redhat.

A CLOUD NATIVE BLUEPRINT FOR ULTRA/FAR EDGE

Open Networking Summit

Azhar Sayeed
asayeed@redhat.com
Chief Technologist

DISCLAIMER

These slides are created for the strict purpose of discussion only. Red Hat Inc. does not plan or commit to offer any functionality described in this material as part of its portfolio of products.

Red Hat Inc. does upstream open source software development work and will continue to participate in upstream activities and communities relative to technologies described in this presentation. This does not imply Red Hat will or have products associated with those software communities Nor is Red Hat committing to having such offerings in future.

Red Hat Inc. does not commit to or subscribe to any opinions, statements regarding third parties described in this presentation. Any disclosures associated with third party companies are a matter of conjecture and should not be taken as known facts or stated strategy of those third parties.

Acknowledgements: I would like to acknowledge the contributions of my colleagues at Red Hat. I have used some of their material in this presentation to make my point.

AGENDA

- Edge Classification/Definition
- Edge Compute Use Cases – Far Edge and Ultra-Far Edge
  - Telco Use Cases
  - Enterprise Use Cases
- Requirements
- Size and Volume
- Architecture
  - Distributed Compute Environment
  - Independent Clusters & Federation
  - Discovery and Auto Configuration
  - Assurance
  - Workload placement and management
- Summary

What does Edge mean?

"Edge is the next infrastructure paradigm for delivering applications and services closer to the user. Edge allows efficient processing and delivery of time sensitive data"

"Edge refers to the geographic distribution of compute nodes in the network"

"Edge is a distributing computing paradigm in which computation is largely performed on distributed devices sitting closer to end users"

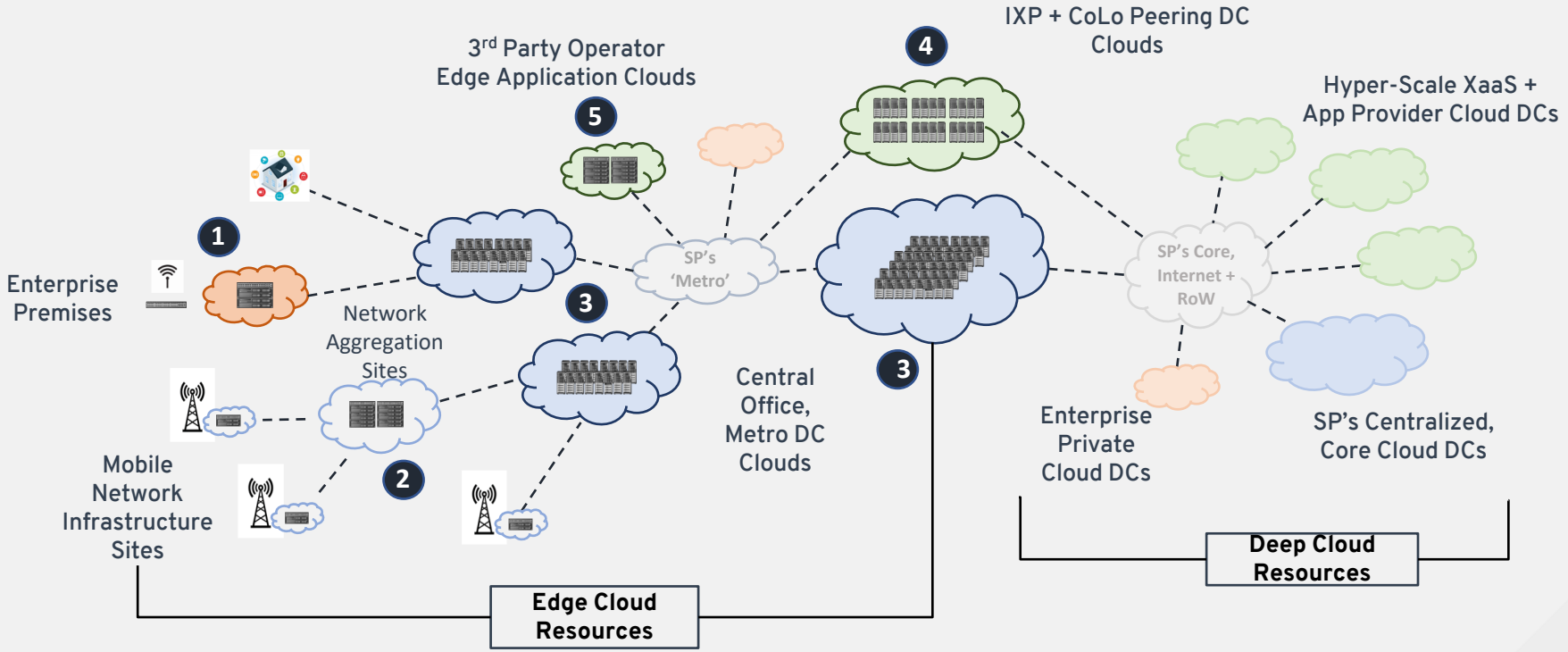
Edge is
- Independent
- Elastic/Optimized
- Massive scale

Edge must be
- Automated
- Resilient
- Be built using public, private or hybrid cloud

Edge Is about building ultra-reliable experiences for people and objects when and where it matters

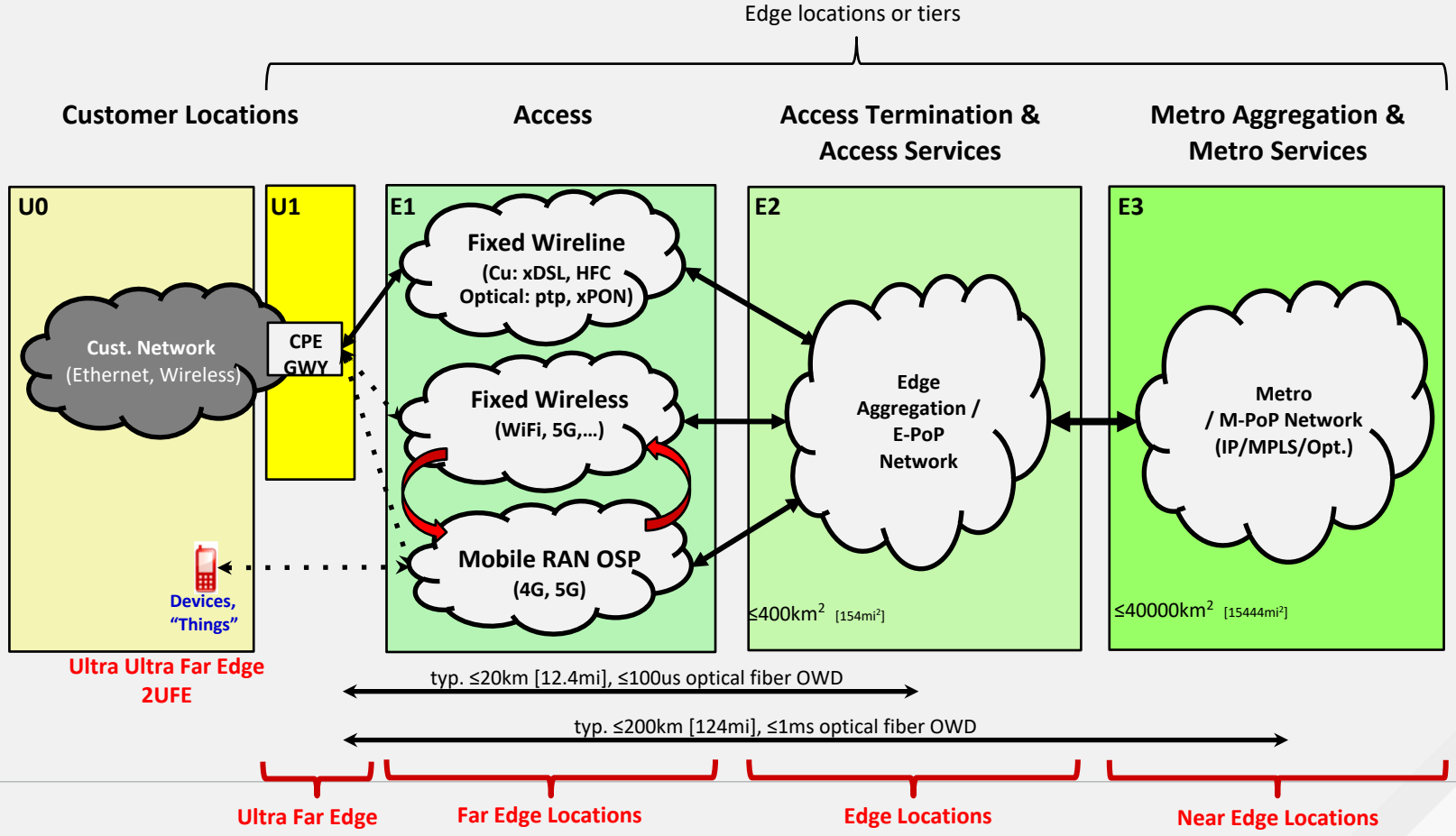
Edge is anything that sits between the Subscriber /User/Endpoint and regional or core data center of the provider

