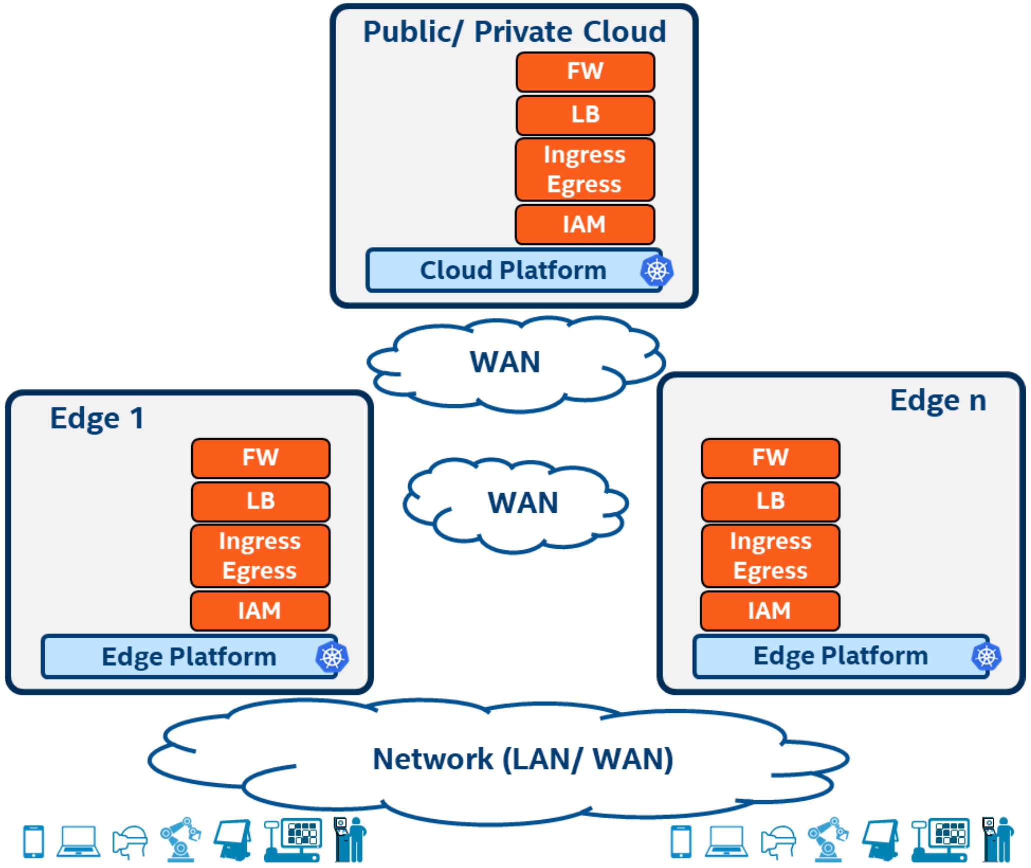
Towards ONAP4K8S R6 and beyond
Application Transformation – Centralized to Geo-Distributed



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

Need for Multi Edge/Cloud Orchestrator

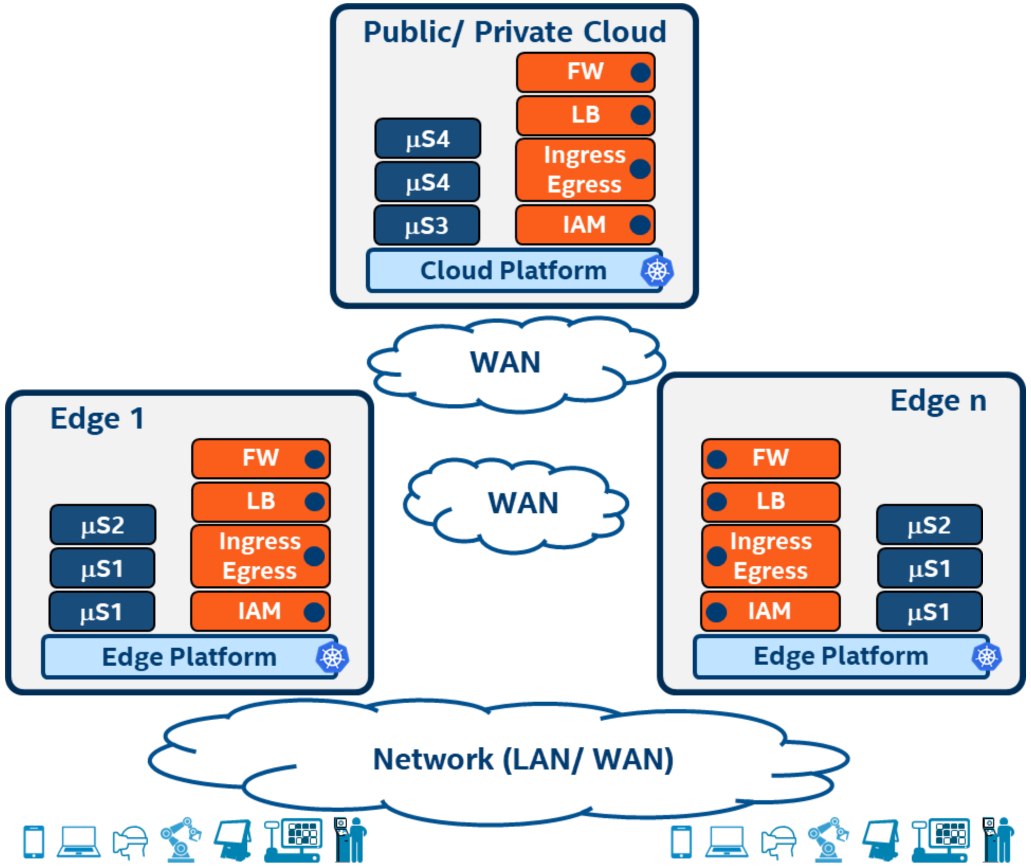
Manage On-Demand & Dynamic Distributed Apps



- **Problem**: Thirty-plus configuration steps are required to program security & connectivity functions at each location
- **Assumption**: Curated security & connectivity functions are resident and available
- **Assumption**: Security & connectivity functions are resource constrained

Activate the application micro-services only when users are present

Manage On-Demand & Dynamic Distributed Apps



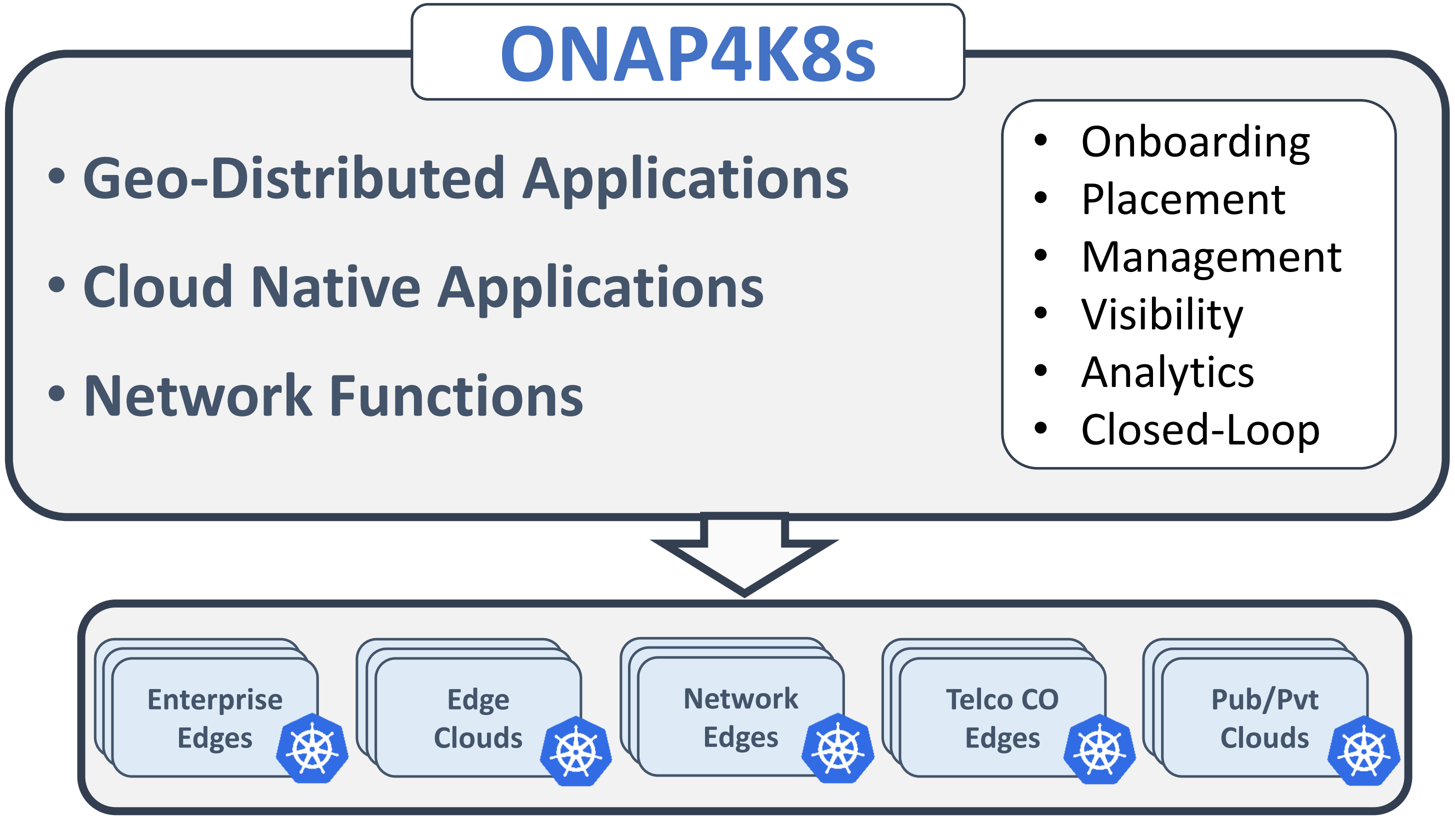
- **Problem**: Thirty-plus configuration steps are required to program security & connectivity functions at each location
- **Assumption**: Curated security & connectivity functions are resident and available
- **Assumption**: Security & connectivity functions are resource constrained

