Integrated Cloud Native NFV & App stack - Proposals

AKRAINO EDGE STACK

Integrated Cloud Native NFV & App related blueprints

Use cases	• SDWAN • Distributed Analytics as a service • IOT framework - EdgeXFoundry • Tiled Streaming Video CDN
Blue print family	Integrated Cloud Native NFV & App family
Blueprints	• Multi-Server Cloud native NFV/App stack
Contributors	• Verizon, Intel, MobileEdgeX, Aarna Networks, VMWare

Will submit another BP with MobileEdgeX in this family

Blueprint Proposal: Multi-server Cloud native stack

Case Attributes	Description	Informational
Type	Blueprint	
Blueprint Family - Proposed Name	Integrated Cloud Native NFV stack	
Use Case	SDWAN, Customer Edge, Edge Clouds – deploy VNFs/CNFs and applications as micro-services	
Blueprint proposed Name	Multi-server Cloud Native stack	
Initial POD Cost (capex)	50K minimum	
Scale & Type	Minimum of 4 Xeon Servers + 1 Xeon server as bootstrap node.	
Applications	SDWAN, ML/DL Analytics, EdgeXFoundry and 360 degree Video streaming	
Power Restrictions		
Infrastructure orchestration	Bare Metal Provisioning : ironic or equivalent Kubernetes provisioning : Cluster Operator with KuD. Docker for containers and Virtlet for VMs Service Orchestration : ONAP MEC framework: OpenNESS Site orchestrator : Kubernetes upstream Traffic Orchestration within a cluster: ISTIO Traffic orchestration with external entities : ISTIO-ingress Knative for function orchestration	Expose HW accelerators Storage Orchestration: Ceph with Rook Futrue: AF-XDP for packet processing based containers, OpenShift for site orchestrator, Kubevirt for VMs.
SDN	OVN, SRIOV, Flannel	
Workload Type	Containers, VMs and functions	
Additional Details	Future: eBPF based CNI (such as PolyCube)	