VARIETY OF EDGE DEPLOYMENTS



Courtesy: Paul Johnson - ACG

EDGE CLASSIFICATION



ULTRA/FAR EDGE USE CASES & SIZES - SUMMARY

Use Case	Layer	Descriptions	# of Nodes	Comments
Embedded Compute	U0	Managed Embedded compute	1	Embedded devices - 2UFE (Ultra Ultra Far Edge)
vCPE	U1	Branch Office Connectivity	1-2	Managed CPE service with VNFs for SD-WAN or Branch Office
IoT Gateway	U1/E1	IoT Gwy for SCADA, Protocol/Messaging conversion and data analytics, Atom class	1-2	Usually ruggedized devices collecting sensor data – low cost
Enterprise Edge	U1/E1	Xeon-D class servers	2+	Enterprise Edge – “Edge Cloud”
vRAN	E1	Intel Xeon Class with FPGA	1-2	vDU/DU on baremetal
SDR	U1	Software Radio or RIU	1	

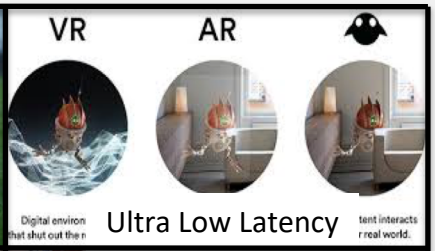
EDGE USE CASES

Edge Use Cases


Manufacturing – I 4.0

Autonomous Vehicles


Drones



Ultra Low Latency


Specialized Apps

Smart City


Emergency Services


Media & Entertainment

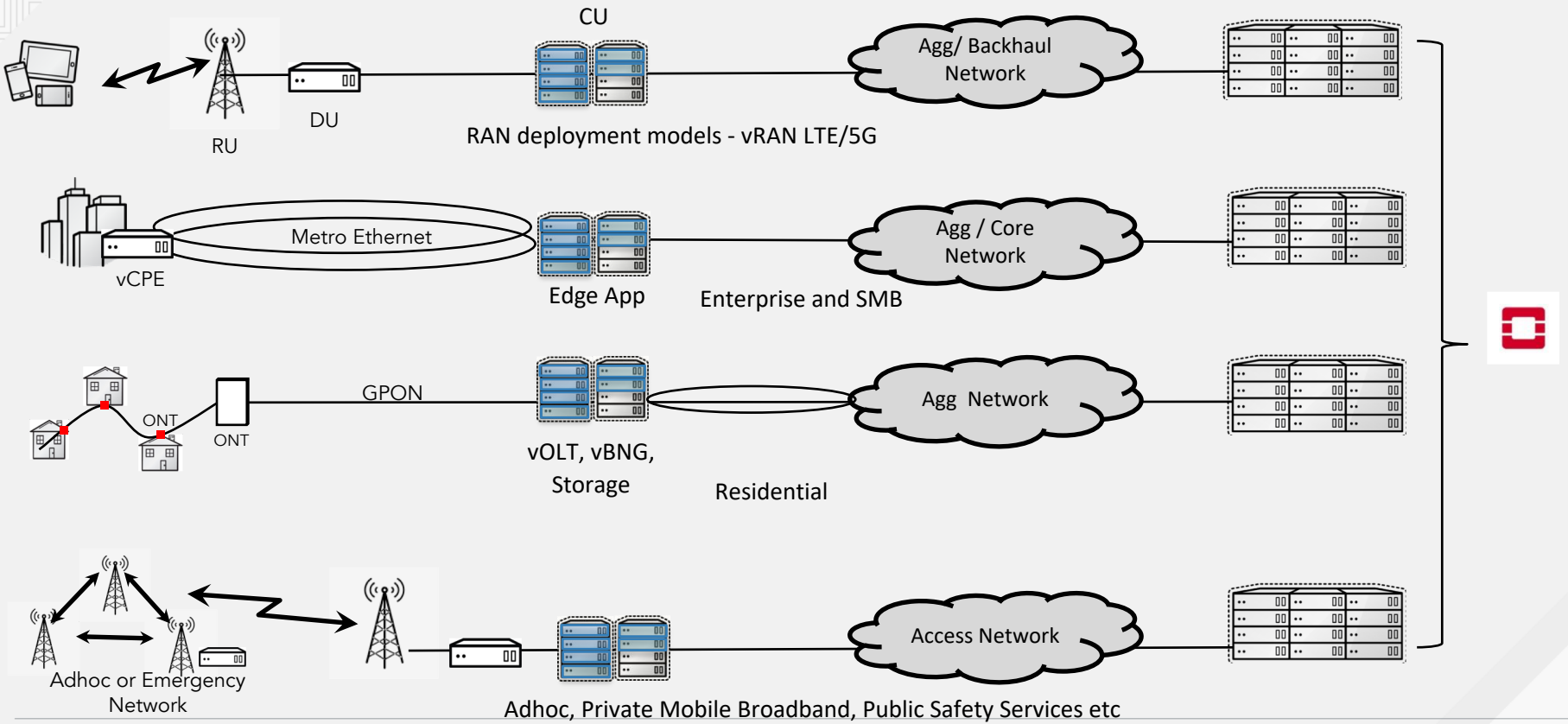

Supply Chain/Logistics

Smart Agriculture


Healthcare


High Speed Broadband

EDGE DEPLOYMENTS SUMMARY



SIZE AND SCALE

Use Case	# of Nodes per location	Scale Network wide	Comments
Enterprise vCPE	1-2	50K+	Managed and Unmanaged CPE business or Enterprise Gateways
vRAN	1-2	200K+	Averaged to 6 sectors per DU – varies widely by provider, geography and type – Example China Mobile needs 1M+ nodes
IoT Gateway	1-2	100K+	Common IoT services – the numbers here are unpredictable – dependent on services
Residential GWs	1	1M++	Telco access network interface, customer Fixed/wireless Internet access services, POTS phone(s), Cable/Video services GW