Activate the application micro-services only when users are present

Automate configuration of security & connectivity with application deployment

Edge Multi-Cluster Orchestrator (ONAP4K8s)

- **Application/Service Orchestration**
  - For applications and services delivered across multiple administrative infrastructures
- Resource Orchestration
  - For management of virtual compute, storage & networking resources shared by apps/ services in an administrative zone
- Infrastructure Orchestration
  - For the building and management of Kubernetes clusters, including deployment of infrastructure services such as software updates, networking, and storage



ONAP4K8s

- **Geo-Distributed Applications**
- **Cloud Native Applications**
- **Network Functions**

- Onboarding
- Placement
- Management
- Visibility
- Analytics
- Closed-Loop

Enterprise Edges

Edge Clouds

Network Edges

Telco CO Edges

Pub/Pvt Clouds

ONAP4K8s is developed in LFN/ONAP repo as a standalone project

What is ONAP4K8S?

Is Multi Site Distributed App Orchestrator

Independently can be deployed or deployed with rest of ONAP

Supporting deployment of both applications and network functions

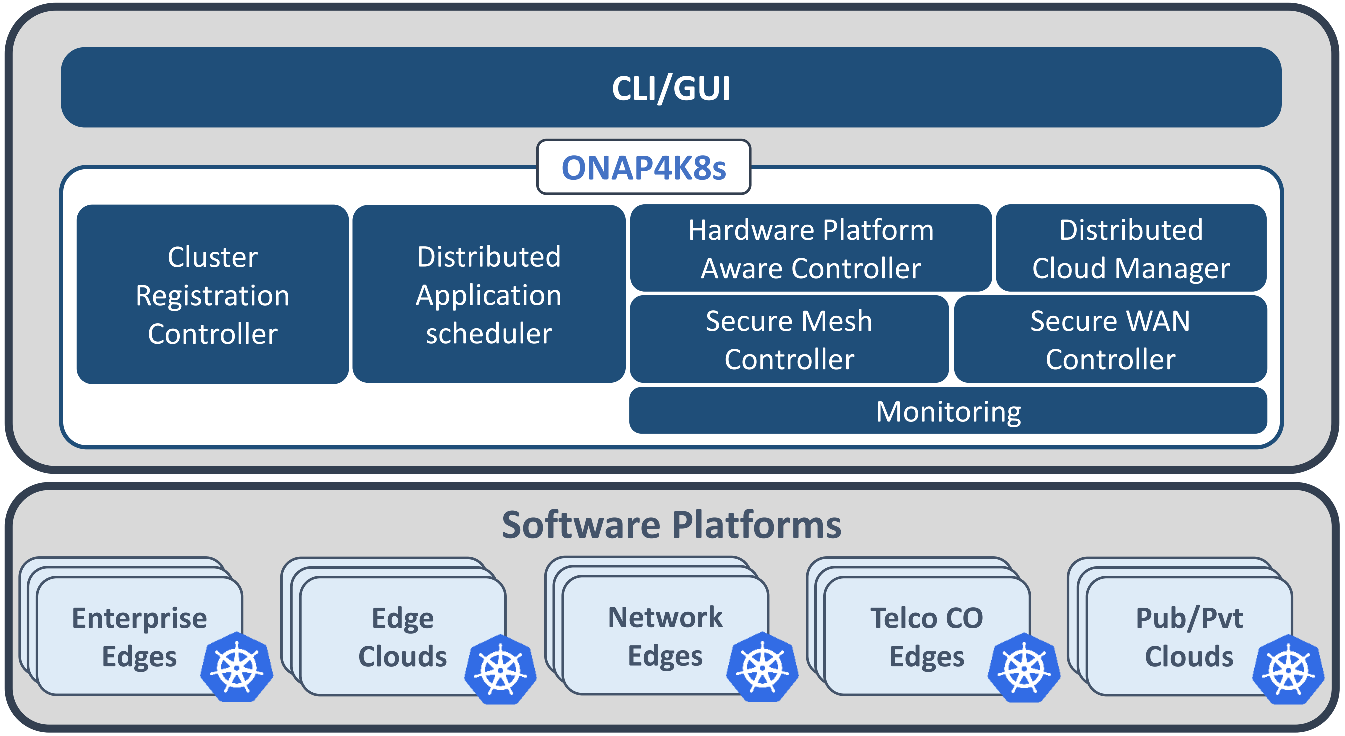
Supporting workload types - VMs, Containers, VNFs and CNFs

Lightweight & high performance

Micro-service based architecture

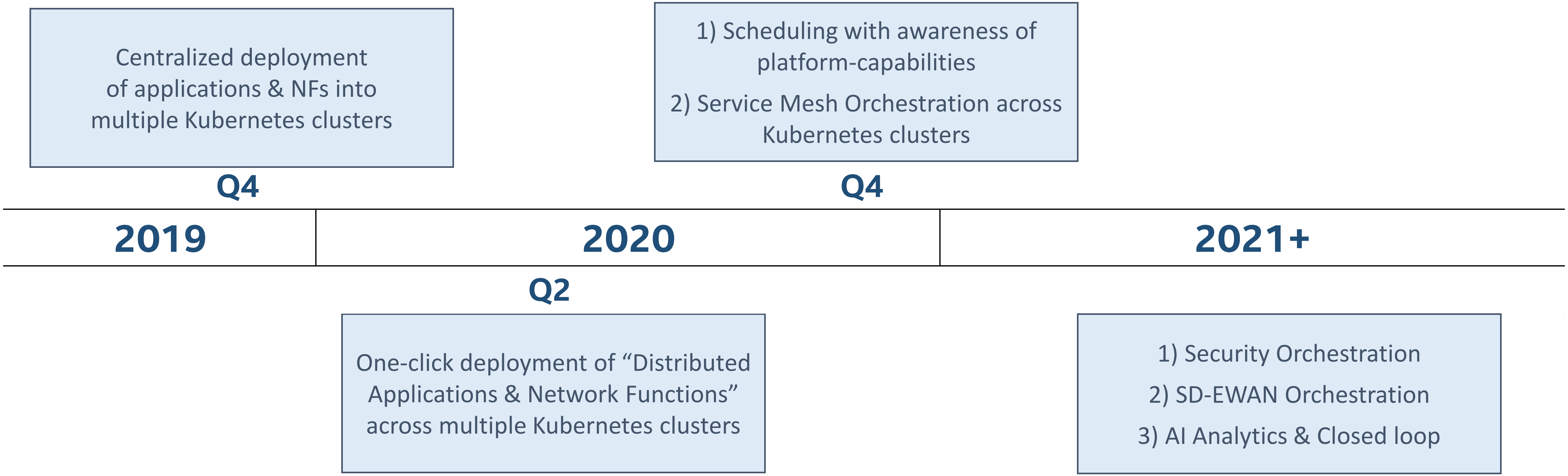
(Leverages ISTIO, AuthService & KeyCloak for IAM, Mutual TLS
FluentD for logging
Prometheus for metrics
Jaeger for distributed tracing.
No centralized configuration DB, One Document DB and One KV DB

High Level ONAP4K8s Architecture



- **Cluster Registration Controller** registers clusters by cluster owners
- **Distributed Application Scheduler** provides simplified, and extensible placement
- **Hardware Platform Aware Controller** enables scheduling with auto-discovery of platform features/ capabilities
- **Distributed Cloud Manager** presents a single logical cloud from multiple edges
- **Secure Mesh Controller** auto-configures both service mesh (ISTIO) and security policy (NAT, firewall)
- **Secure WAN Controller** automates secure overlays across edge groups
- **Monitoring** covers distributed application performance, and accesses

ONAP4K8s Roadmap



Q4 **2019**

Centralized deployment of applications & NFs into multiple Kubernetes clusters

Q2 **2020**

One-click deployment of “Distributed Applications & Network Functions” across multiple Kubernetes clusters