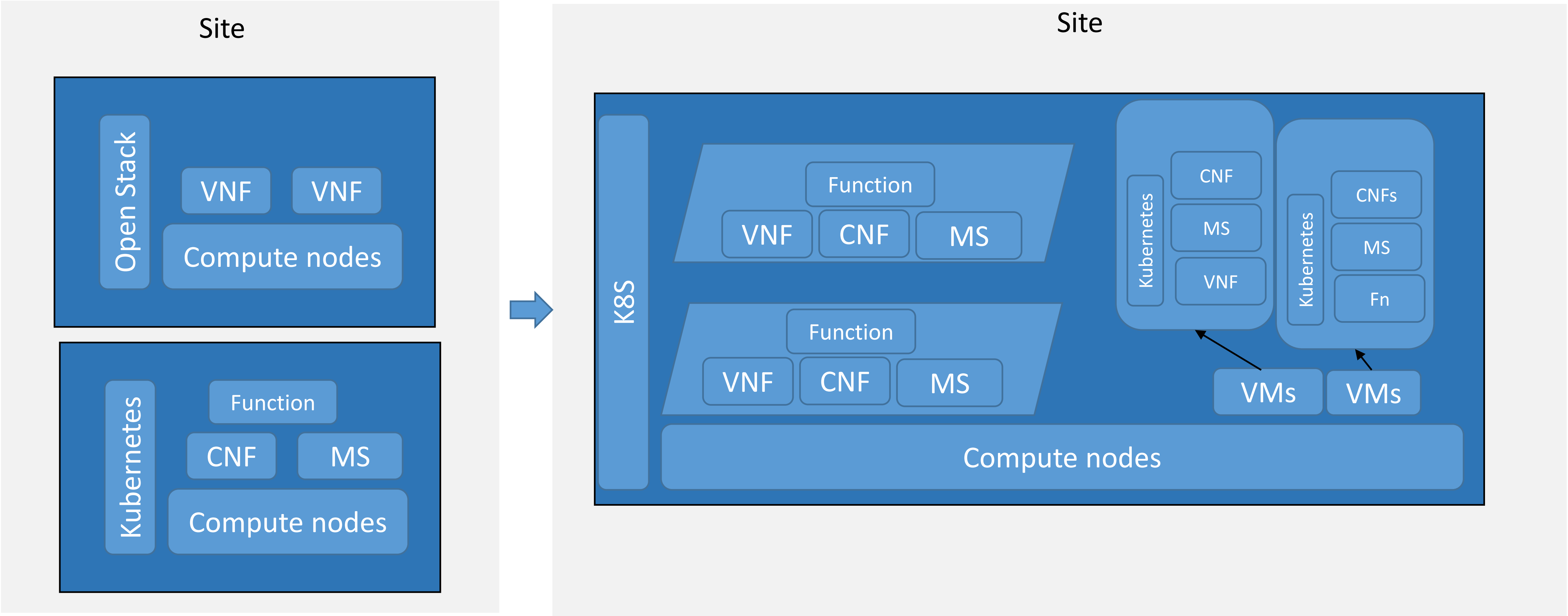
Assessment Criteria

Criteria	Multi-Server Edge NFV/APP stack
Each initial blueprint is encouraged to take on at least two committers from different companies	Verizon, Intel, MobileEdgeX, Aarna Networks, VMWare
Complete all templates outlined in this documents	Detailed in this slide
A lab with exact configuration required by the blueprint to connect with Akraino CI and demonstrate CD. User should demonstrate either an existing lab or the funding and commitment to build the needed configuration.	Continuous Deployment will be provided in Intel community lab (Similar to the way we did for OPNFV)
Blueprint is aligned with the Akraino Edge Stack Charter	All opensource, Edge use case, Aligned with the Akraino Charter
Blueprint code that will be developed and used with Akraino repository should use only open source software components either from upstream or Akraino projects.	Yes, all open source. Actual applications (use cases) in some cases are from third party
For new blueprints submission, the submitter should review existing blueprints and ensure it is not a duplicate blueprint and explain how the submission differs. The functional fit of an existing blueprint for a use case does not prevent an additional blueprint being submitted.	An comprehensive platform with all kinds of deployment types – VNFs, CNFs, Micro-Services and functions, Advanced networking support, Multi-tenancy, slicing, OVN based data interfaces

AKRAINO EDGE STACK

Criteria	Multi-Server Edge NFV/APP stack
Name of the project is appropriate(no trademark issues etc.); Proposed repository name is all lower-case without any special characters.	Integrated Cloud Native NFV & App stack
Project contact name, company, and email are defined and documents	Identified. Will be documenting them in the page
Description of the project goal and its purpose are defined.	Developing an integrated cloud native stack solution for VNFs, CNFs
Scope and project plan are well defined.	For release 2. Yes. Various milestones with the release are identified.
Resource committed and available	There is a team, resources and lab in place.
Contributors identified	Verizon, Intel, MobileEdgeX, Aarna Networks, VMWare
Initial list of committers identified (elected/proposed by initial contributors)	Yes. PTL and 4 Engineers are identified.
Meets Akraino TSC policies	The project will operate in a transparent, open, collaborative, and ethical manner at all the times.
Proposal has been socialized with potentially interested or affected projects and/or parties	○ Have already reached a consensus with sponsors
Cross Project Dependencies.	CNCFs Projecs, K8s, CRI, OCI, Virtlet, Kubevirt, Kata container, gVisior, OpenWRT, Docker, OVN, OVS, DPDK, AF_XDP, ONAP

Transformation journey (to Kubernetes)



Two different resource orchestrators
Compute nodes are divided

- K8S for VNFs, CNFs, Micro-Services and functions
- Soft Multitenancy with one K8S
- Strict Multitenancy with K8S clusters from VMs (when required)

Goals of ‘Integrated Cloud Native NFV & App’ BPs

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)

Multi Site Scheduler using ONAP
(Auto Edge registration, Workload placement, On-demand tenant/slice creation)