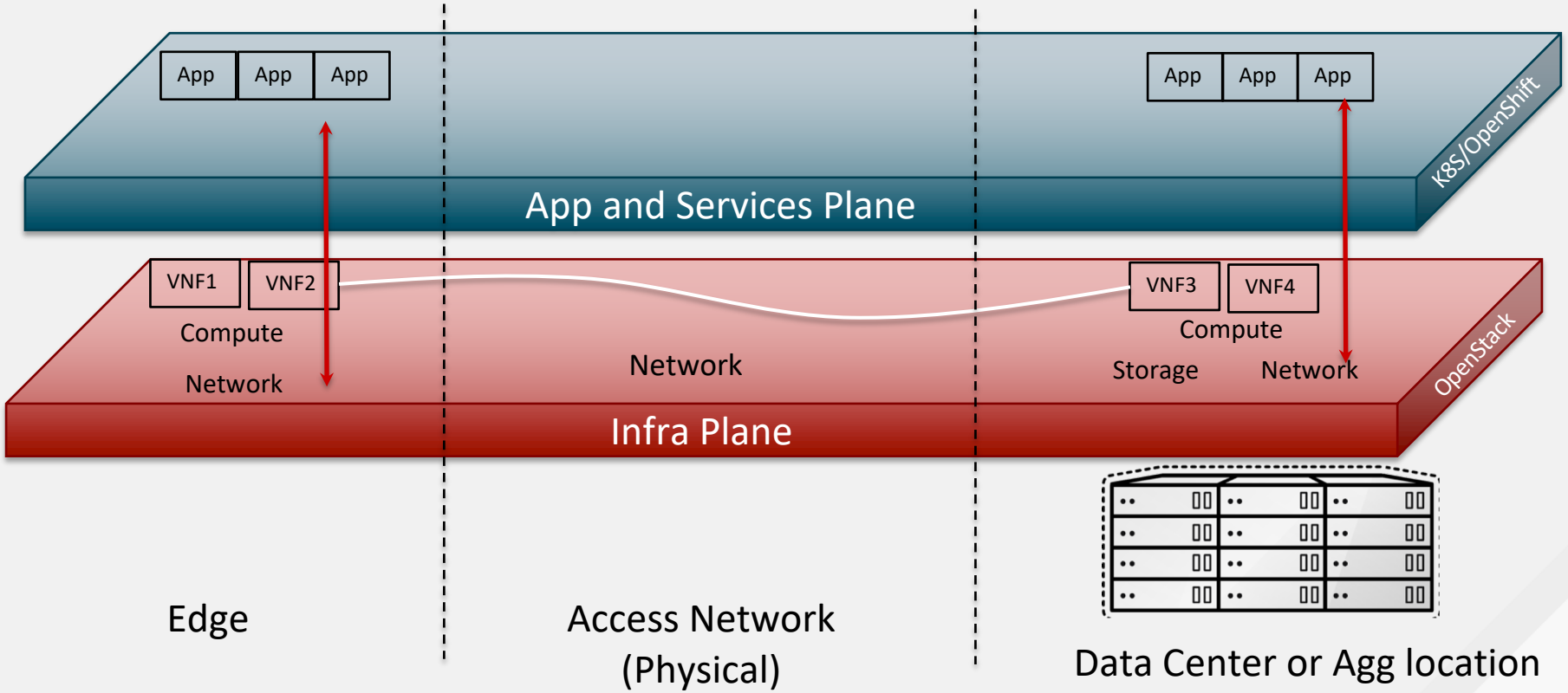
ULTRA FAR/FAR EDGE REQUIREMENTS & ARCHITECTURE

ULTRA FAR/FAR EDGE CHARACTERISTICS AND REQUIREMENTS

- One to a few nodes
- Usually no device level HA – Only Network HA (non redundant configuration)
- Usually Single Tenancy
- Low powered to reasonable powered devices
- Large volumes
  - vDRAN - >500K servers
  - vCPE/Enterprise Managed Service - >50K servers
  - IoT Gateways - >100K servers
  - Enterprise Edge applications - >100K servers
- Centrally provisioned (automated) and centrally managed
- Ephemeral storage at those nodes
- Smart NIC, FPGA, GPU and other accelerators
- Cost/performance competitive

ARCHITECTURE

SEPARATION OF SERVICES FROM INFRA



Edge Scenarios

Capabilities and constraints

OpenStack

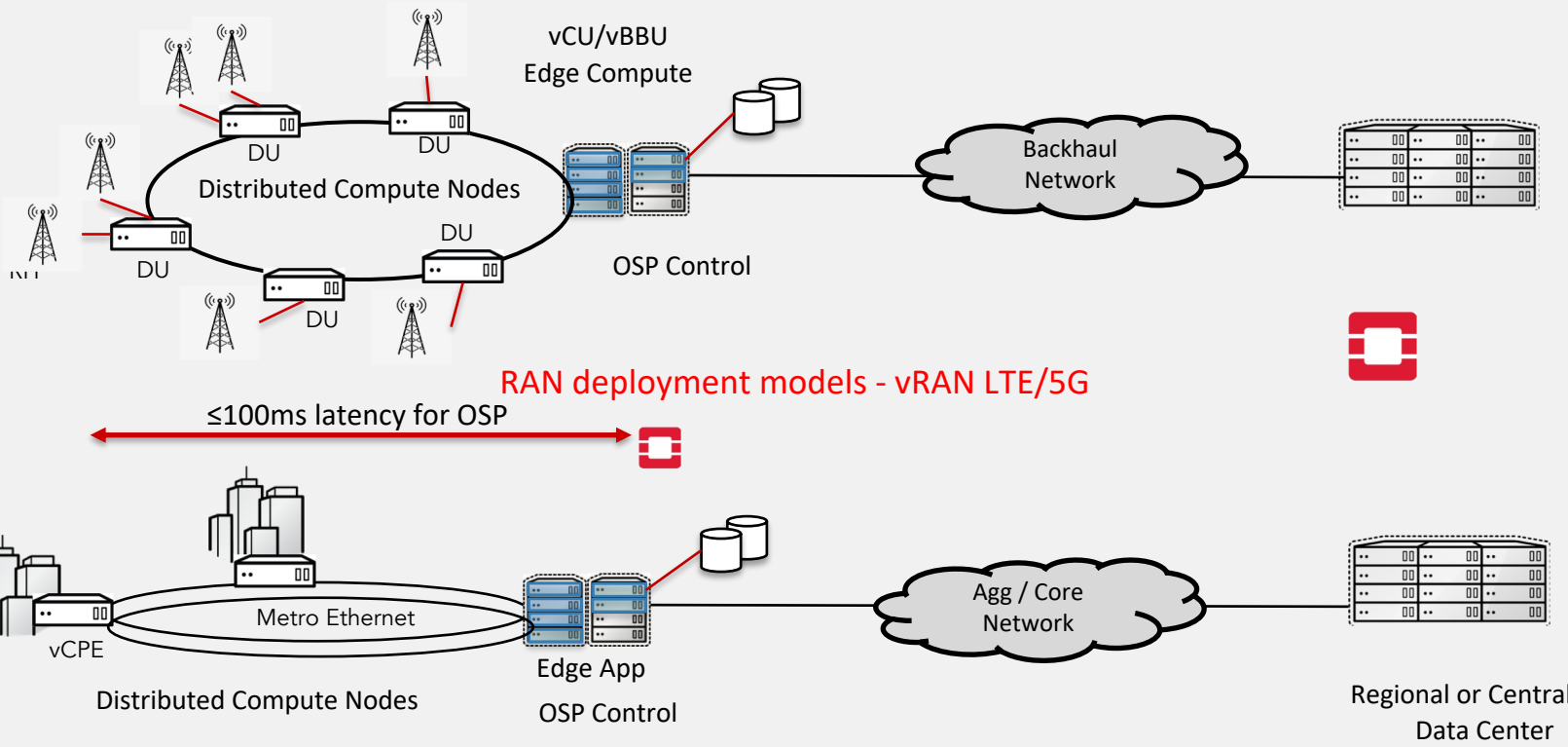
- <100ms Latency for OSP Nova to OSP control Plane
- OSP Director deployable
- Ephemeral Storage
- Uses OpenStack Ironic to initialize and manage the node
- Ironic conductor
- Up to 300 remote nodes per controller cluster
- Support of real time Linux and real time KVM

K8S/OpenShift

- Up to 250ms latency or more for remote worker nodes of K8S
- VMs via Kubevirt in future
- Node/remote cluster must be initialized and available with an IP address
- Up to 1000 nodes possible per controller cluster
- Support of real time Linux