Q4 **2020**

1) Scheduling with awareness of platform-capabilities

2) Service Mesh Orchestration across Kubernetes clusters

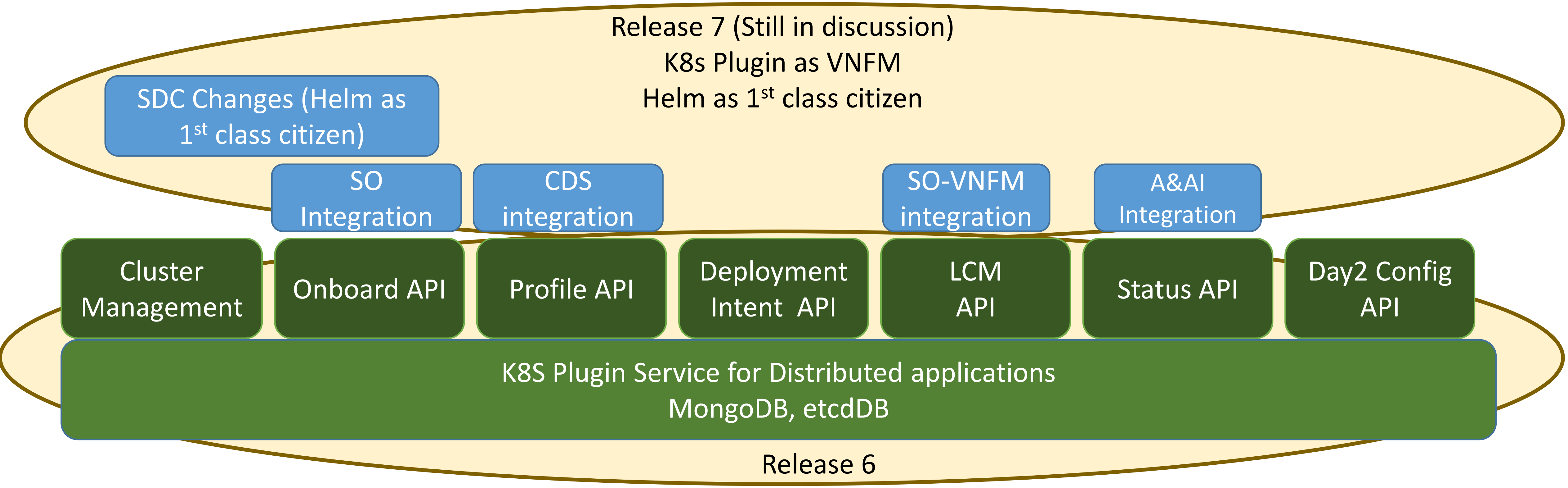
2021+

1) Security Orchestration

2) SD-EWAN Orchestration

3) AI Analytics & Closed loop

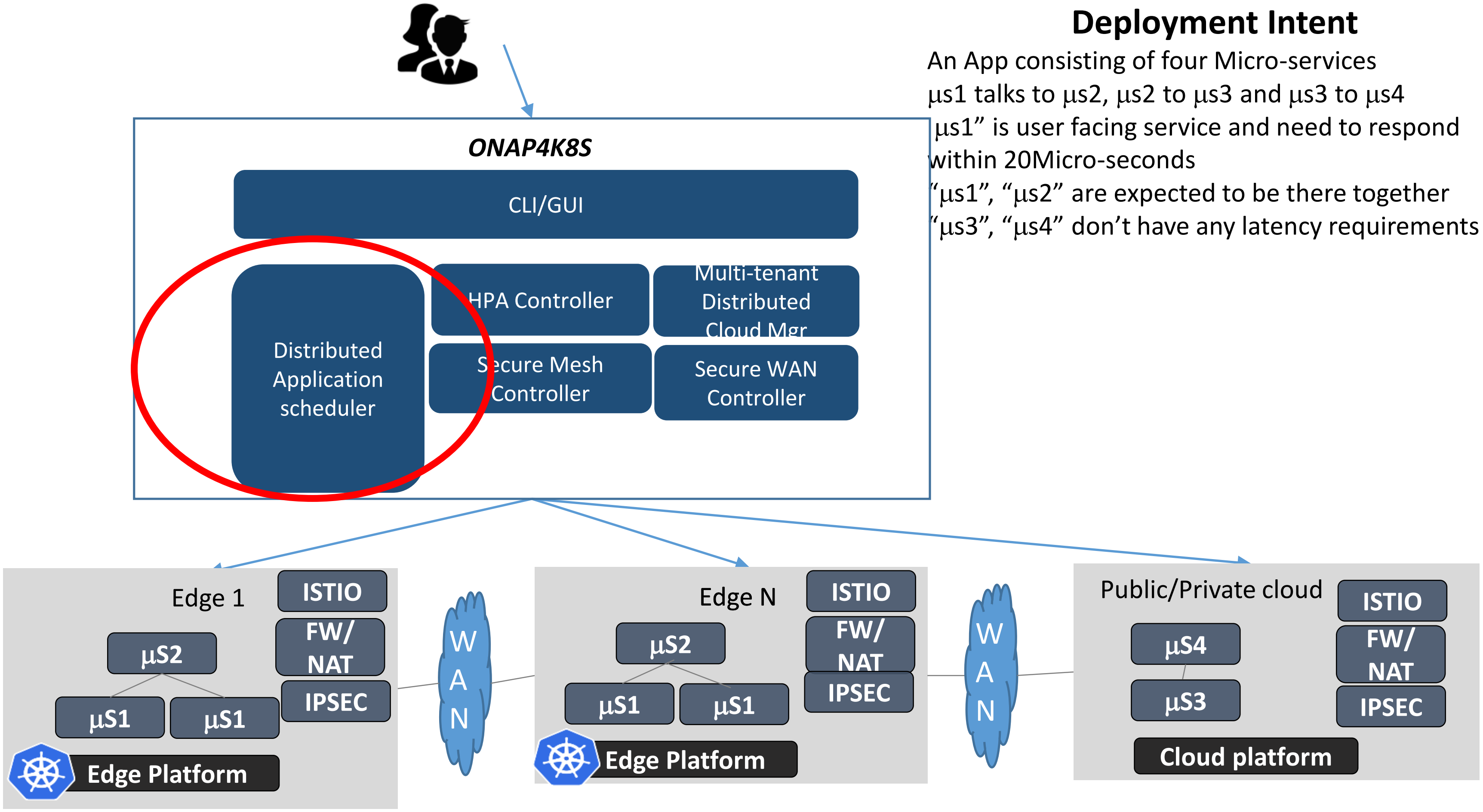
R6 and R7 Plan (With rest of ONAP)



ONAP

OPEN NETWORK AUTOMATION PLATFORM

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

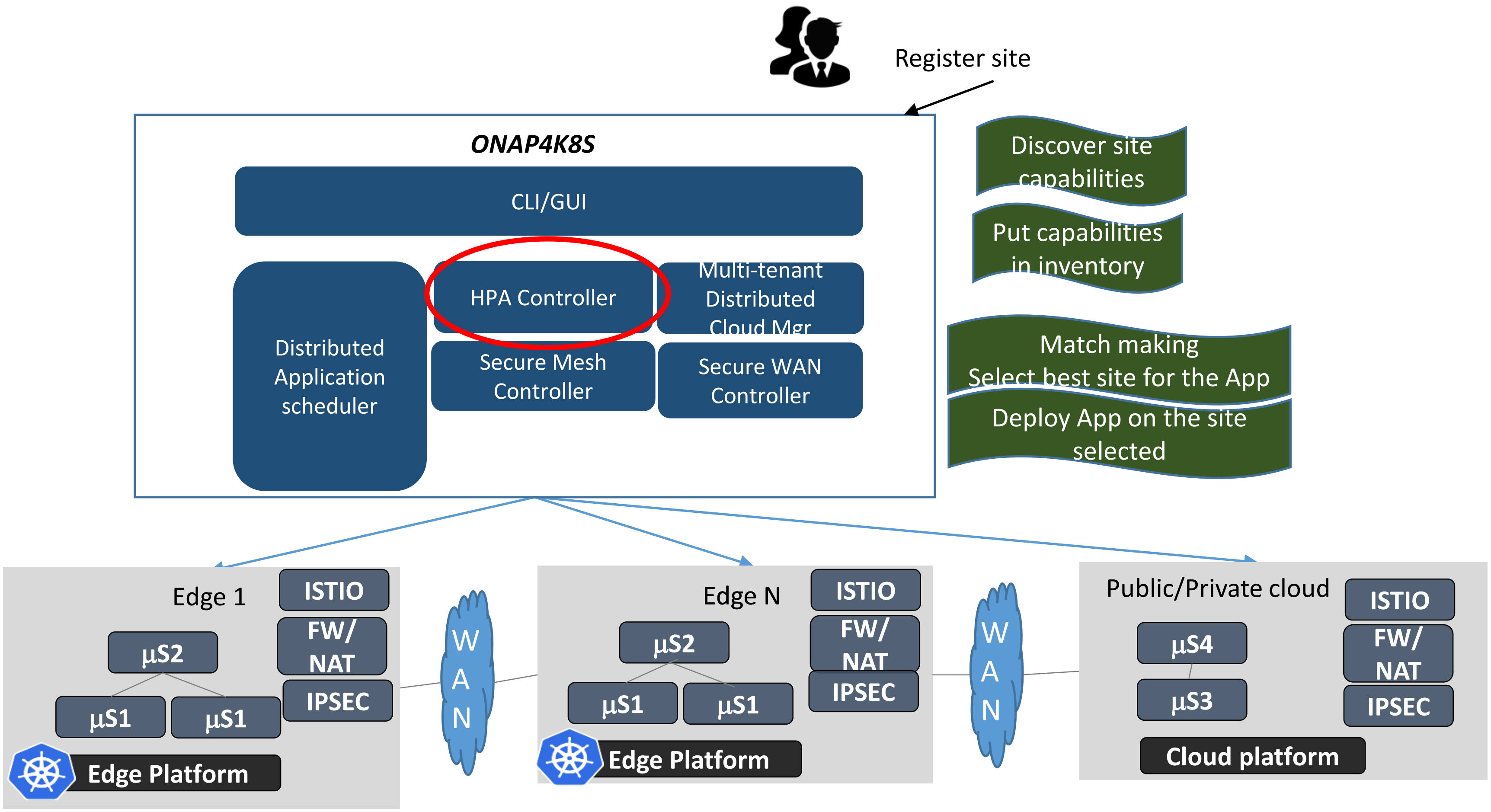
Why?