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

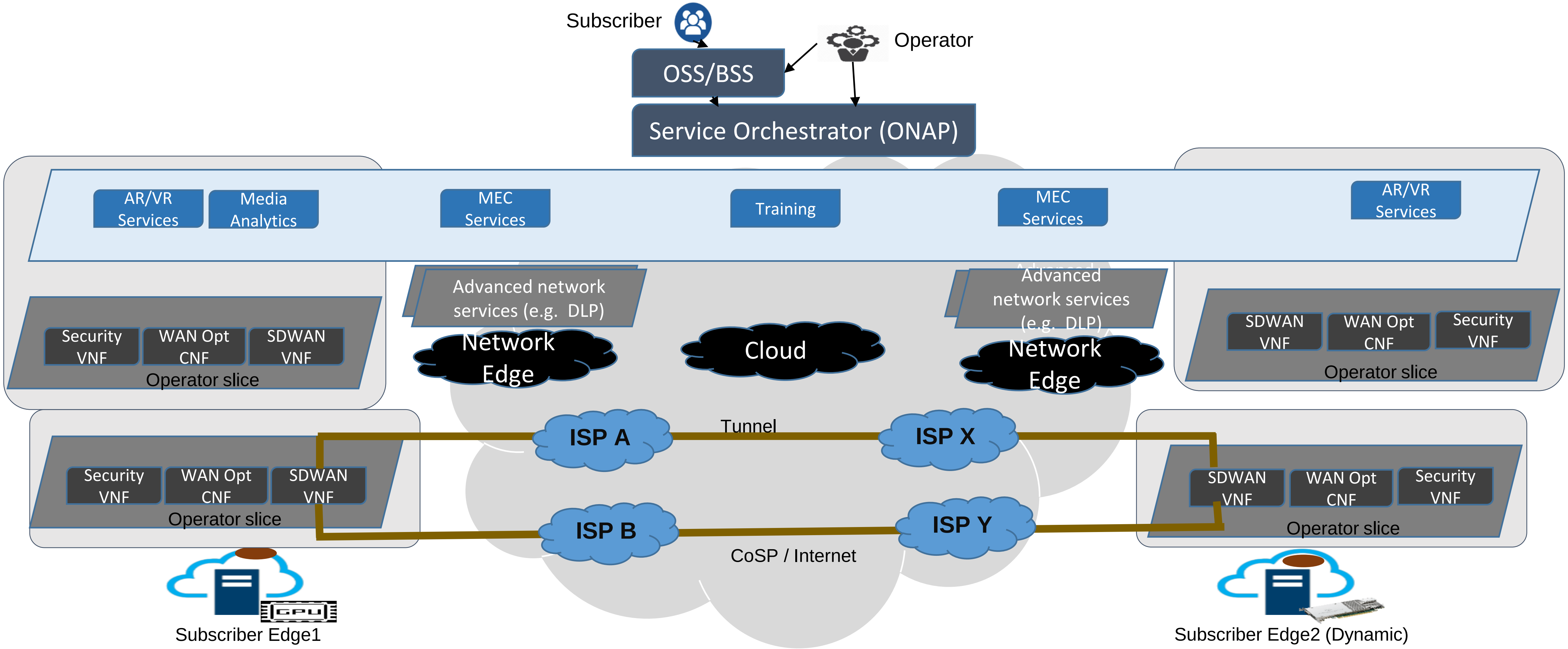
Programmable CNI
(to allow SFC and avoid multiple protocol layers)

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

Prove with either test cases or use cases

Managed SDWAN use case

View in slide show mode



Subscriber

Operator

OSS/BSS

Service Orchestrator (ONAP)

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

Tunnel

ISP A

ISP X

ISP B

ISP Y

CoSP / Internet

Security VNF

WAN Opt CNF

SDWAN VNF

Operator slice

SDWAN VNF

WAN Opt CNF

Security VNF

Operator slice

GPU

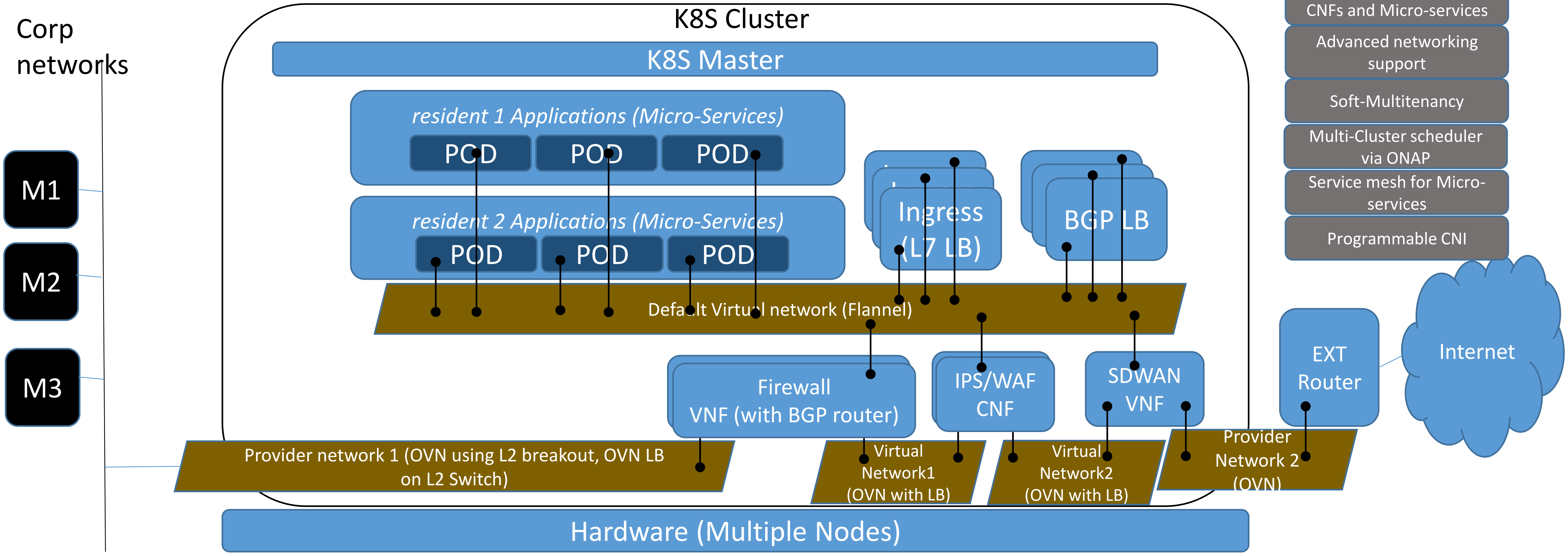
Subscriber Edge1

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)