EDGE SCENARIO MAPPING

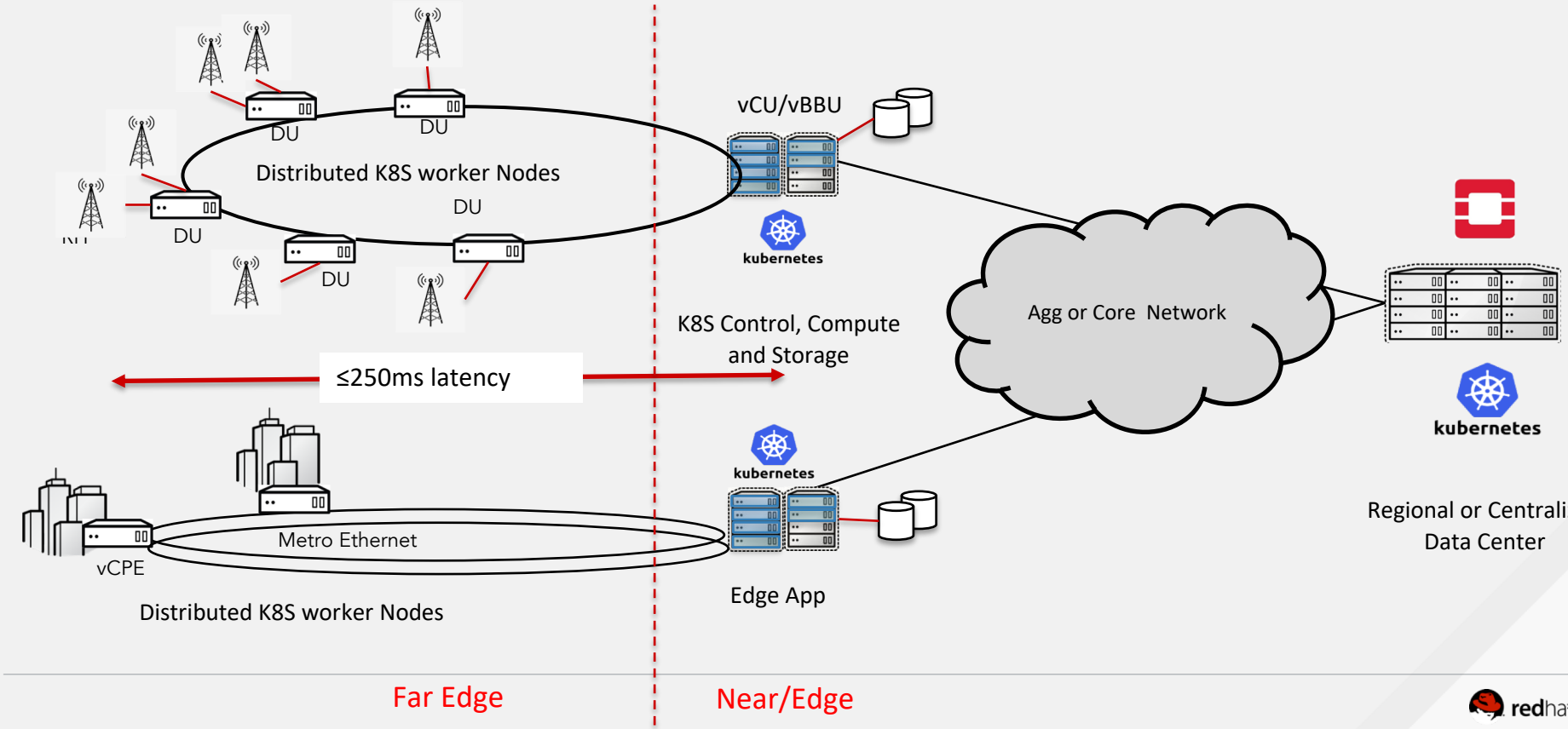
Distributed Compute Node - Edge Site - OpenStack Model