- Geo replication
- Geo Distribution

New Edges locations -> No manual intervention

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

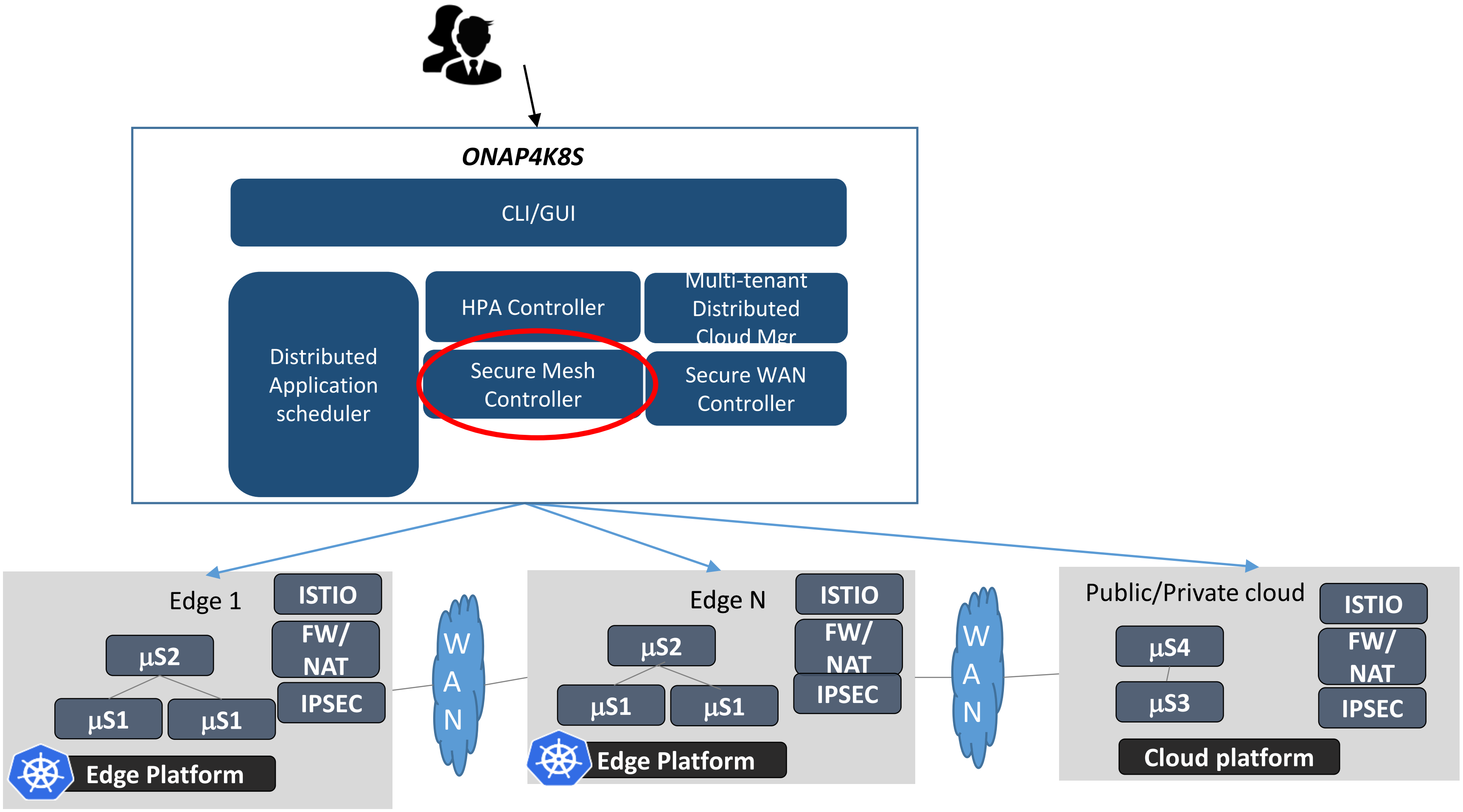
HPA Controller



Why?

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

Secure Mesh Controller



Why?

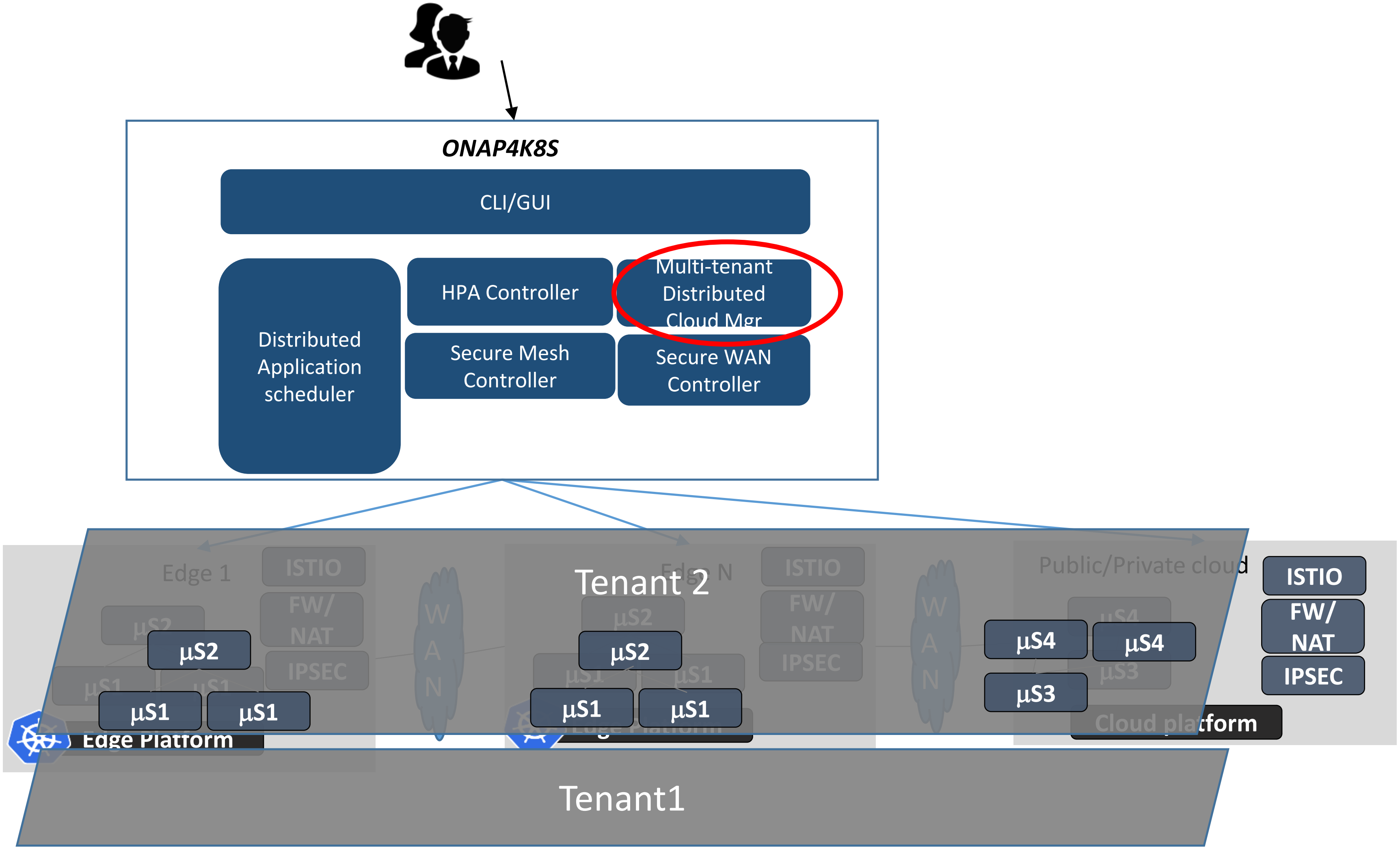
To enable secure communication among microservices in different locations

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?