Corp networks
M1
M2
M3
K8S Cluster
K8S Master
resident 1 Applications (Micro-Services)
POD
POD
POD
resident 2 Applications (Micro-Services)
POD
POD
POD
Ingress (L7 LB)
BGP LB
Default Virtual network (Flannel)
Firewall VNF (with BGP router)
IPS/WAF CNF
SDWAN VNF
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)

What it proves

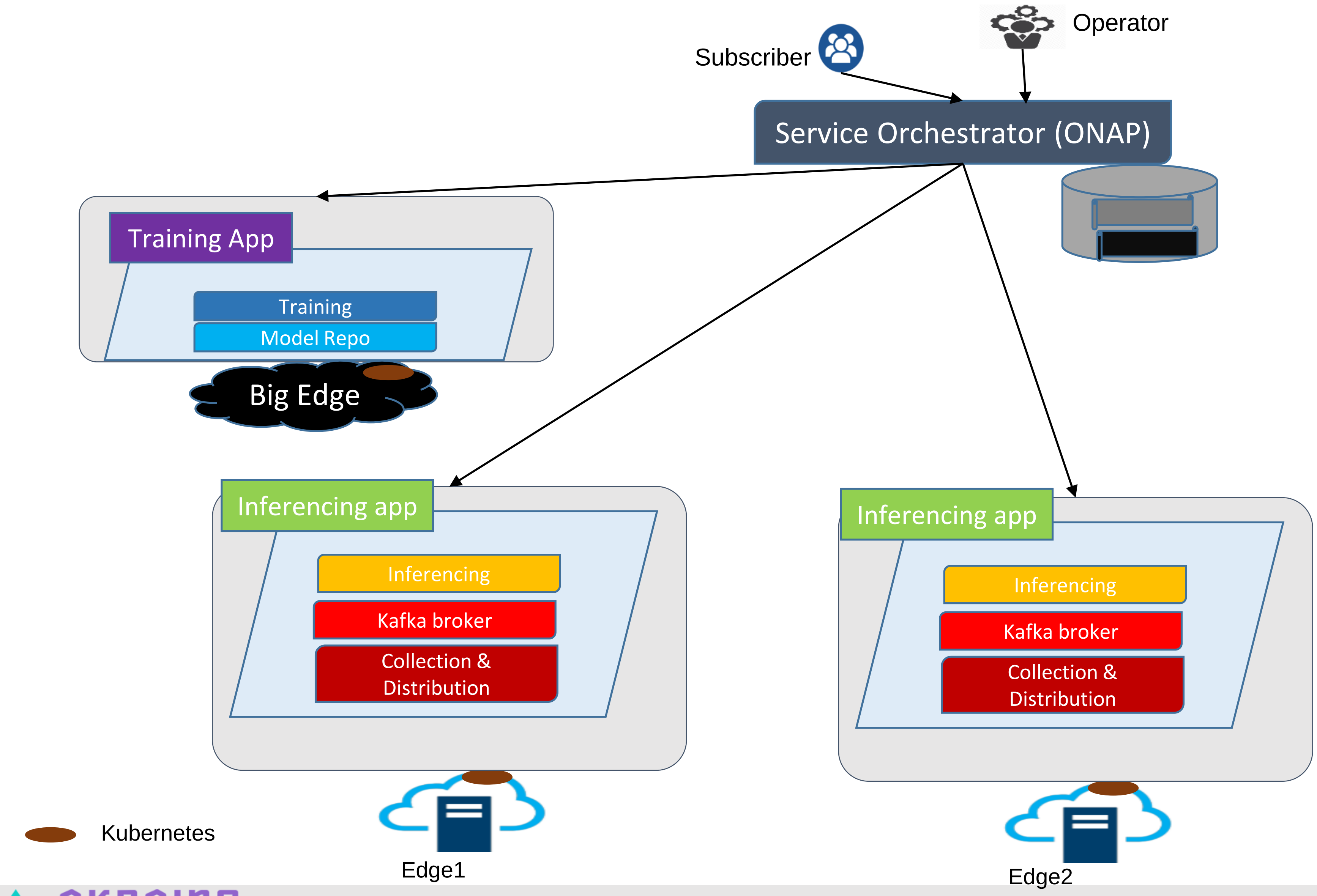
- Coexistence of VNFs, CNFs and Micro-services
- Advanced networking support
- Soft-Multitenancy
- Multi-Cluster scheduler via ONAP
- Service mesh for Micro-services
- Programmable CNI

Initial testing to be done using Ubuntu for all VNFs/CNFs.