Enterprise and SMB

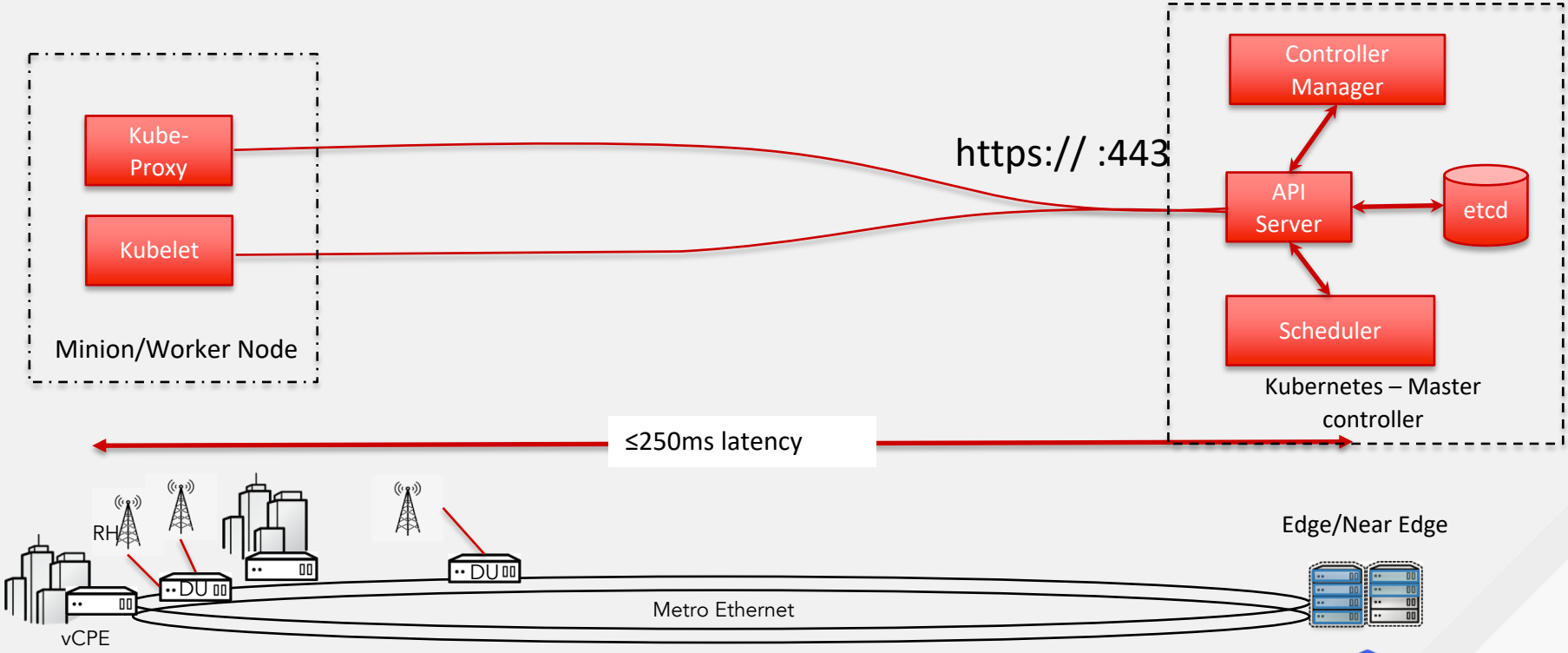
DISTRIBUTED COMPUTE

Stretched K8S Cluster - No Infrastructure control plane at the Far Edge Site



DISTRIBUTED COMPUTE

Stretched K8S Cluster - No Infrastructure control plane at Edge Site



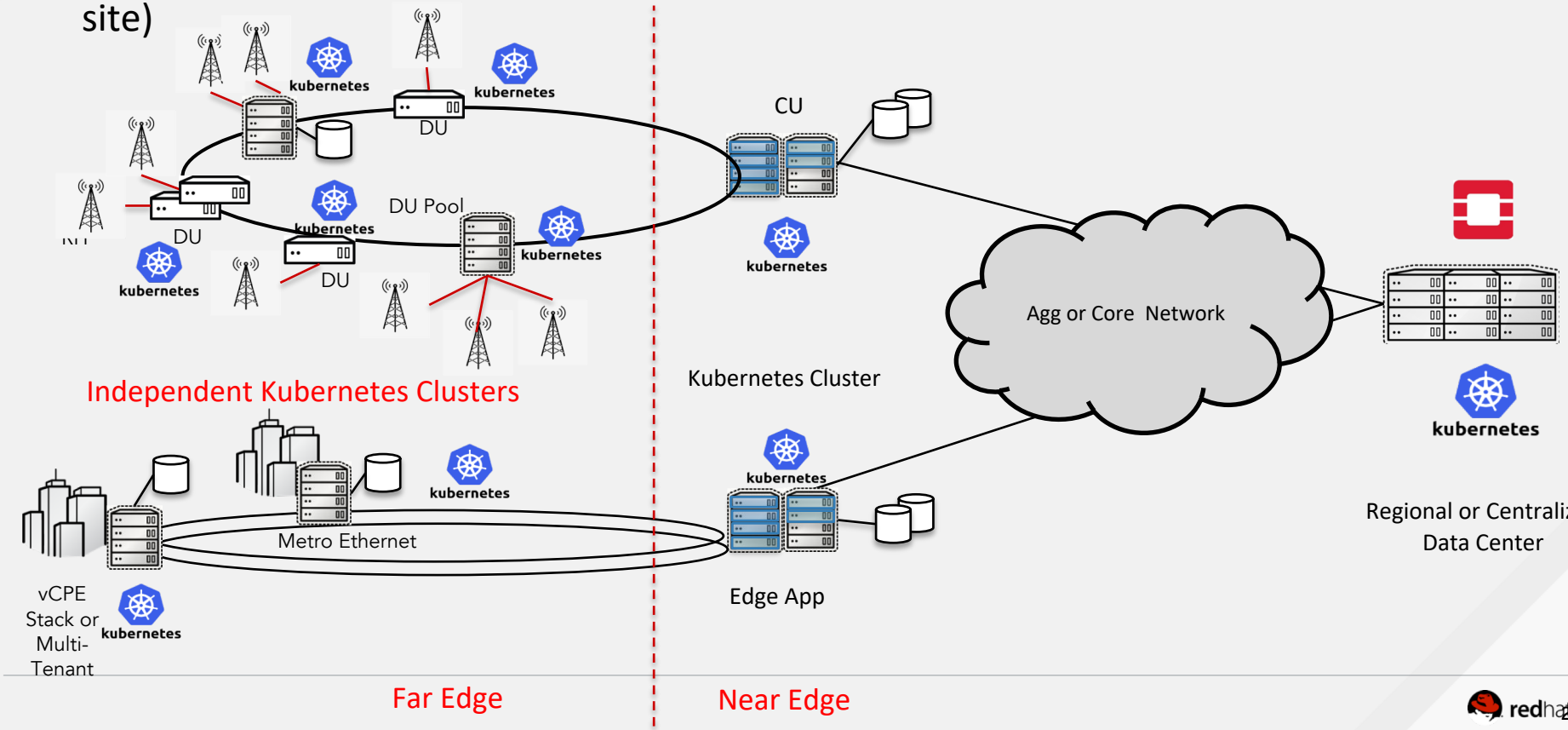
Distributed K8S worker Nodes

Enterprise and SMB

kubernetes

MULTI-CLUSTER ENVIRONMENT

Independent K8S clusters (1 node to n nodes in each Far Edge site)



Multi-Cluster Federation

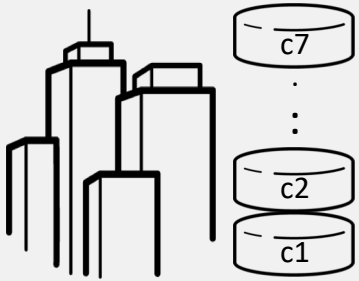
Multiple Independent Clusters

- Each Cluster is independent of each other
- Each Cluster can be installed from a central site
  - With the site and topo descriptions (yaml declarative topology) - Push model
  - From the site - Pull model
- Each Cluster manages local resources including storage
- Cookie cutter approach to site install and management
- Federation across clusters
  - From any cluster in the federation, workloads can be scheduled to any other member cluster including replica scheduling
  - Workloads can be moved from one member in the federation to another member in the federation
- Federation of API any type including CRDs

Multi-Cluster Federation

- Multi-Cluster Networking
  - Tunneling Scenarios – creating secure tunnels between clusters
  - Routing – Managing routing rules to get traffic from one site to another
  - Service Discovery – Discovering services across all the clusters
  - Network Policy – Cross Cluster Network Policy
- Placement decisions
- Dynamic Scheduling

MULTI-CLUSTER – HYBRID CLOUD



OpenShift Clusters c1 through c7

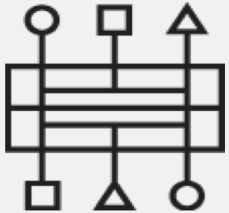
$ openshift-install create cluster

Cluster Registry CRD

Single Source of Truth

$ oc get clusters

Federated API

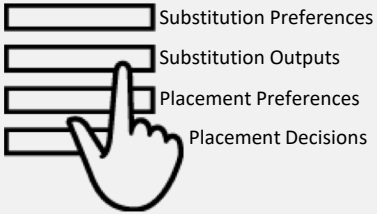


Base Federated Resources

FederatedDeployment
FederatedSecret
FederatedReplicaSet
FederatedConfigMap

Bonus: *Federate any CRD without writing code*

Schedule and Reconcile



Auxiliary Resources

```
overrides:
  clusters:
  - clusterName: c1
    replicas: 5
  - clusterName: c3
    replicas: 10
  - clusterName: c7
    replicas: 15
```

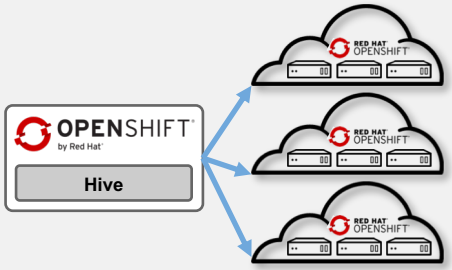
https://github.com/kubernetes/community/tree/master/sig-multicluster

redhat.

Example: OpenShift Hive

API Driven Multi-cluster Provisioning & Lifecycle Management

- Reliably provision/deprovision, upgrade, & configure clusters
  - Public Cloud Support
  - Private Cloud
  - Cluster creation for CI
- Leverages:
  - ***openshift-install*** - Uses CLI to launch clusters in the public cloud
  - **Kubernetes Cluster API** - Declarative, Kubernetes-style APIs for cluster creation, configuration, and management
  - **Kubernetes Federation** - Makes it easy to manage multiple clusters
- Working code & documentation now available:
  - *https://github.com/openshift/hive*

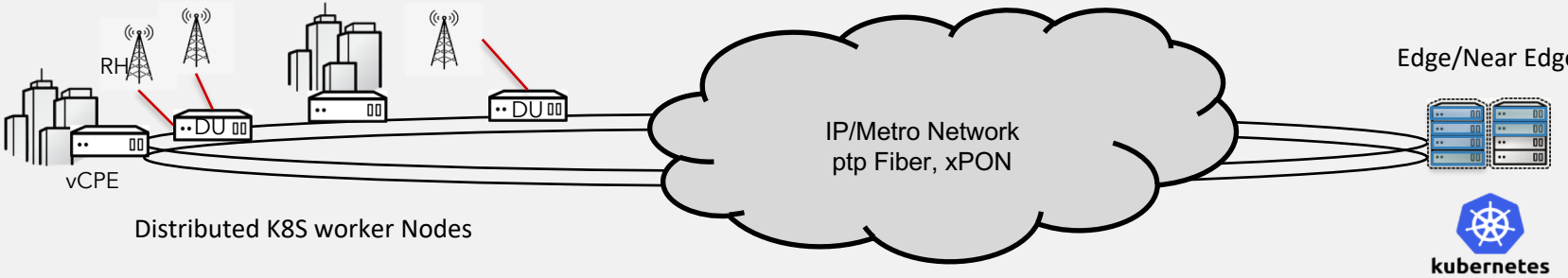



```
- apiVersion: hive.openshift.io/v1alpha1
  kind: ClusterDeployment
  metadata:
    labels:
      controller-tools.k8s.io: "1.0"
    annotations:
      hive.openshift.io/delete-after: "8h"
      hive.openshift.io/try-install-once: "${TRY_INSTALL_ONCE}"
    name: ${CLUSTER_NAME}
  spec:
    platformSecrets:
      aws:
        credentials:
          name: "${CLUSTER_NAME}-aws-creds"
    images:
      hiveImage: "${HIVE_IMAGE}"
      hiveImagePullPolicy: "${HIVE_IMAGE_PULL_POLICY}"
      installerImage: "${INSTALLER_IMAGE}"
      installerImagePullPolicy: "${INSTALLER_IMAGE_PULL_POLICY}"
      releaseImage: "${OPENSHIFT_RELEASE_IMAGE}"
    sshKey:
      name: "${CLUSTER_NAME}-ssh-key"
    clusterName: ${CLUSTER_NAME}
    baseDomain: ${BASE_DOMAIN}
    networking:
      type: OpenshiftSDN
      serviceCIDR: "172.30.0.0/16"
      machineCIDR: "10.0.0.0/16"
      clusterNetworks:
        - cidr: "10.128.0.0/14"
          hostSubnetLength: 9
    platform:
      aws:
        region: us-east-1
    pullSecret:
      name: "${CLUSTER_NAME}-pull-secret"
```