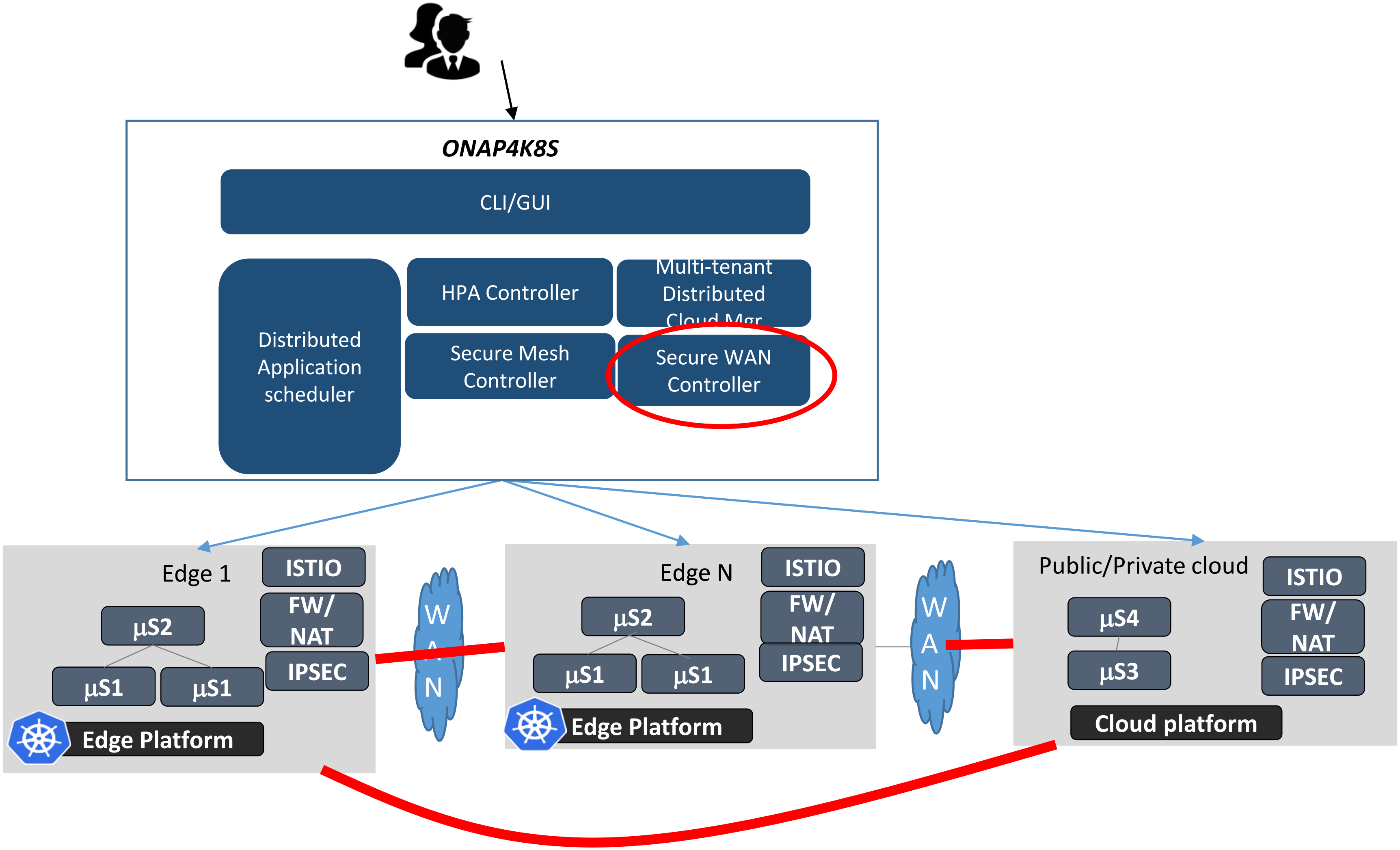
Easy creation of tenants across multiple edges using one user operation

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**

Q&A

ONAP

OPEN NETWORK AUTOMATION PLATFORM

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)

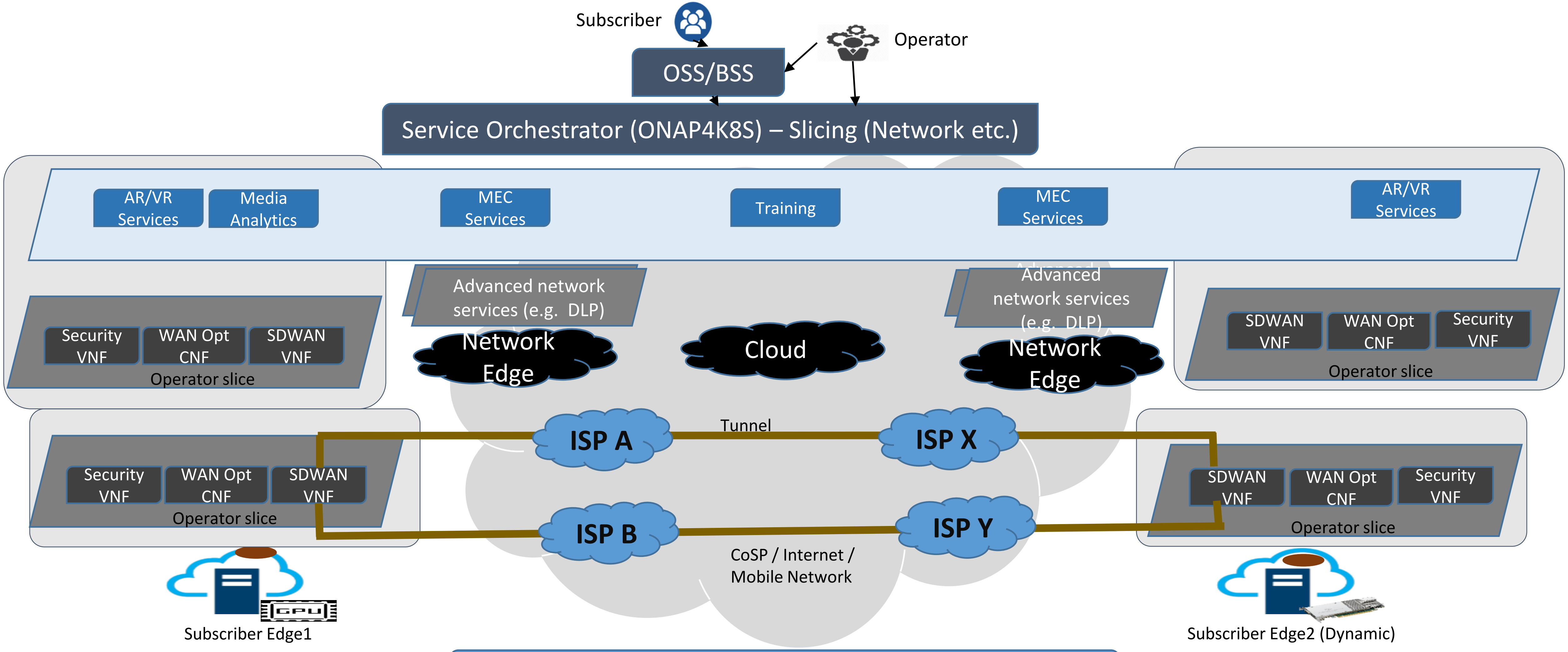
Service Mesh for Micro-services
(Acceleration using Cilium' Kernel bypass among service mesh side cars - e.g. Envoys; and others)

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



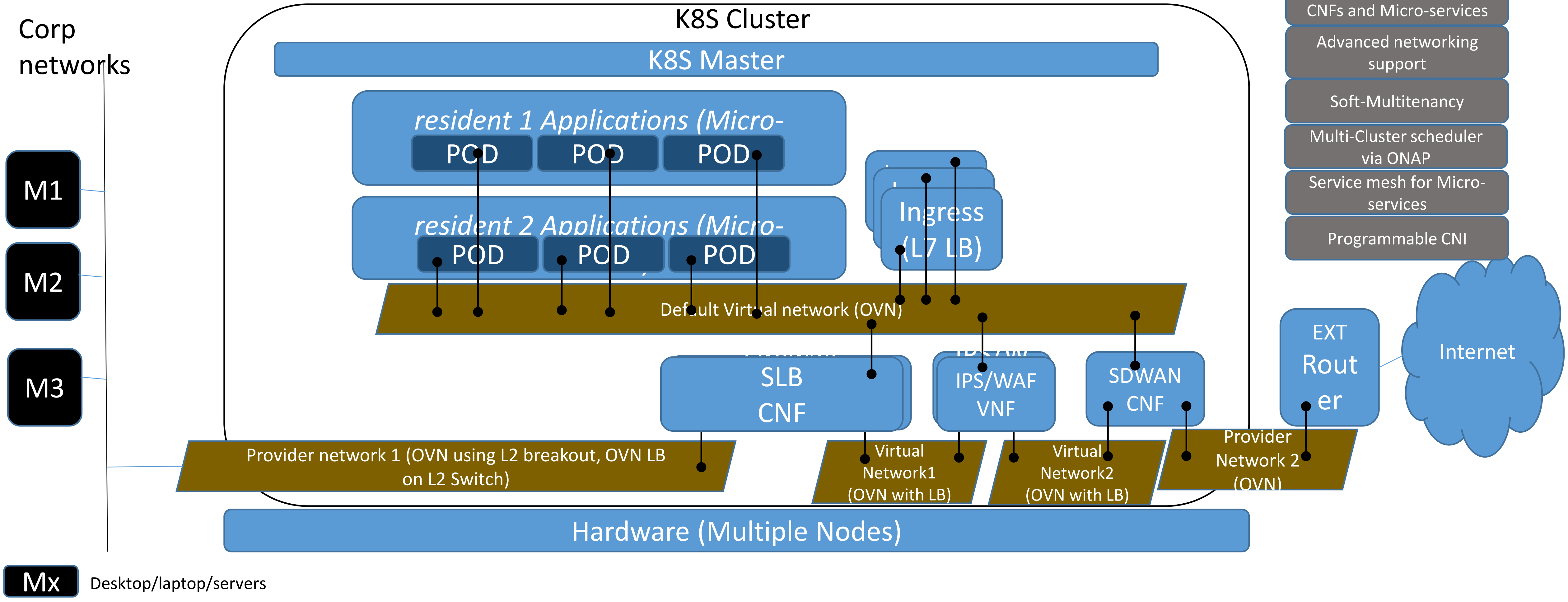
May be tested with dummy apps and OpenWRT, if no real apps from community