Mx Desktop/laptop/servers

Distributed Analytics as a Service
(Each site to have self contained inferencing and few sites with training)

View in slide show mode



Subscriber

Operator

Service Orchestrator (ONAP)

Training App

Training

Model Repo

Big Edge

Inferencing app

Inferencing

Kafka broker

Collection & Distribution

Edge1

Inferencing app

Inferencing

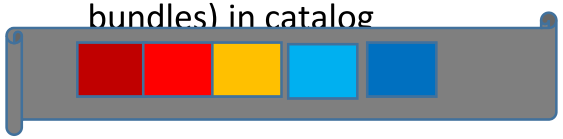
Kafka broker

Collection & Distribution

Edge2

Kubernetes

1. Onboard Analytics framework (With 6 bundles) in catalog



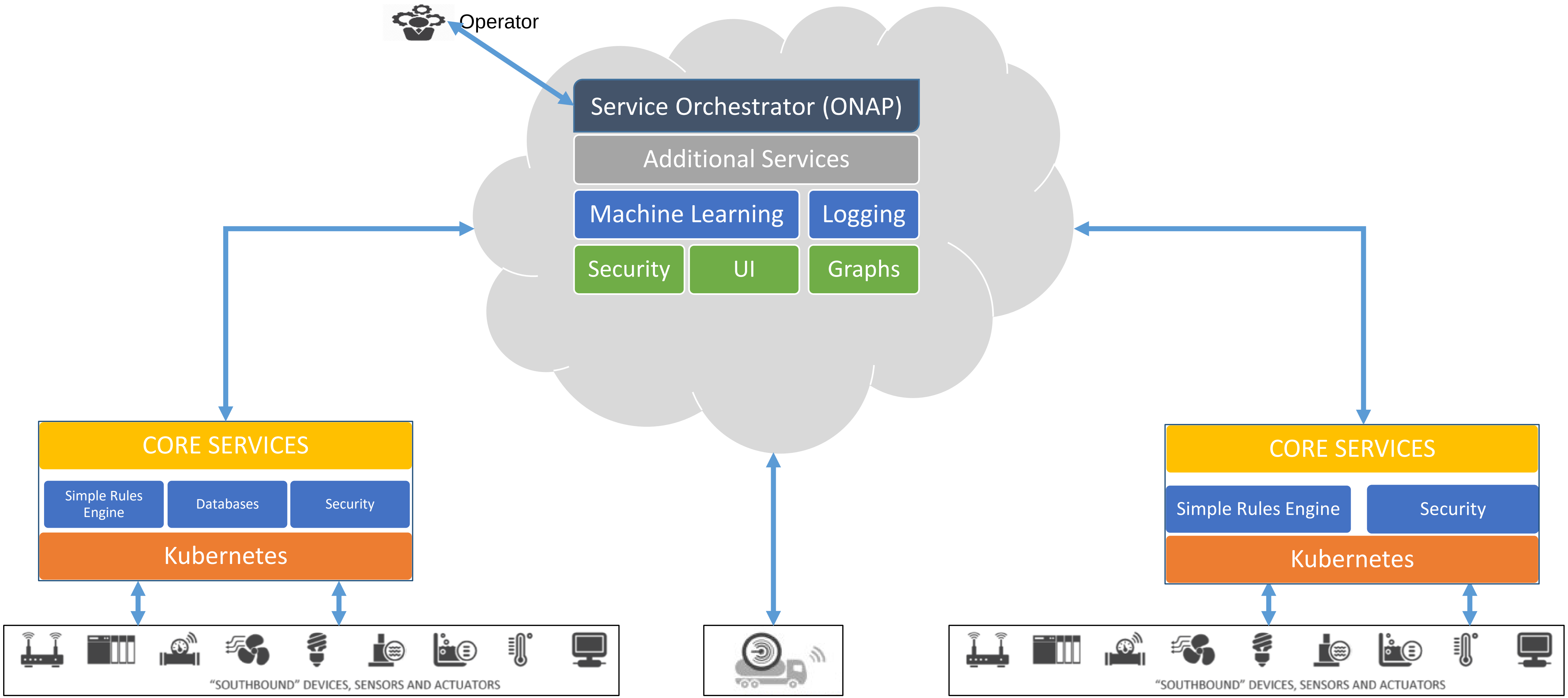
2. Activate framework

3. Onboard an analytics app

4. Activate analytics app

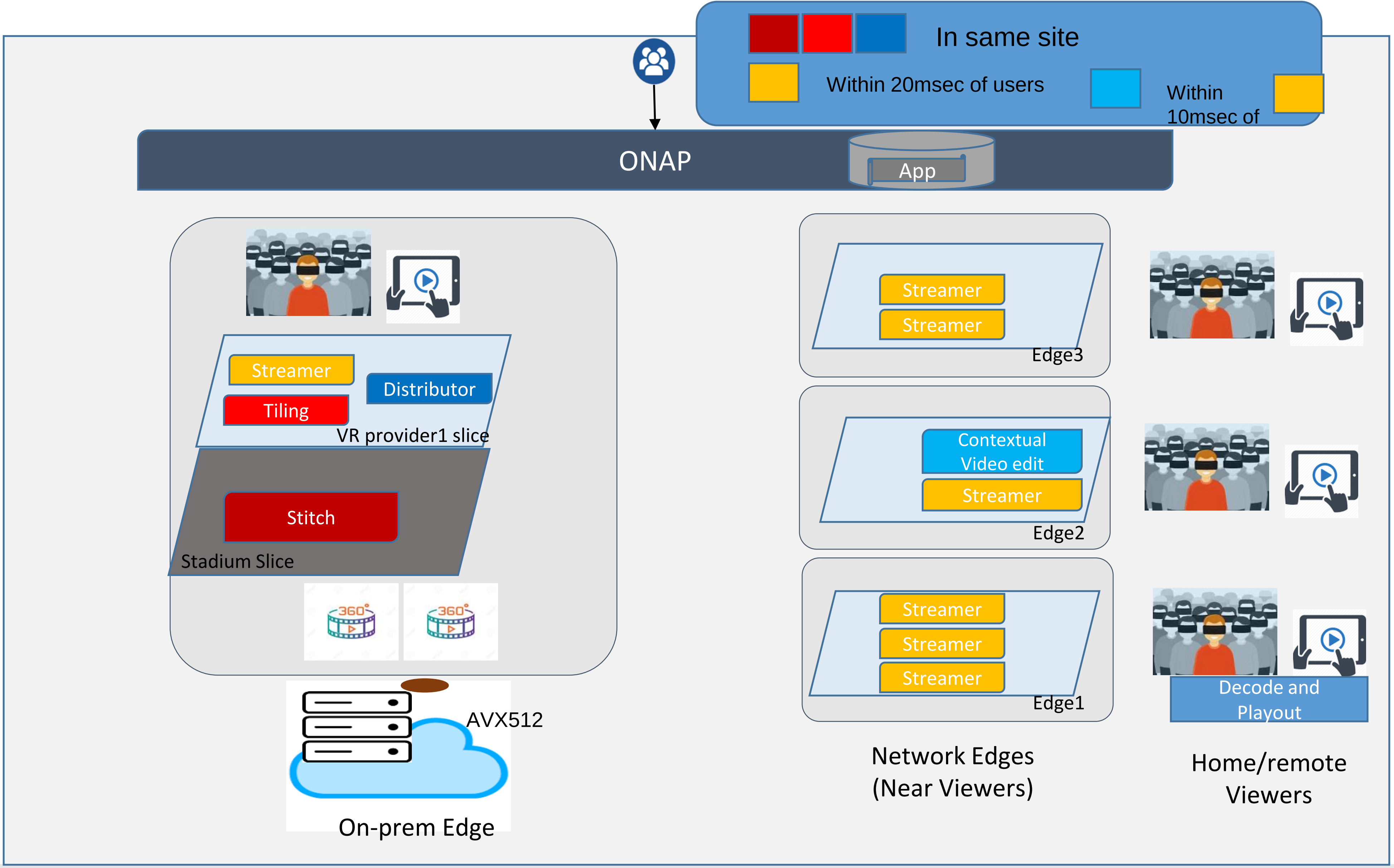
Federated Learning (future)

EdgeXFoundry use case

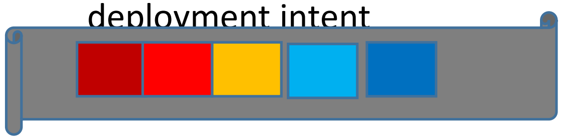


VR 360 streaming– Enable remote users to view the events/games via Edge-compu

View in slide show mode



1. Onboard App (With 6 services) in catalog with deployment intent



2. Active App (when event starts)

3. New users join the event (Auto bring up of services at Edge1 and Edge3)

4. Users join in a geo that requires additional context to be added to the stream.

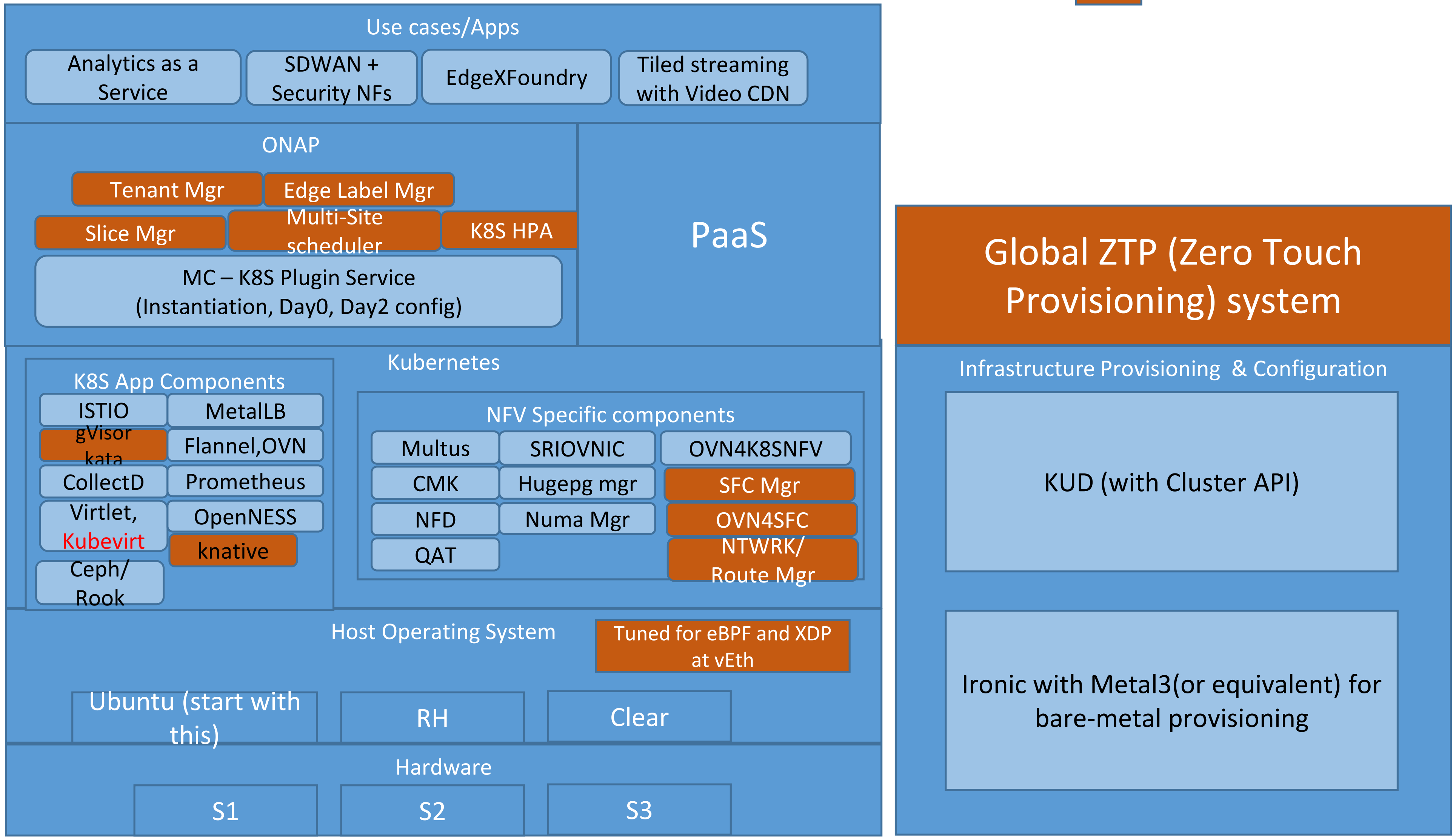
5. More users join near edge1

6. Users disappear Edge2

AKRAINO EDGE STACK

Cloud Native App & NFV Stack – Putting it all together

🟧 Future



Use cases/Apps

Analytics as a Service

SDWAN + Security NFs

EdgeXFoundry

Tiled streaming with Video CDN

ONAP

Tenant Mgr

Edge Label Mgr

Slice Mgr

Multi-Site scheduler

K8S HPA

MC – K8S Plugin Service (Instantiation, Day0, Day2 config)

PaaS

Global ZTP (Zero Touch Provisioning) system

Kubernetes

K8S App Components

ISTIO

MetalLB

gVisor kata

Flannel,OVN

CollectD

Prometheus

Virtlet, Kubevirt

OpenNESS

knative

Ceph/ Rook

NFV Specific components

Multus

SRIOVNIC

OVN4K8SNFV

CMK

Hugepg mgr

SFC Mgr

NFD

Numa Mgr

OVN4SFC

QAT

NTWRK/ Route Mgr

Infrastructure Provisioning & Configuration

KUD (with Cluster API)

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

Host Operating System

Tuned for eBPF and XDP at vEth

Ubuntu (start with this)

RH

Clear

Hardware

S1

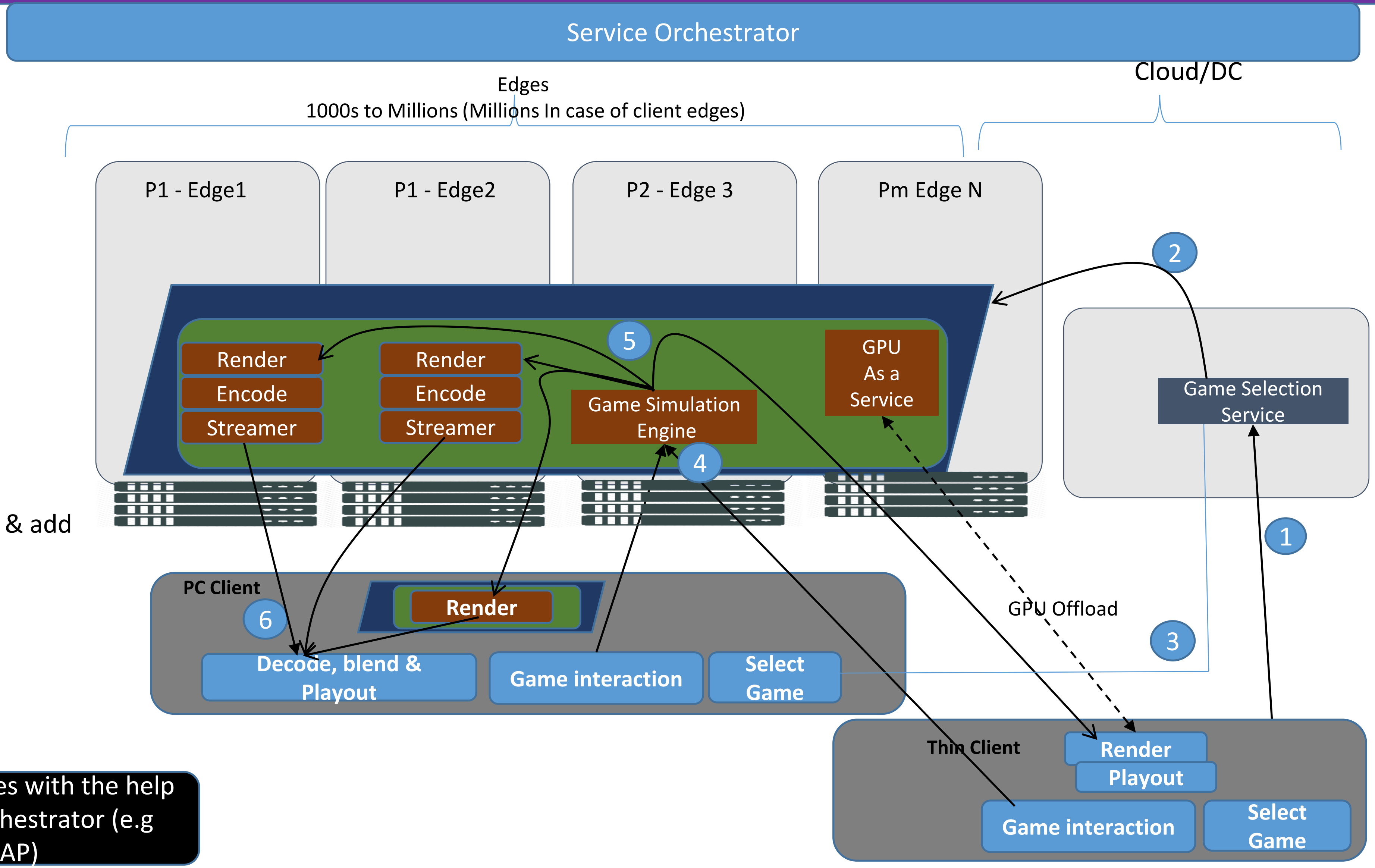
S2

S3

BACKUP

Edge Computing – Cloud gaming app example

1. Game Selection
2. Slice bring up and App bring up
3. New player joins up, expand slide & add new micro-services
4. Game interaction
5. Distributed Rendering
6. Playout



Px Edge Provider x

□ Edge

▨ Slice

■ Game Services

Bring up services with the help of service orchestrator (e.g ONAP)

Thank you

AKRAINO EDGE STACK