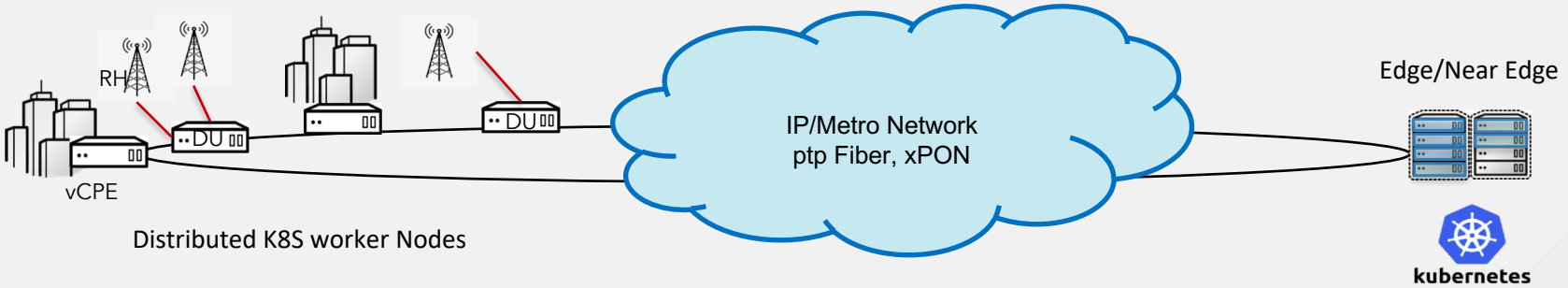
DISCOVERY
&
AUTO CONFIGURATION

Scenarios - provider owned vs. 3rd party Infra

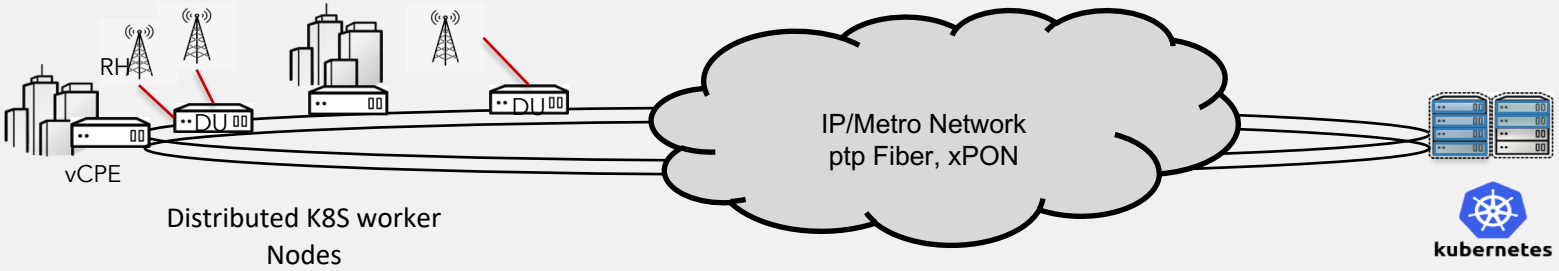
Scenario – A – Entire Network Infra is owned by the provider



Scenario – B– The Network belongs to another provider



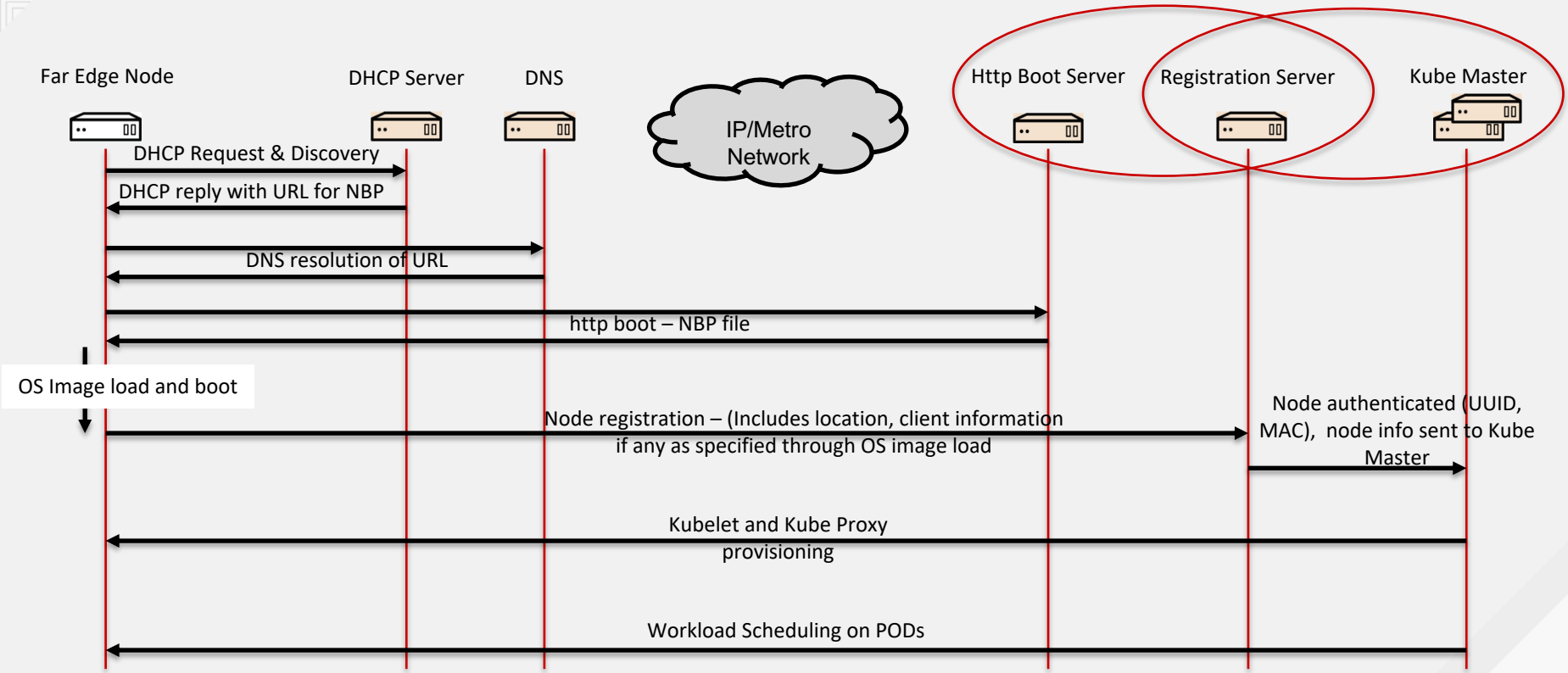
Scenario A



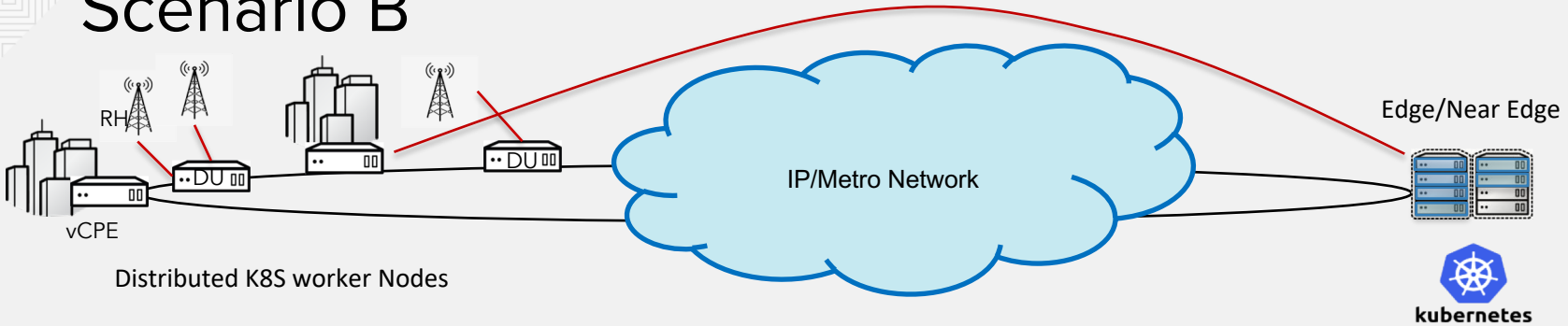
Scenario – A – Entire Network Infra is owned by the provider

- Provider controls the infrastructure
- Remote Node gets the IP address from the DHCP server of the provider – Provider controls the address space
- DHCP option allows passing of boot URL for UEFI URL boot process – http boot procedure
- The URL specifies the boot server with NBP information
- Once the device loads the OS (Deployment kernel+Ramdisk)
    - Pulls the disk image
    - Initial config
    - Registered to the registration server which then add it to kube cluster
- Kubelet and Kube proxy are then installed and Pod is prepared - Included in RH Container OS
- Once the Pod is active – K8S schedules workloads