● Kubernetes

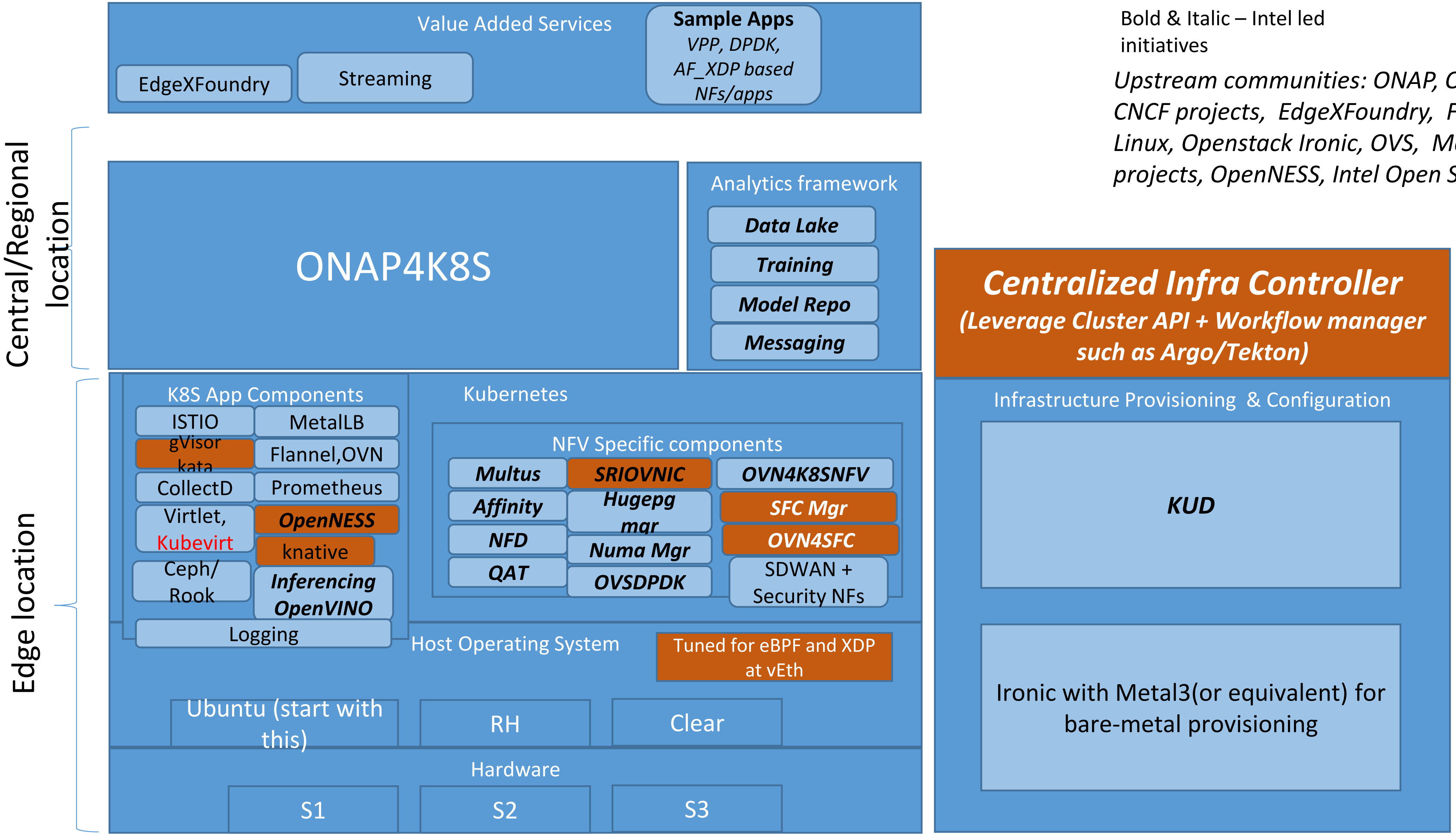
How does NFV based deployment with Cloud-native applications look like (Taking SDWAN with security NFs as an example)

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
Rout
er
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)

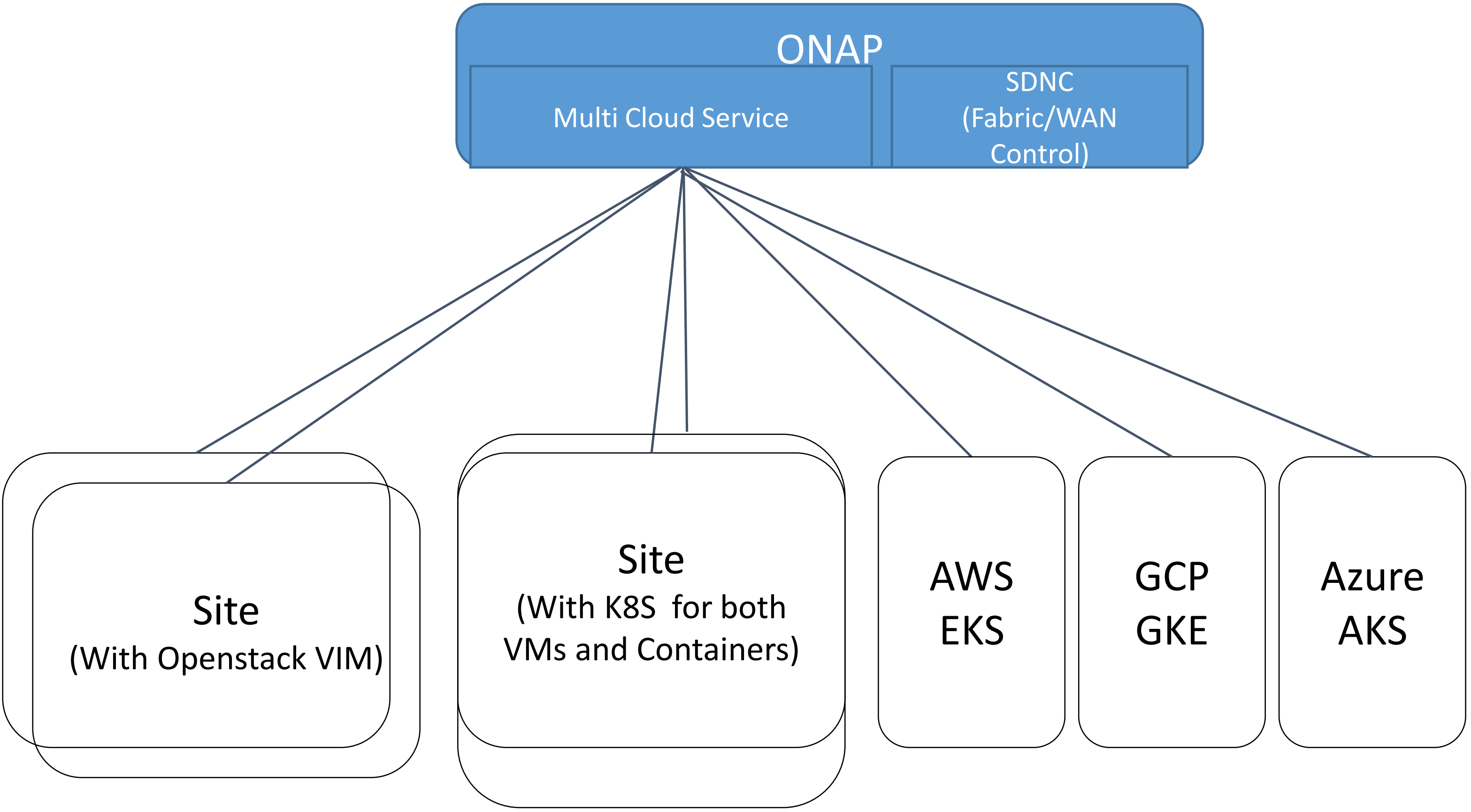


Bold & Italic – Intel led initiatives

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

CNF/VNF support via K8s in ONAP R4/R5

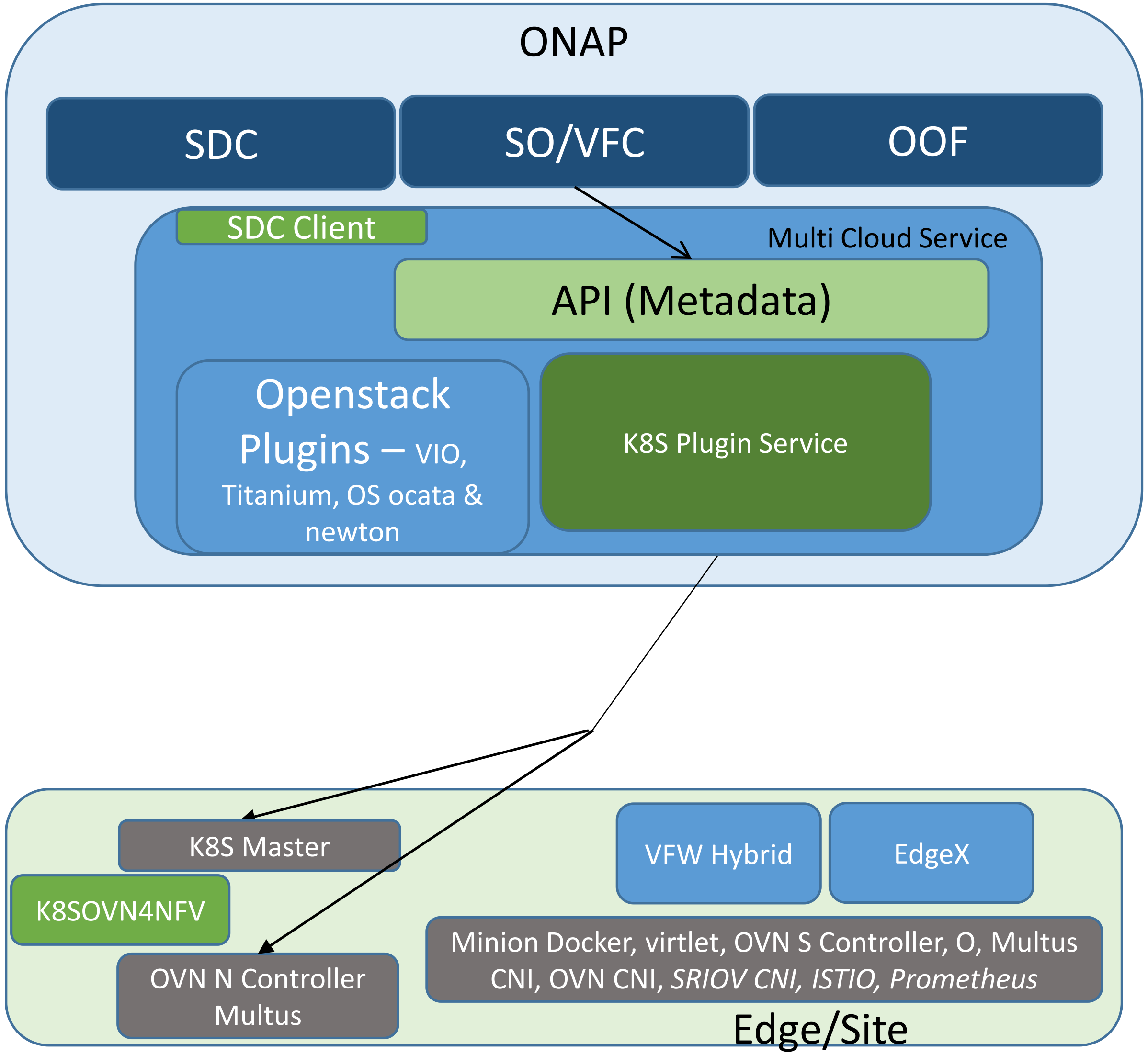
ONAP – Support for K8S based Sites


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)

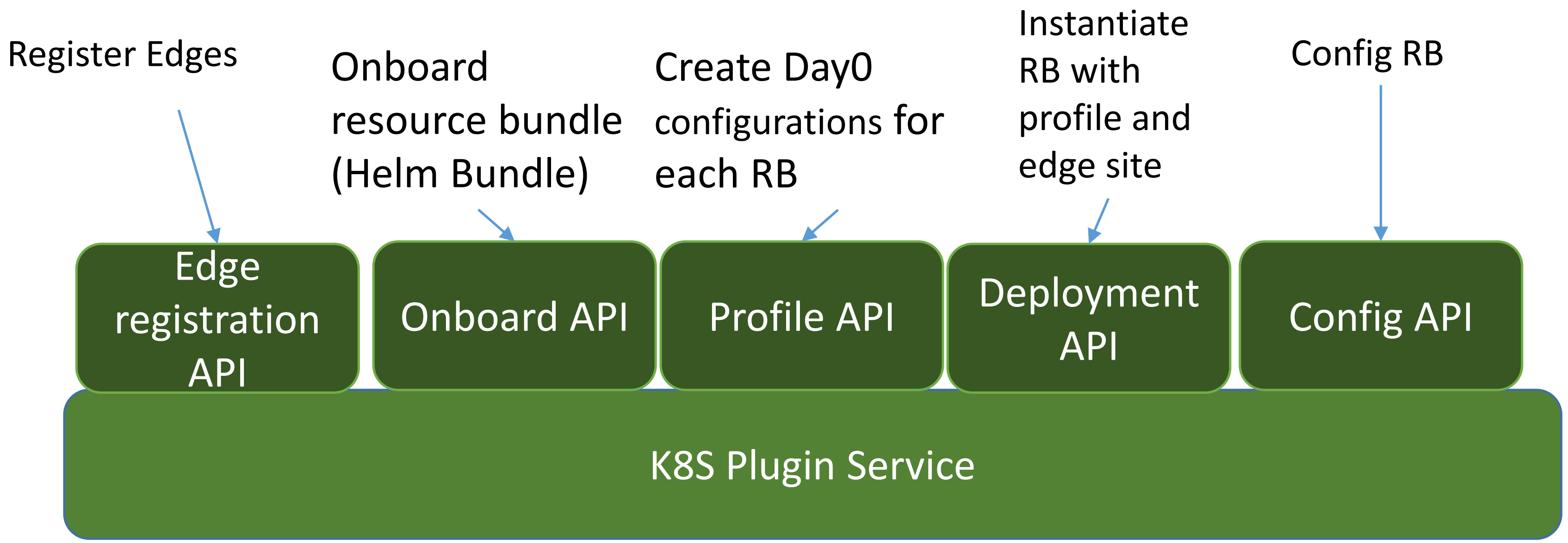
ONAP – K8S Support



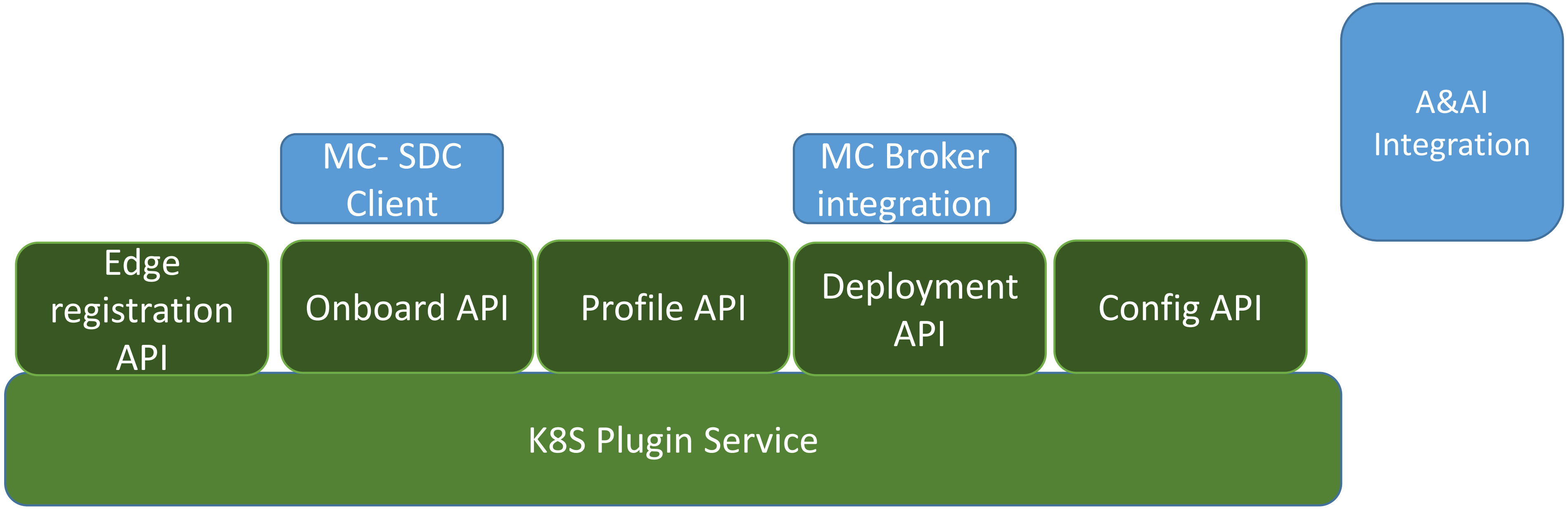
ONAP
SDC
SO/VFC
OOF
SDC Client
Multi Cloud Service
API (Metadata)
Openstack Plugins – VIO, Titanium, OS ocata & newton
K8S Plugin Service
K8S Master
K8SOVN4NFV
OVN N Controller Multus
VFW Hybrid
EdgeX
Minion Docker, virtlet, OVN S Controller, O, Multus CNI, OVN CNI, *SRIOV CNI, ISTIO, Prometheus*
Edge/Site

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. K8S-OVN4NFV (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 containers and with rest of ONAP

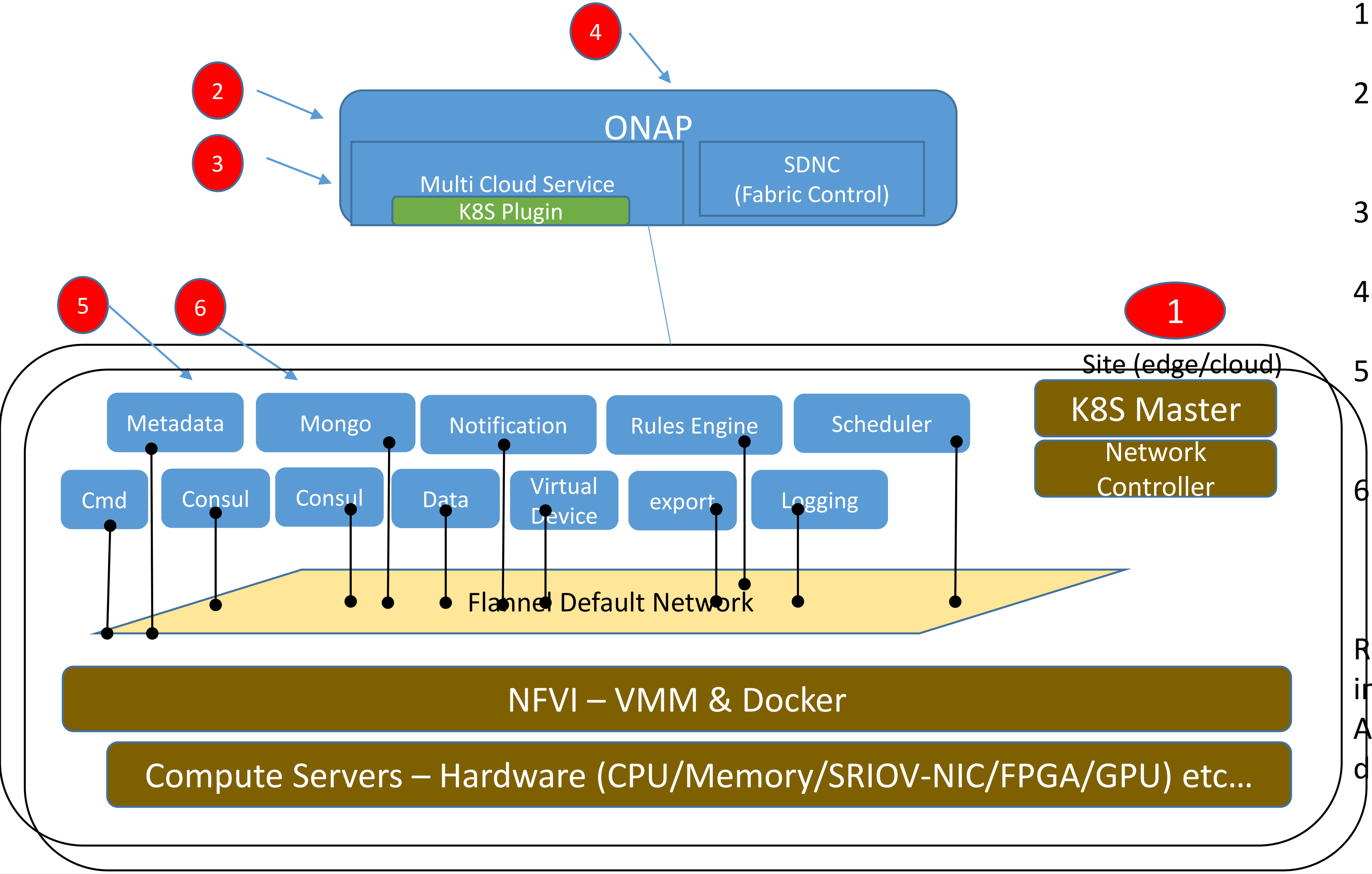
K8S Plugin Service as independent manager – Modular design


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



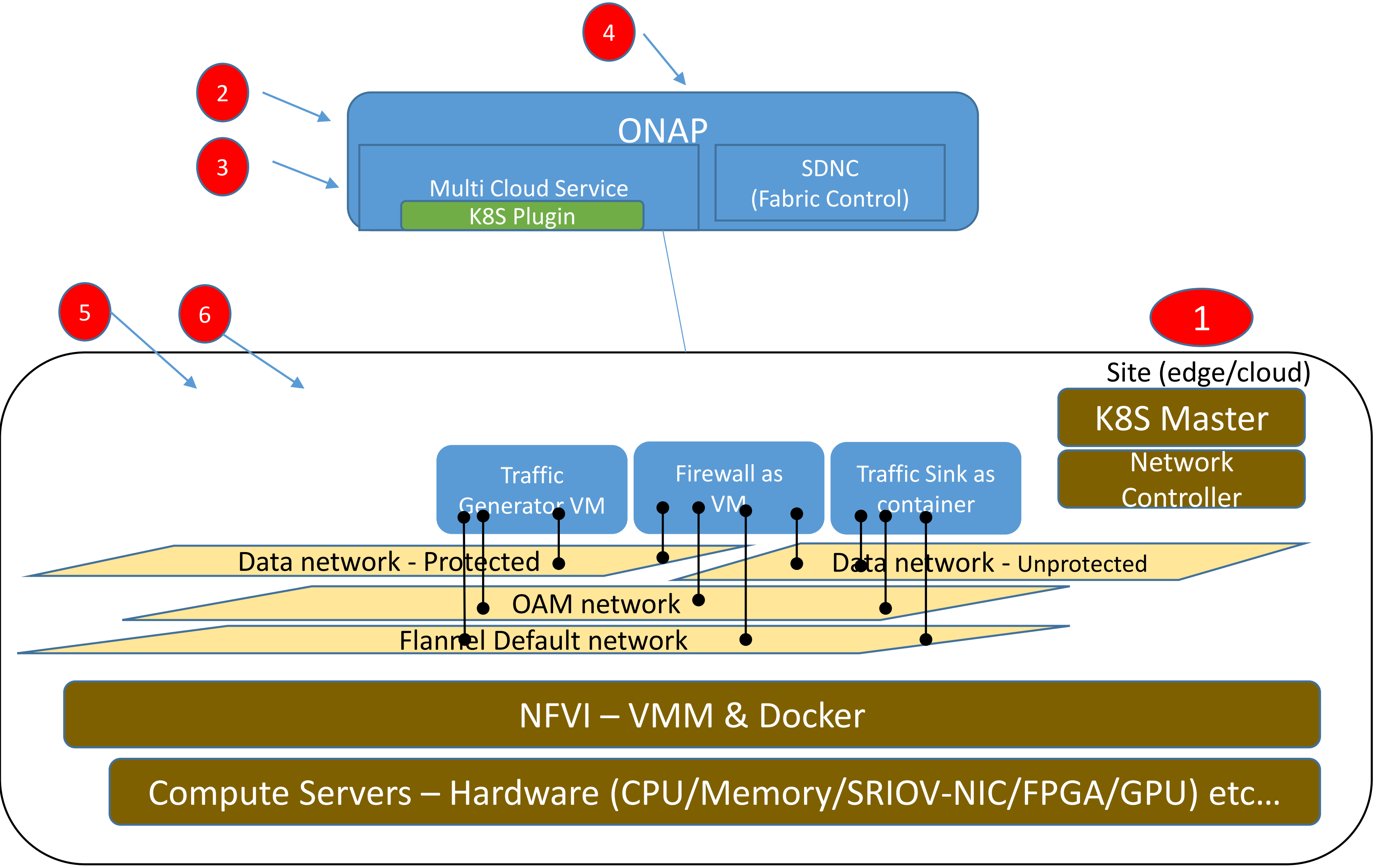
R4 Scenarios – EdgeX deployment



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)



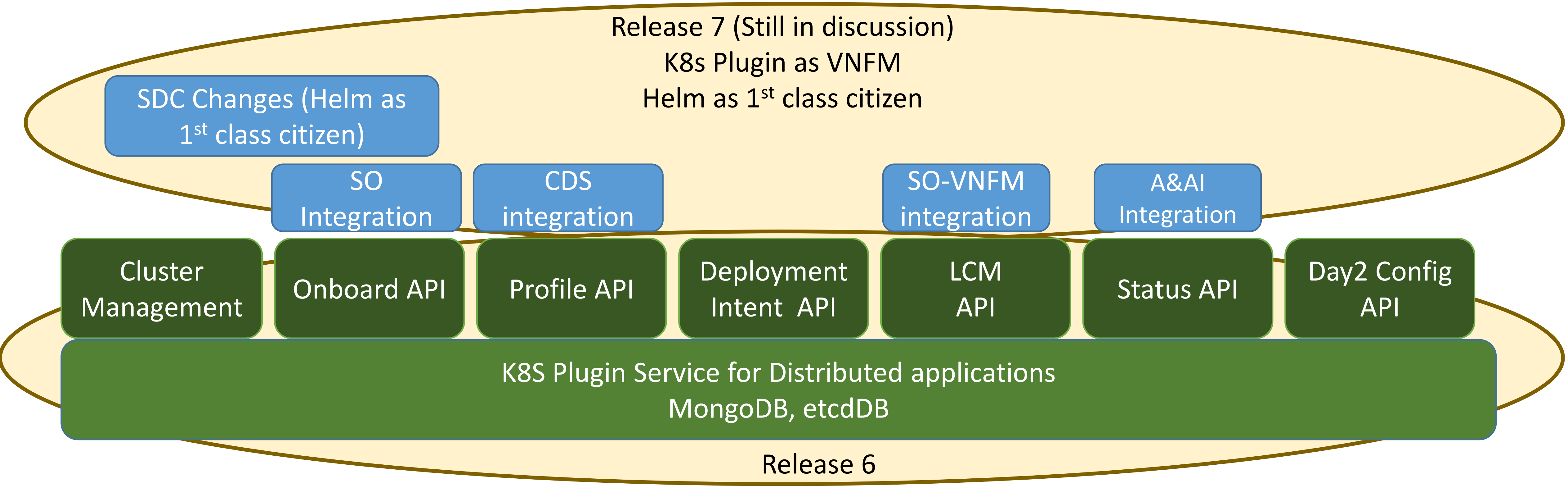
1. One time: Prepare K8S based site using KRD (if it does not exist)
2. One time: Register the K8S Site in ONAP by adding Kubeconfig file in ONAP (if the site is not added earilier)
3. vFirewall onboarding: Create deployment and service yaml
4. Instantiate vFirewall using SO API (or VID GUI)
5. Check 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

R6 and R7 Plan



Release 7 (Still in discussion)
K8s Plugin as VNFM
Helm as 1st class citizen

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 Distributed applications
MongoDB, etcdDB

Release 6