SCENARIO A



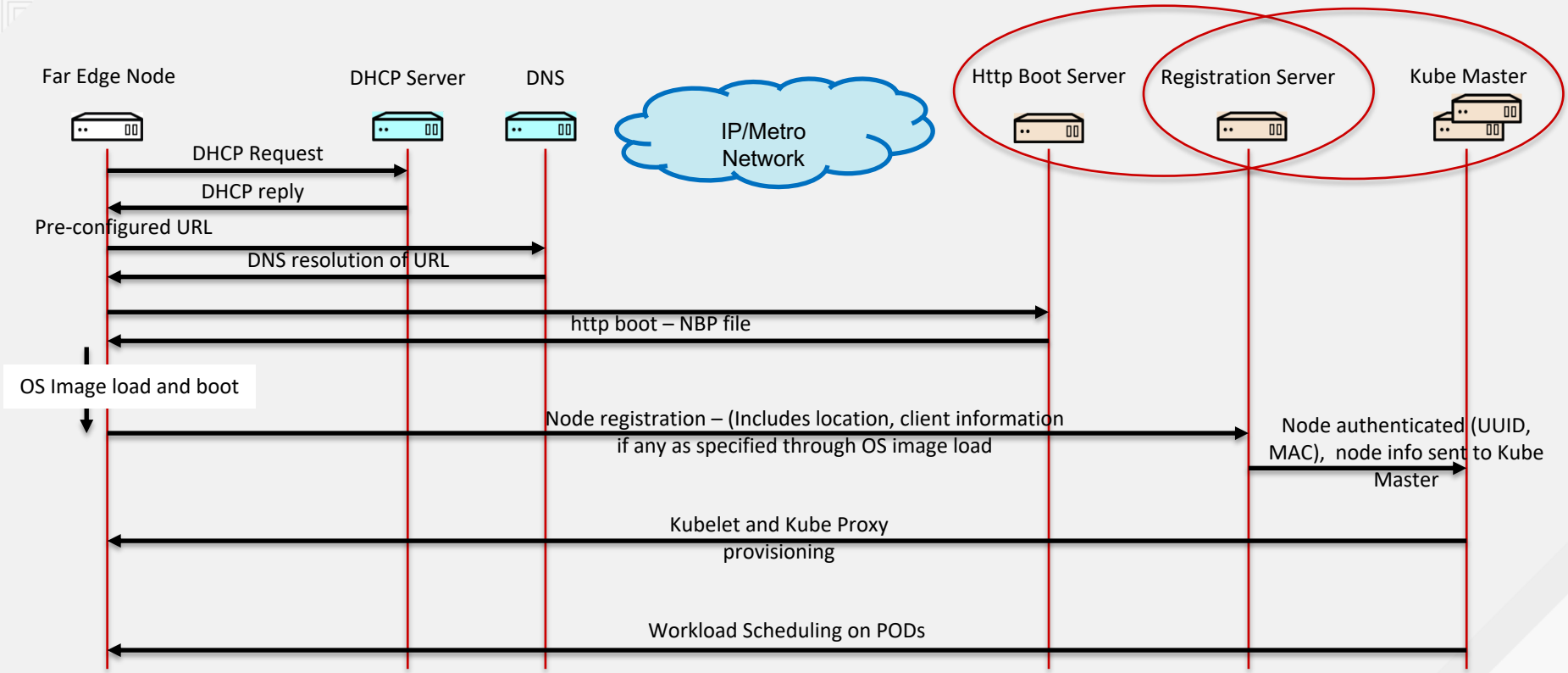
Scenario B



Scenario – B– The Network belongs to another provider

- Provider only controls the Ultra/Far Edge Node(s) and does not control infrastructure/IP Network in between
- Remote Node gets the IP address from the DHCP server of the third party provider –
- Remote Node is pre-configured with http boot URL or DHCP third party option
- The URL specifies the boot server with NBP information
- Once the device loads the OS and the profiles the node is then registered to the registration server which then add it to kube cluster
- Kubelet and Kube proxy are then installed and Pod is prepared
- Once the Pod is active – K8S schedules workloads

SCENARIO B



Multi-Node Far Edge

When Far Edge sites have multiple nodes connected via switch(es)

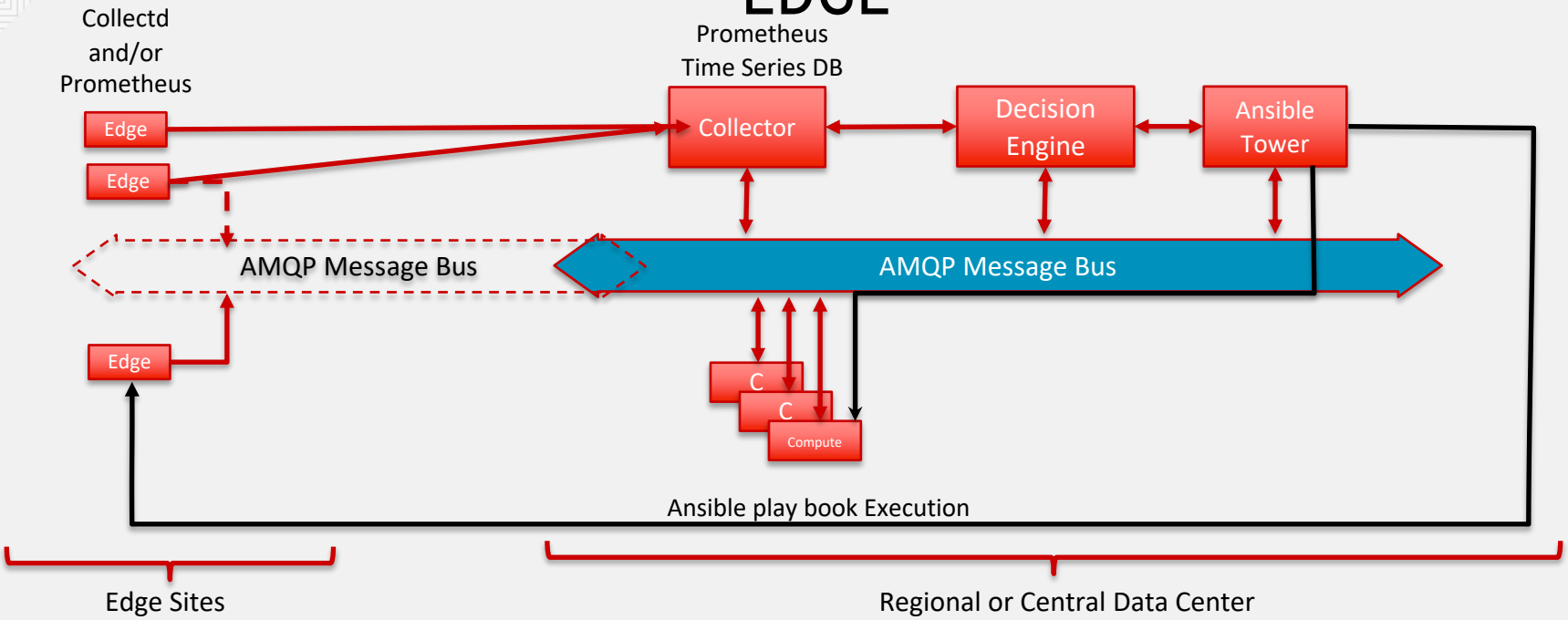
Scenario A

- Topology information of the site (via discovery)
- Config Site/Server Role
  - Read Topology information from topology database
  - Discover the nodes
  - Update profiles and OS – push certificates etc.
  - Register nodes to the Kube Master/Cluster
- Nodes provisioned and ready for workloads

Scenario B

- Only one node in the site must be configured with http boot URL and follows the process outlined in the previous slide
- Once that node is authenticated then a yaml file that describes the site topology is fetched
- All other nodes in the site are now provisioned through the same process.

EXTENDING ASSURANCE MODEL TO ULTRA/FAR EDGE

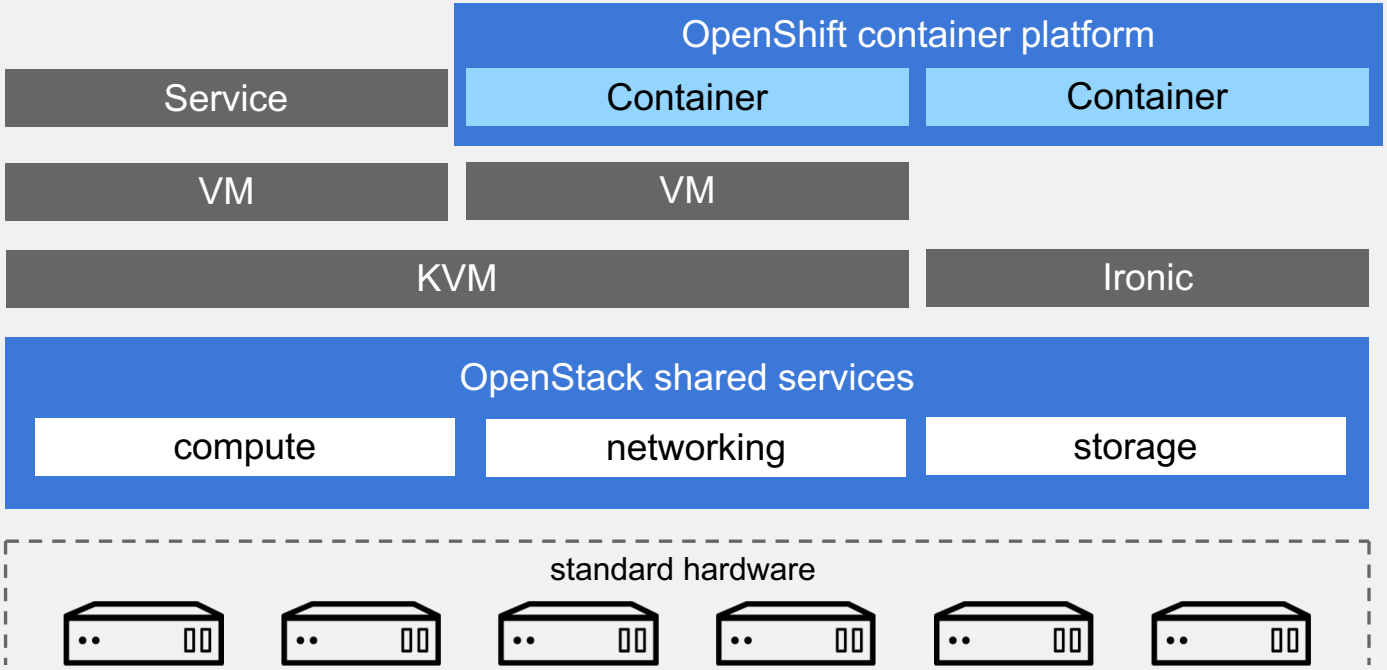


- Automating placement of workloads that meet the criteria
- Monitoring Change
- Proactive and reactive action

- Rules based closed feedback loop
- Requires all layers of stack to communicate
- Correlation

Hybrid workloads for Edge – VNFs and Applications

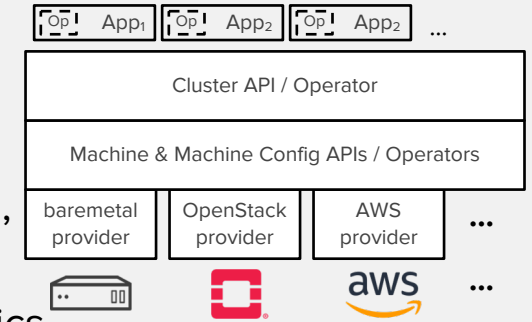
Containers, Virtual Machines, and Bare-metal “pick and mix”



Kubernetes-Native Infra for Edge (KNI-Edge) Family

The KNI-Edge Family unites edge computing blueprints sharing the following characteristics:

- Implement the Kubernetes community's Cluster API
  - declaratively configure and consistently deploy and lifecycle manage Kubernetes clusters on-prem or public cloud, on VMs or bare metal, at the edge or at the core.
- Leverage the community's Operator Framework for app LCM
  - applications lifecycle managed as Kubernetes resources, in event-driven manner, and fully RBAC-controlled
  - more than deployment + upgrades, e.g. metering, analytics
  - created from Helm Charts, using Ansible or Go
- Optimize for Kubernetes-native container workloads
  - but allow mixing in VM-based workloads via KubeVirt as needed.



KNI-Edge Family - Proposal Template

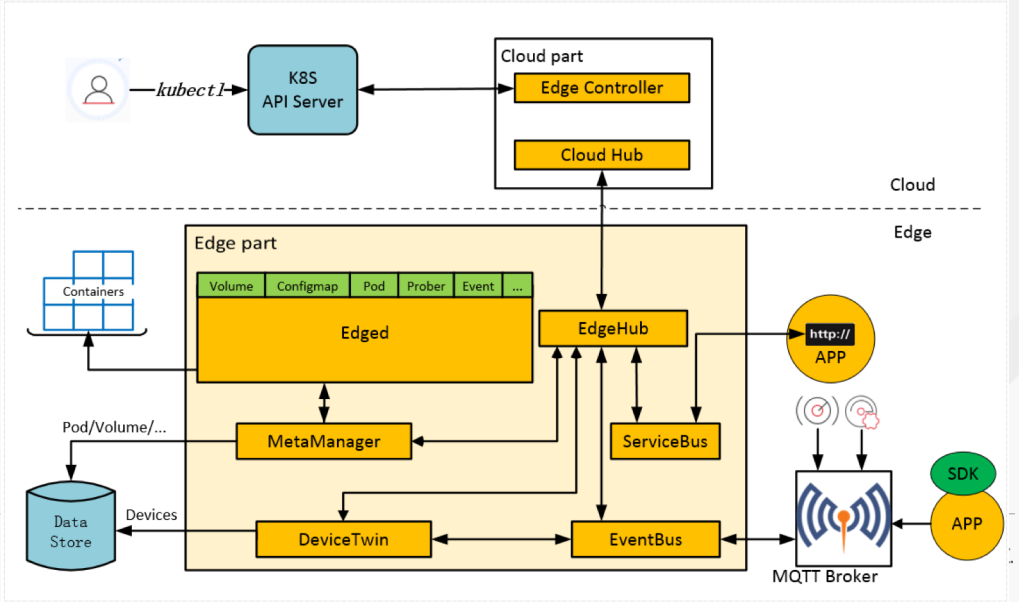
Case Attributes	Description
Type	New
Blueprint Family - Proposed Name	Kubernetes-Native Infrastructure for Edge (KNI-Edge)
Use Case	various, e.g.: ● Provider Access Edge (Far/Near), MEC ● Industrial Automation ● Enterprise Edge ● ...
Blueprint proposed	various; initially: ● Provider Access Edge (PAE) ● Industrial Edge (IE)
Initial POD Cost (capex)	(depends on blueprint)
Scale	1 to hundreds of nodes, 1 to thousands of sites.
Applications	any type of workloads: ● containerized or VM-based ● real-time, ultra-low latency or high-throughput ● NFV, IoT, AI/ML, Serverless, ...
Power Restrictions	(depends on blueprint)
Preferred Infrastructure orchestration	End-to-end Service Orchestration: depends on use case; e.g. ONAP App Lifecycle Management: Kubernetes Operators Cluster Lifecycle Management: Kubernetes Cluster API/Controller Container Platform: Kubernetes (OKD) Container Runtime: CRI-O w/compatible backends VM Runtime: KubeVirt
	OS: RHEL, RHEL-RT, CentOS, CentOS-rt, or CoreOS
Additional Details	

Topics not covered

- Networking
  - Tunneling
  - Routing
- HA models and configurations
  - Headless operation of the Ultra/Far Edge
  - Recovery of Ultra/Far Edge
- Orchestration & Policy
- Security
- Storage permutations and combinations
  - Volume mapping
  - Persistence

Other solutions

- Third Party solutions for Kube cluster management for multisite and distributed deployment
- KubeEdge Project accepted into CNCF
  - Targeted towards edge devices - U0 layer
  - MQTT for IoT
  - New release v0.2 out today



SUMMARY

- Edge implies massive distribution of sites – high scale and volume
- Has specialized requirements wrt real time requirements & Hybrid Container/VM Models
- Kubernetes Native blueprint for Far Edge and Ultra Far Edge possible
- Separating Infra functions from application deployment allows usage of Kubernetes for Edge
- Work in the community to enhance multi-site and multi-cluster configurations
- Networking options to be further solidified
- Call to Action : Join us in building an “Autonomous Intelligent Edge”

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos