Open Networking Foundation

# CORD: Multi-Access Edge Cloud

*– Building an Extensible Edge Platform –*

Larry Peterson

# Background

CORD is a multi-access edge cloud

- Built using commodity servers and white-box switches/access devices (PON, RAN)
- Runs both scalable cloud services and disaggregated Telco services (BNG, EPC)
- Configured as *Base Platform* + One or more *Service Profiles*

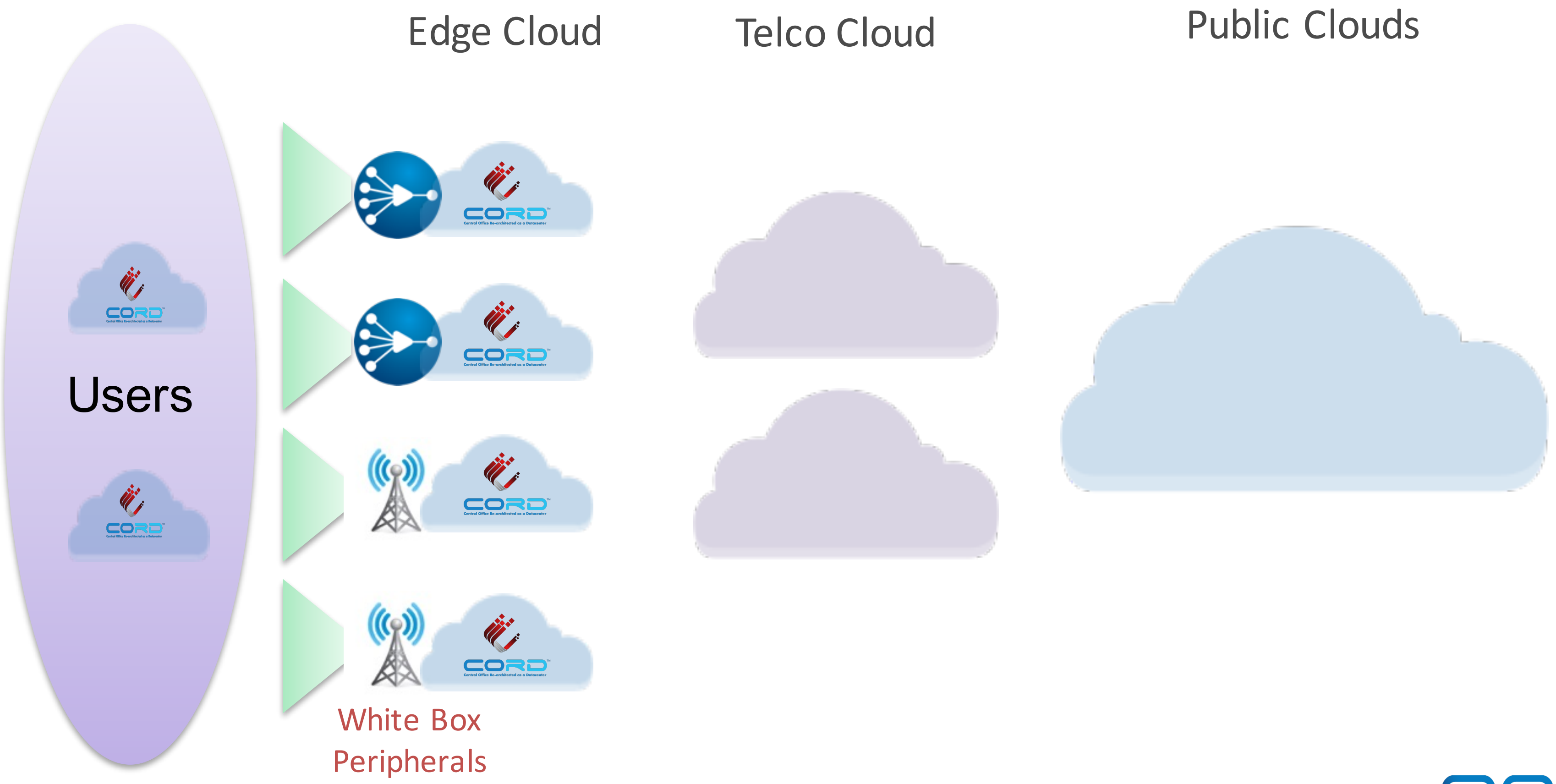
XOS is a framework for configuring and operating a cloud platform

- Decouples *Service Control Plane* and *Service Data Plane*
- Generates the control plane from a set of *declarative models*

CORD and XOS are open source projects of the ONF

- Working with network operators to take these technologies to production

# CORD – Reinventing the Network Edge



Edge Cloud

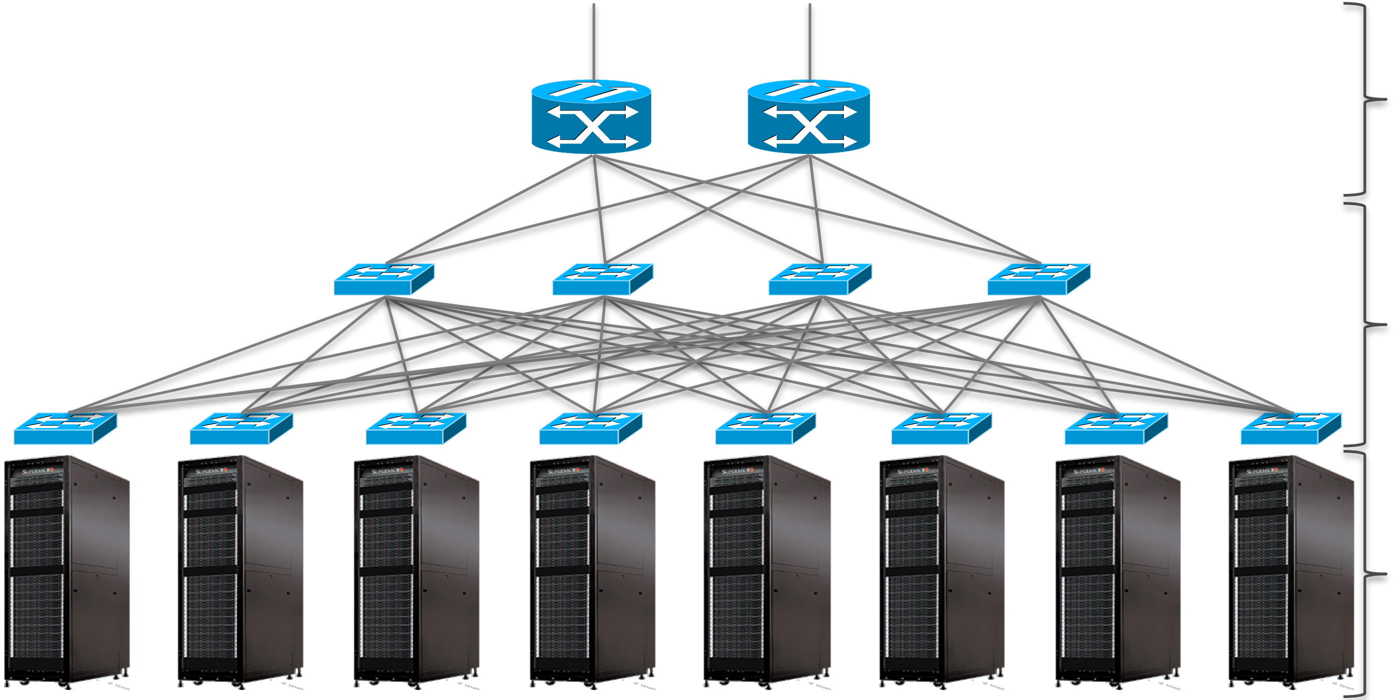
Telco Cloud

Public Clouds

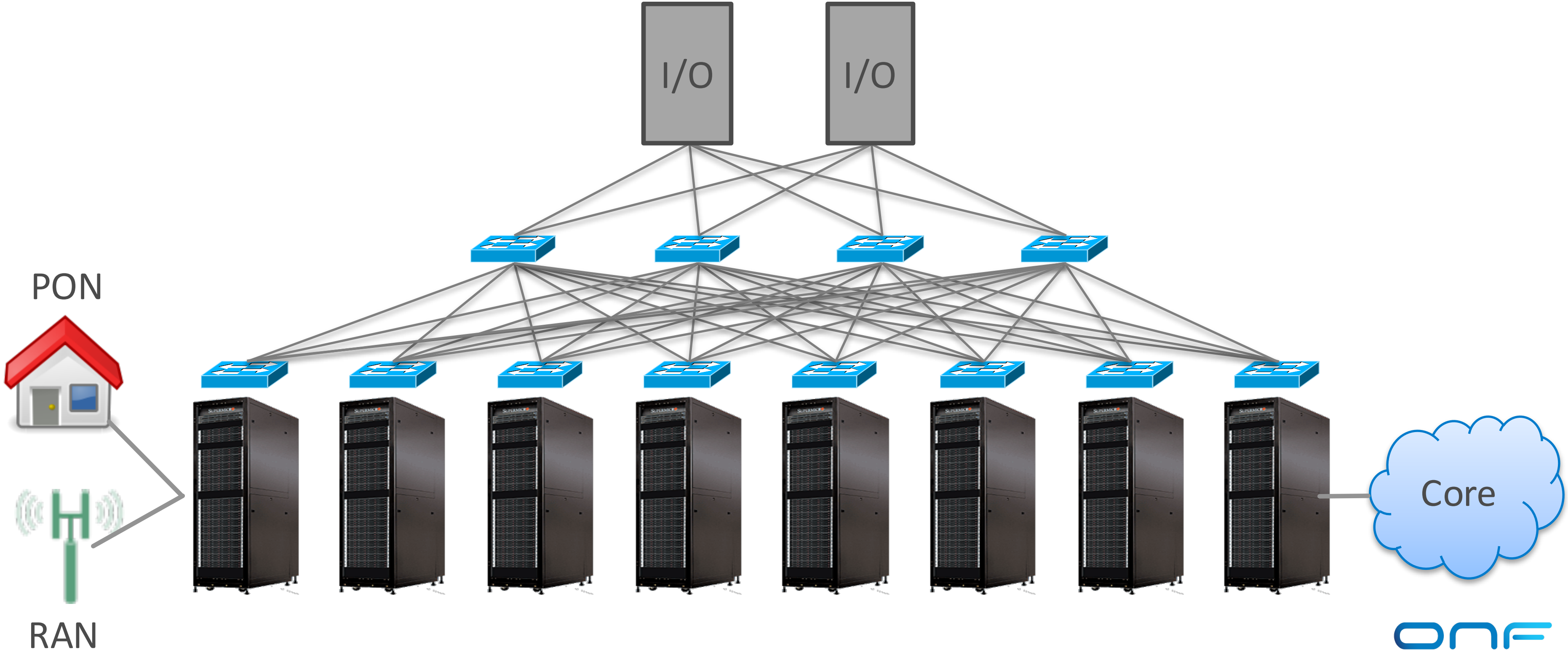
Users

White Box Peripherals

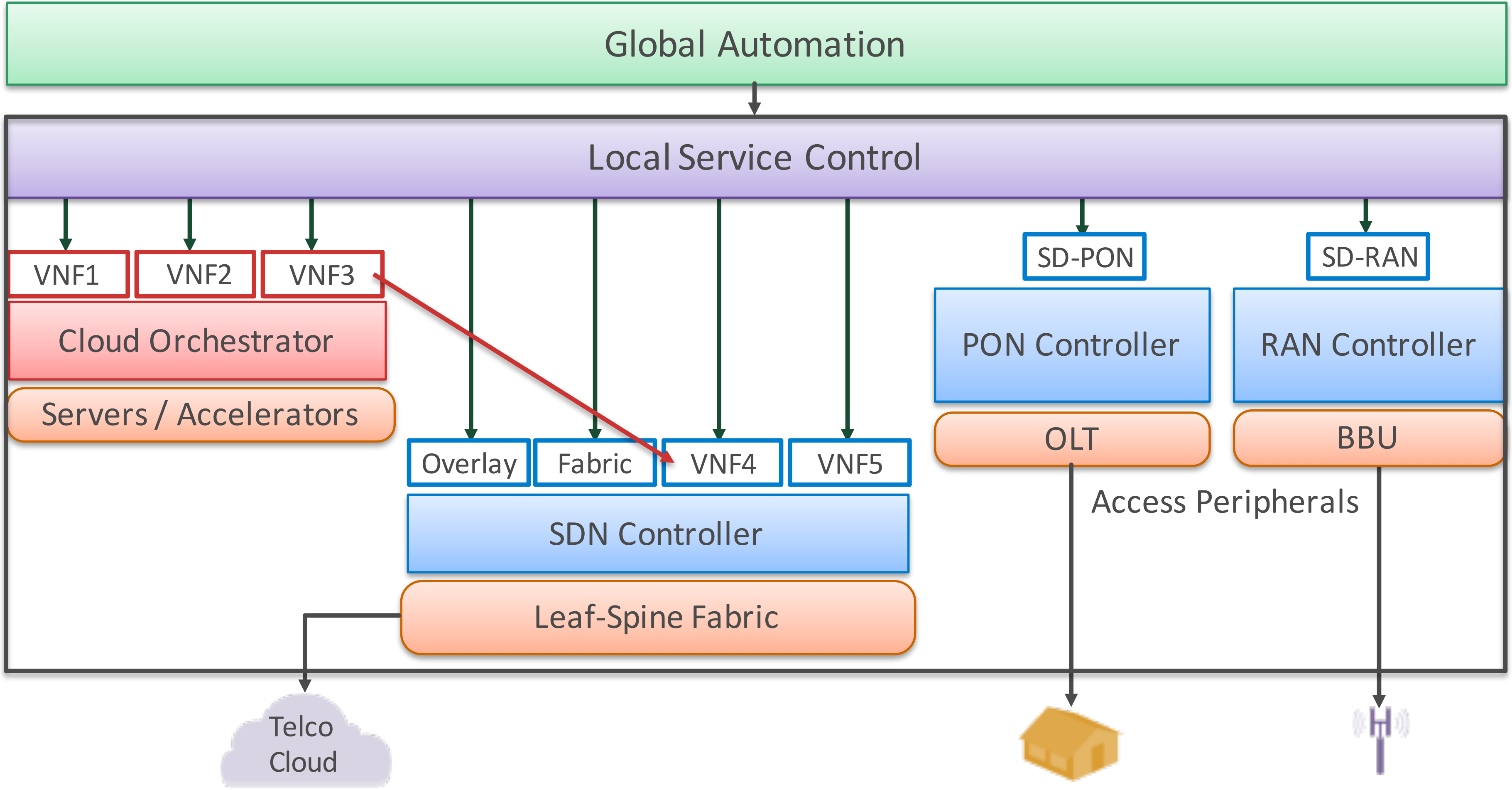
# Data Center



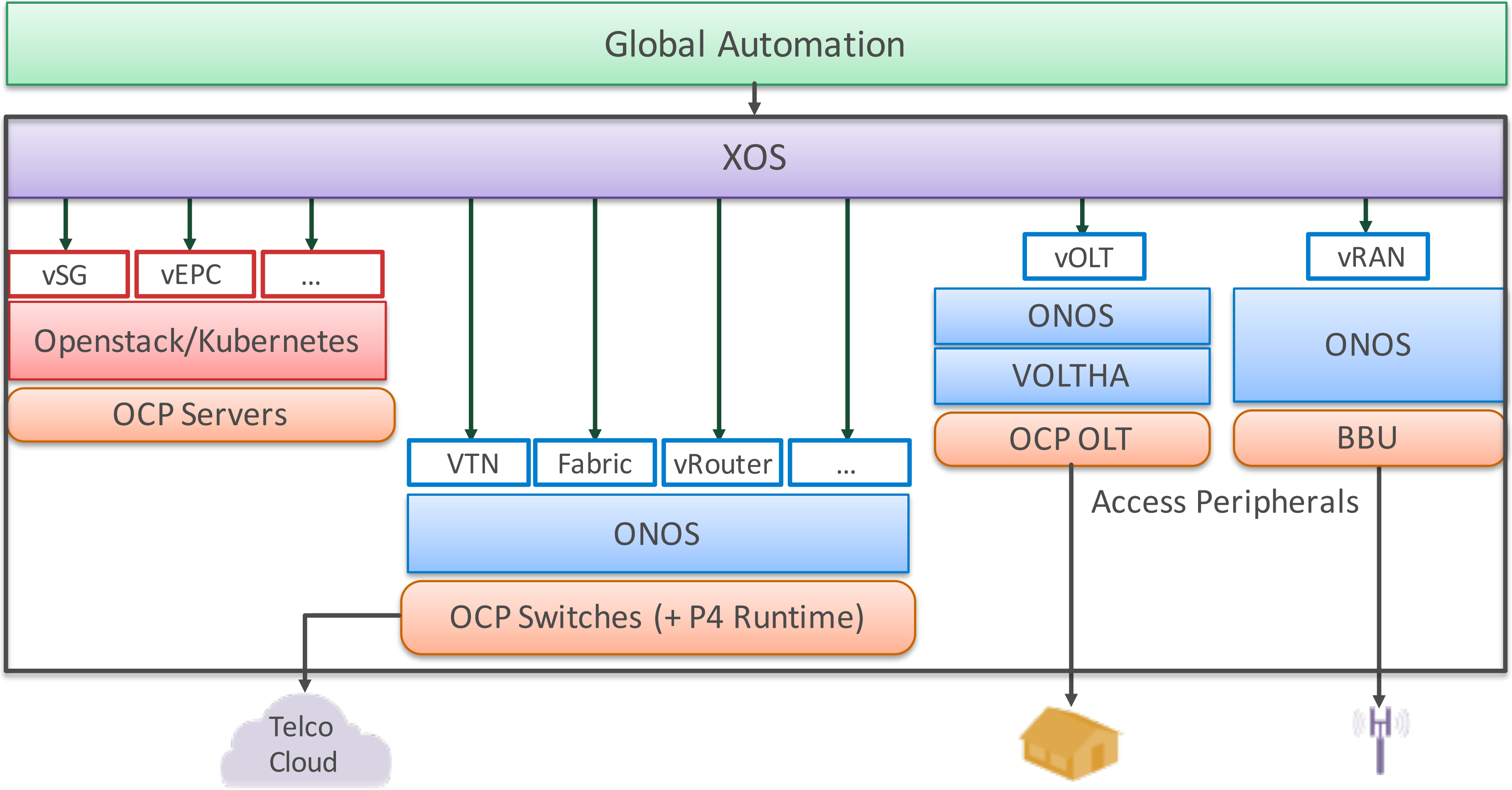
# Multi-Access Edge



# Reference Design – Multi-Access Edge Cloud



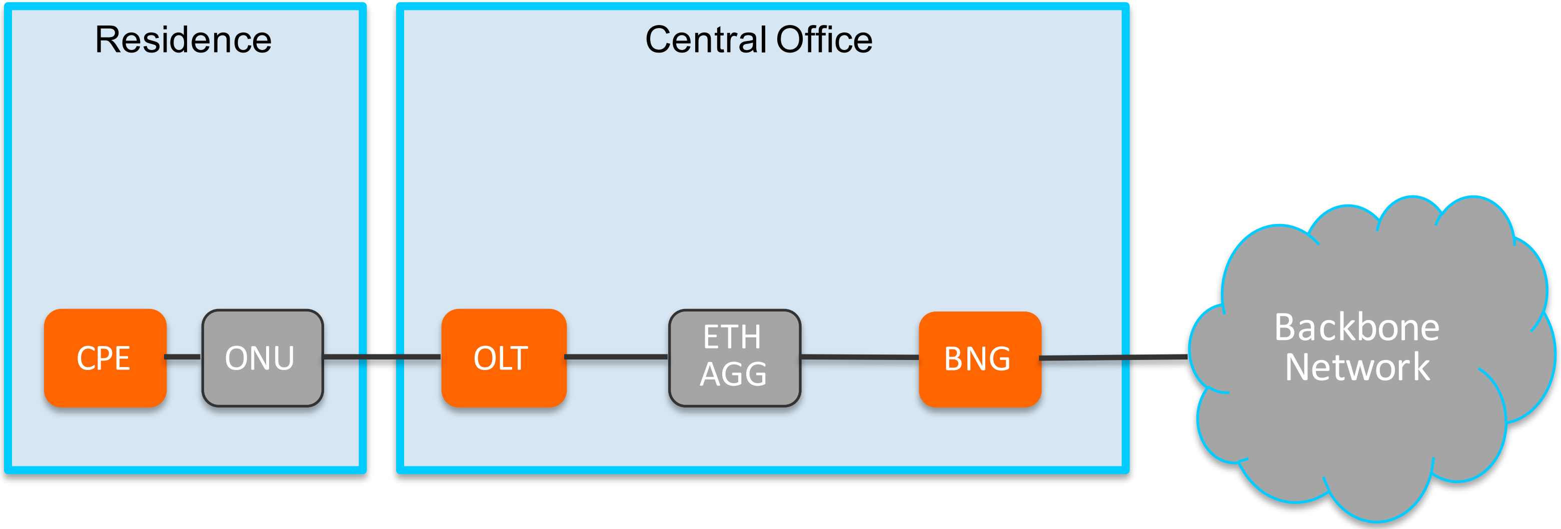
# Exemplar Platform – CORD



# Disaggregation

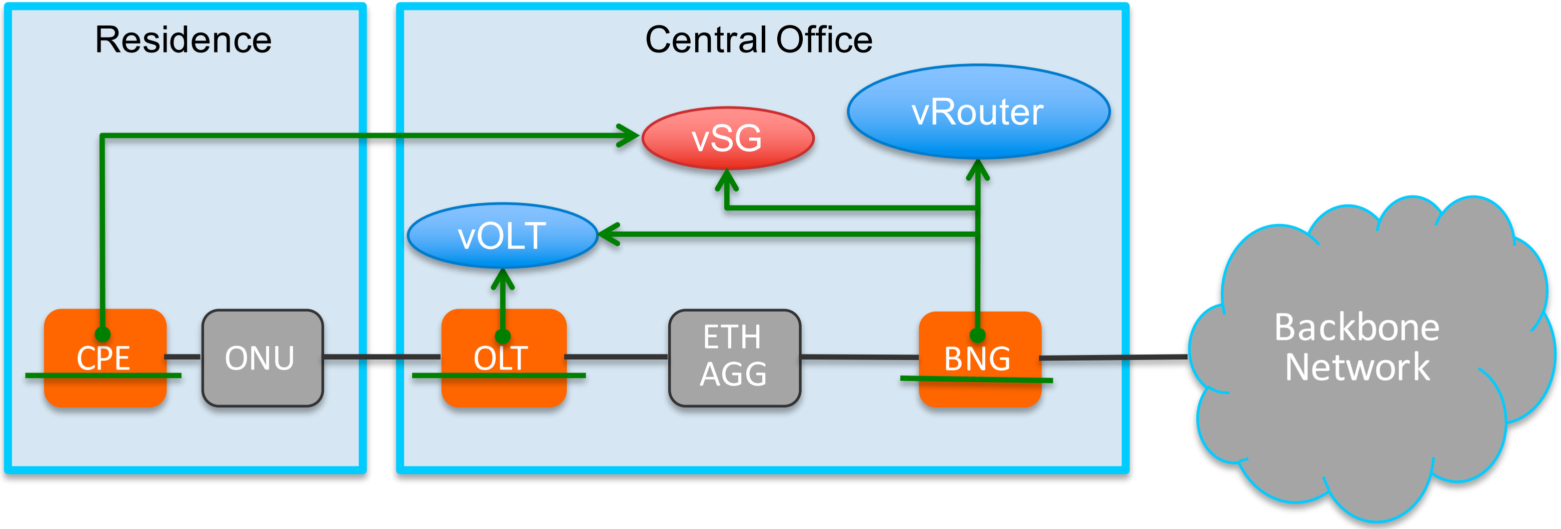
*– Micro-Services & SDN Applied to the Central Office –*

# Legacy Central Office



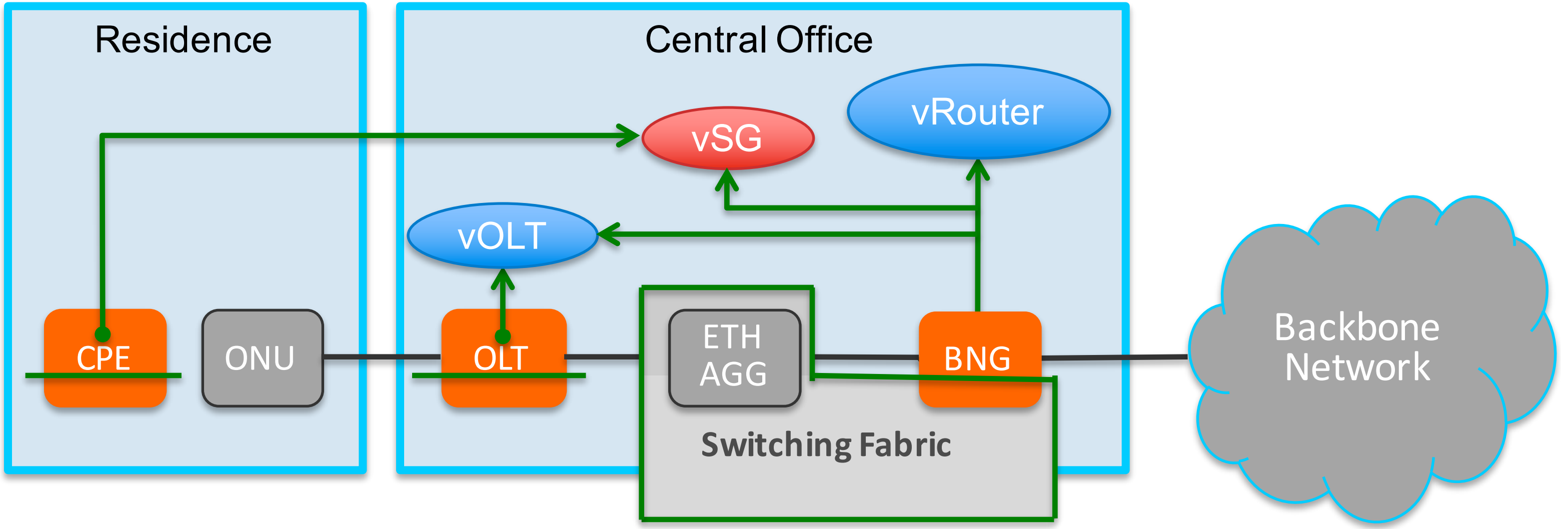
CPE – Customer Premises Equipment
OLT – Optical Line Termination
BNG – Broadband Network Gateway

# Disaggregation



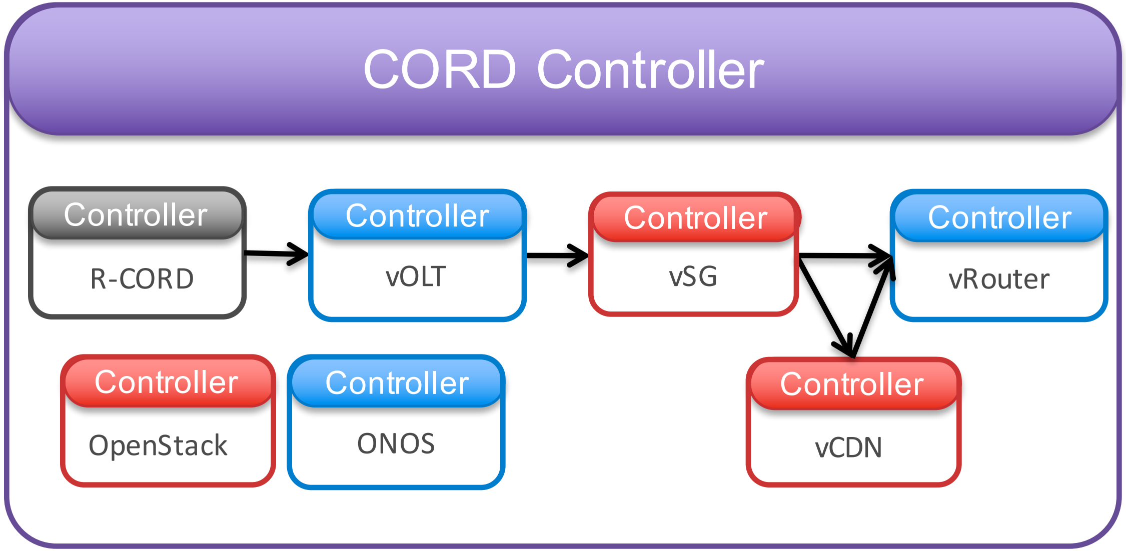
CPE – Customer Premises Equipment
OLT – Optical Line Termination
BNG – Broadband Network Gateway

# Disaggregation



CPE – Customer Premises Equipment
OLT – Optical Line Termination
BNG – Broadband Network Gateway

# Functional Specification



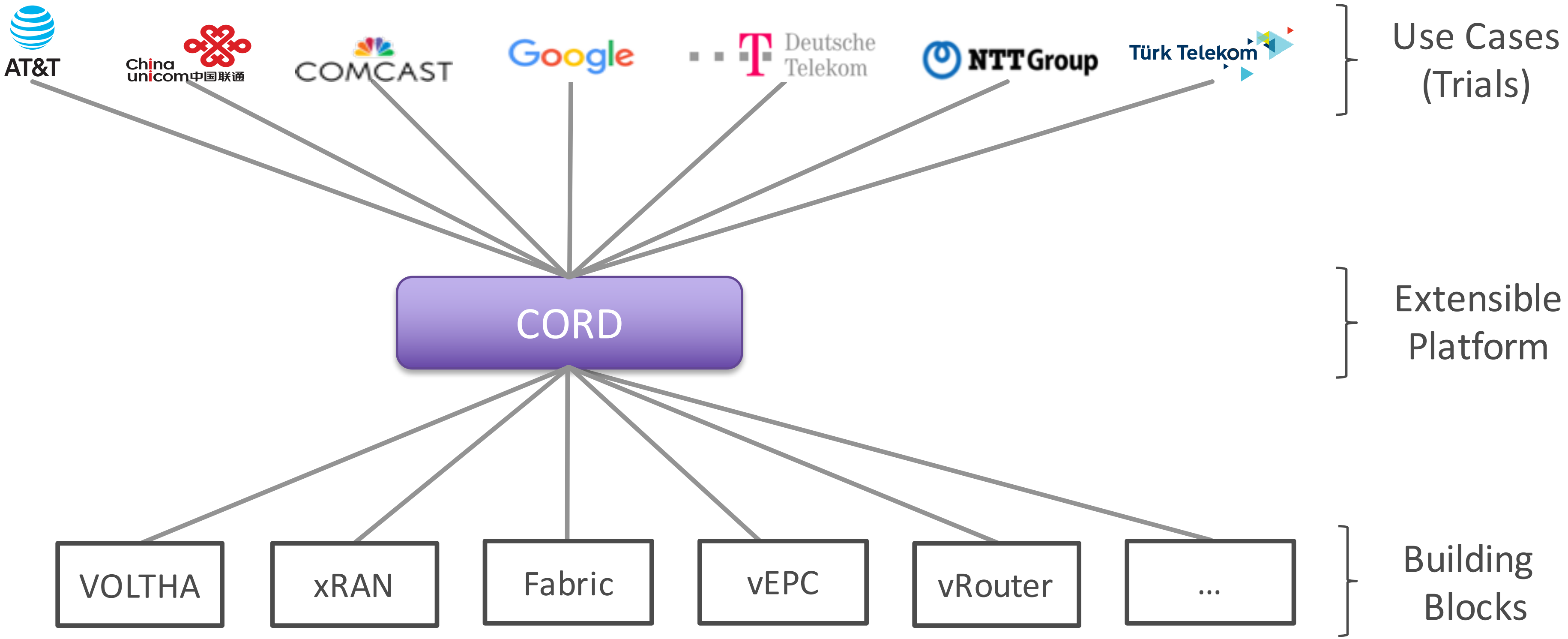
Service Graph for Residential CORD

Open Networking Foundation

# CORD – An Extensible Platform

*– Configuring and Controlling an Integrated System –*

# Challenge



CORD

VOLTHA

xRAN

Fabric

vEPC

vRouter

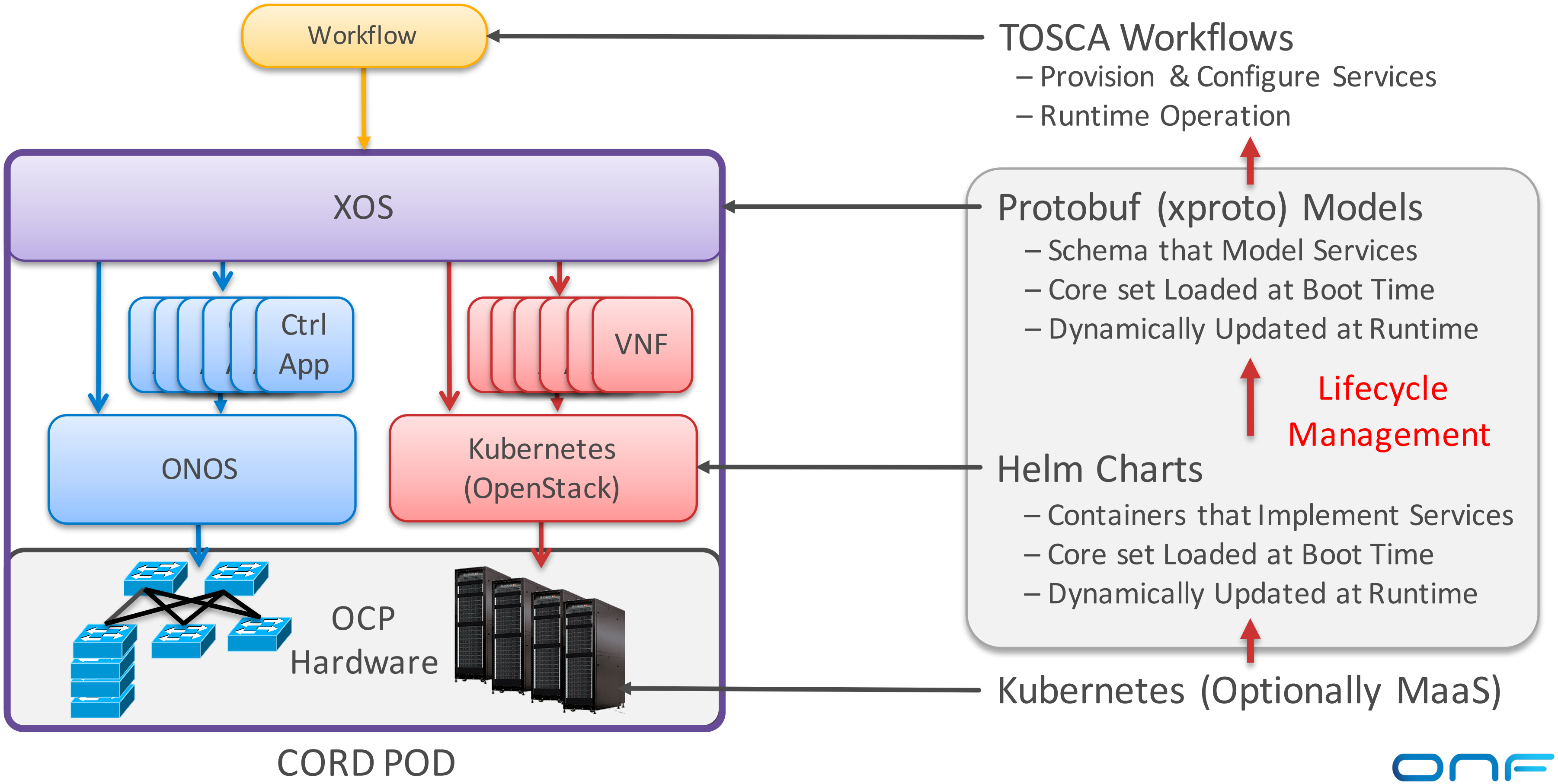
...

Use Cases (Trials)

Extensible Platform

Building Blocks

# Automated Configuration



TOSCA Workflows

- Provision & Configure Services
- Runtime Operation

Protobuf (xproto) Models

- Schema that Model Services
- Core set Loaded at Boot Time
- Dynamically Updated at Runtime

Lifecycle Management

Helm Charts

- Containers that Implement Services
- Core set Loaded at Boot Time
- Dynamically Updated at Runtime

Kubernetes (Optionally MaaS)

# Lifecycle Management

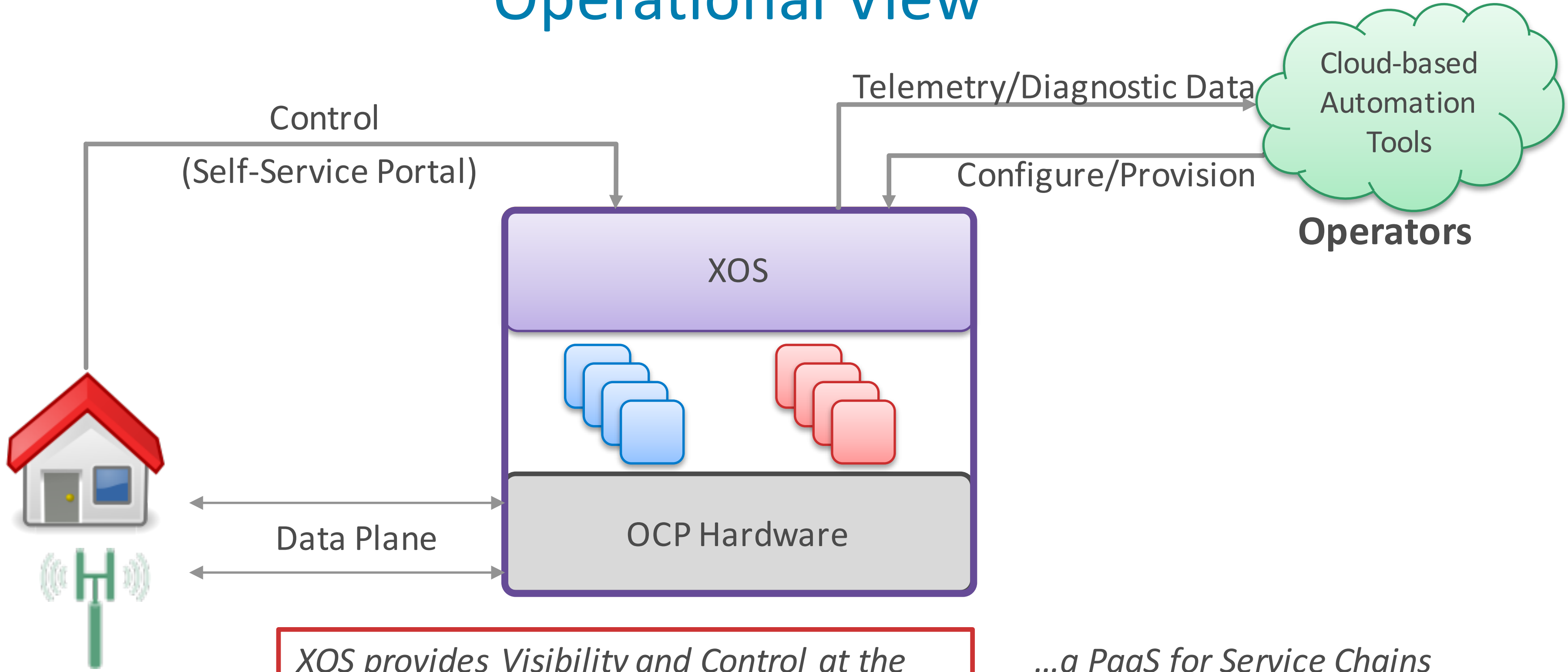
XOS is responsible for *Service Control Plane*

- Support for *configuring and controlling* services
- Support for incremental upgrades (transitioning state/interfaces)

Kubernetes is responsible for *Service Data Plane*

- Support for *implementing* services (scale up/down, HA)
- Support for incremental upgrades (rollout/rollback)

# Operational View



**Subscribers**

*XOS provides Visibility and Control at the granularity of per-subscriber service chains.*

*…a PaaS for Service Chains*

- *Provision*
- *Isolate*
- *Distribute*
- *Migrate*

Open Networking Foundation

# Mobile Cloud

*– Value of Service Chains –*

# What’s Different about 5G?

Earlier generations were about improving broadband technology

5G is fundamentally about supporting new services

- Internet-of-Things
- Immersive UIs
- Public Safety

What unique capability does the mobile access network offer?

- Low-latency proximity to end-users
- Intrinsic support for mobility

Challenge of 5G is to Simultaneously Support…

- Low Latency – Moving functionality to the edge, closer to devices
- Mobility – Accessing that edge functionality while continuing to be mobile

# Challenge

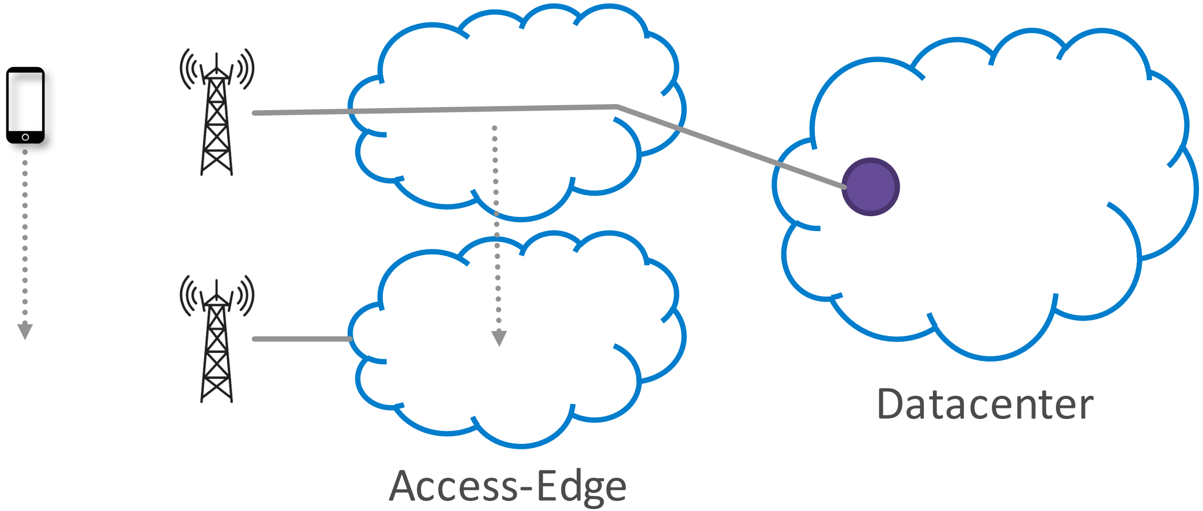
## Central Challenge of 5G is to Simultaneously Support…

- Low Latency – Moving functionality to the edge, closer to devices.
- Mobility – Accessing that edge functionality while continuing to be mobile.

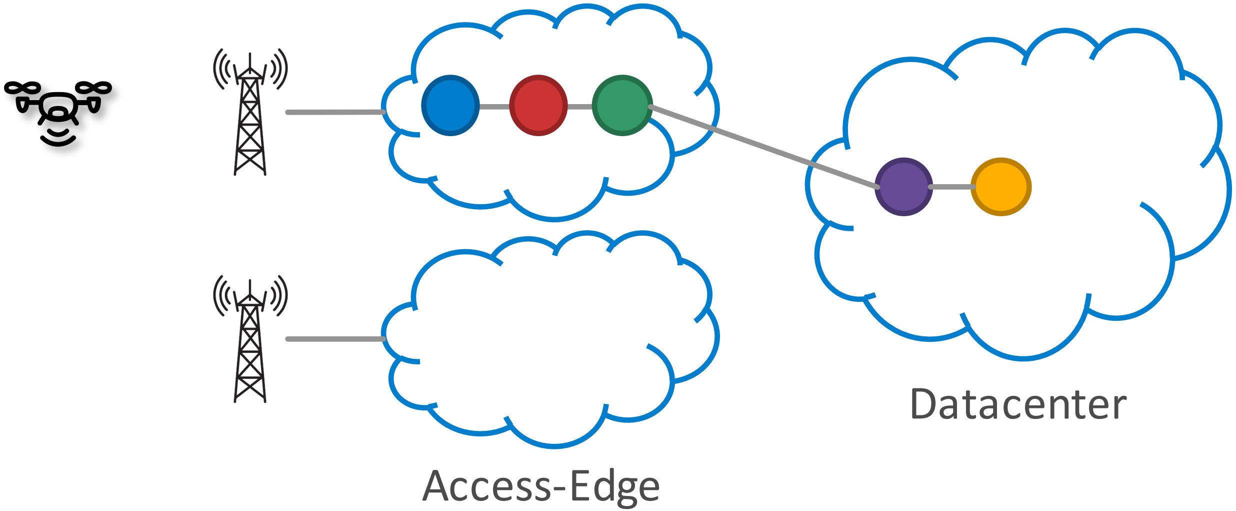
## Other Factors…

- Performance dictates that functionality be implemented in the most appropriate hardware (e.g., GPUs, Switching Fabric).
- Autonomy dictates that different stakeholders will be responsible for controlling and managing different components.
- Monetization dictates the need to offer differentiated services to different classes of subscribers/applications.
- Costs dictates a distributed solution, with some functions running in the datacenters and some running in a scalable number of edge sites.
- Dynamicity dictates the need for local (edge) control with tight control loops.

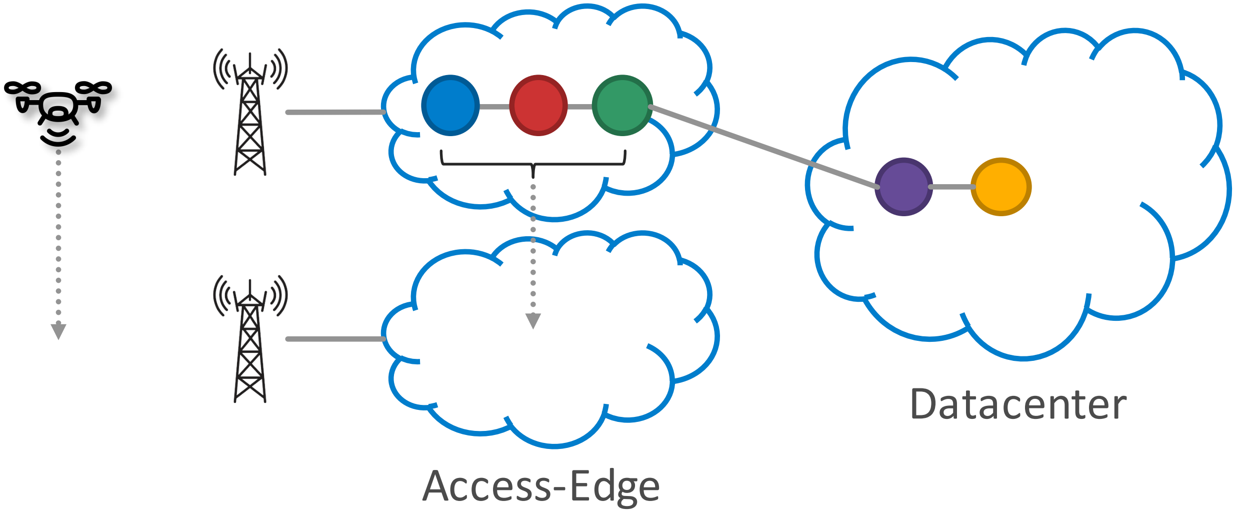
# Mobile Broadband (2G – 4G)



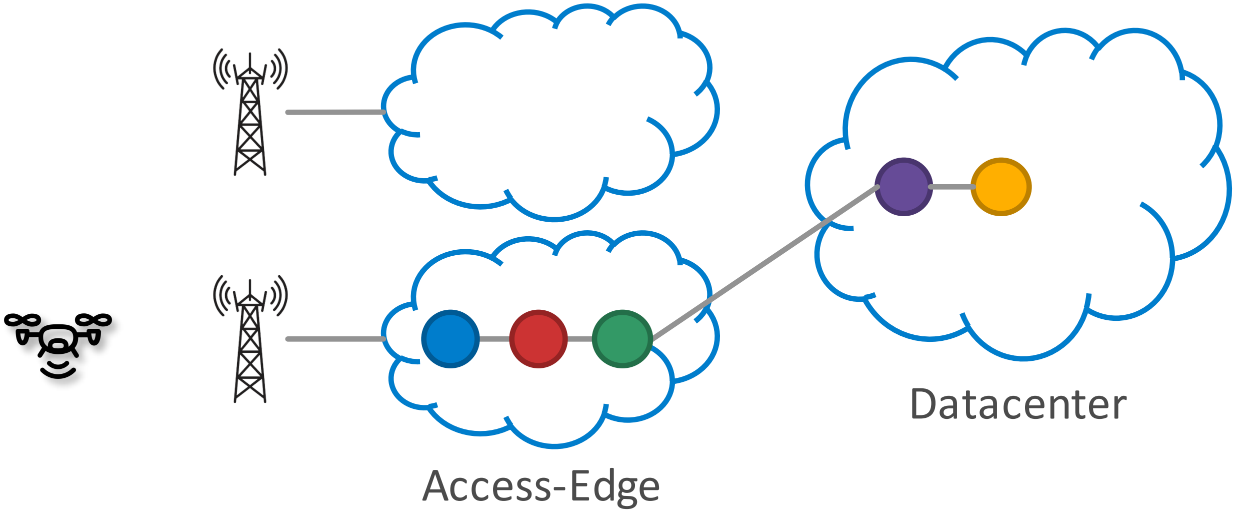
# Move Functionality to the Edge



# Mobile Cloud (5G)



# Mobile Cloud (5G)



# Requirements

Heterogeneous – Range of functional element implementations

Multi-Tenant – Multiple stakeholders managing functional elements

Distributed – Functional elements span multiple clouds

Isolation – Differentiated resource allocation between service chains

Mobility – Move service chains from one edge cloud to another

Open Networking Foundation

# XOS Overview

*– A PaaS for Service Chains –*

# CORD Innovations

Virtualization and Disaggregation

- Pre-requisite for moving functionality to the edge
- Ability to run functionality in both switches and servers

Explicit Support for Service Chains

- A first class abstraction that defines a control framework
- Operations to provision, distribute, isolate, and migrate

# What is XOS?

***xproto*** – A declarative language for specifying models

- *Protocol Buffers:* extended to support inheritance, relationships, and predicates

***xosgenx*** – An extensible toolchain to enforce models on an operational system

- *Targets:* APIs, Access Control, ORM, Synchronizer Framework,…

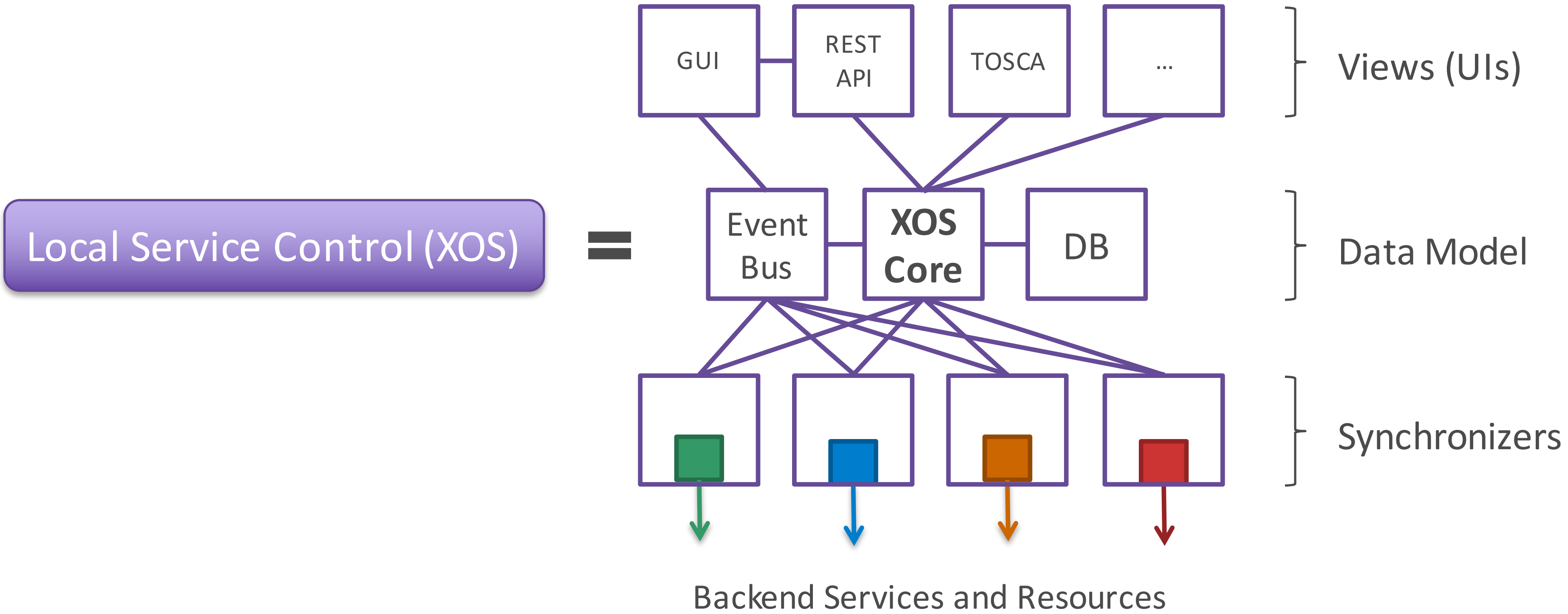
***core.xproto*** – A default (and malleable) set of core models

- *Models:* Service, ServiceDependency, ServiceInstance, ServiceInstanceLink,…

***Chart.yaml*** – A Helm Chart (plus set of container images) to deploy XOS

- *Micro-services:* xos-core, xos-gui, xos-tosca, xos-db, xos-ws, redis,…

# XOS Constructed from Micro-Services



# Example Model and Policy

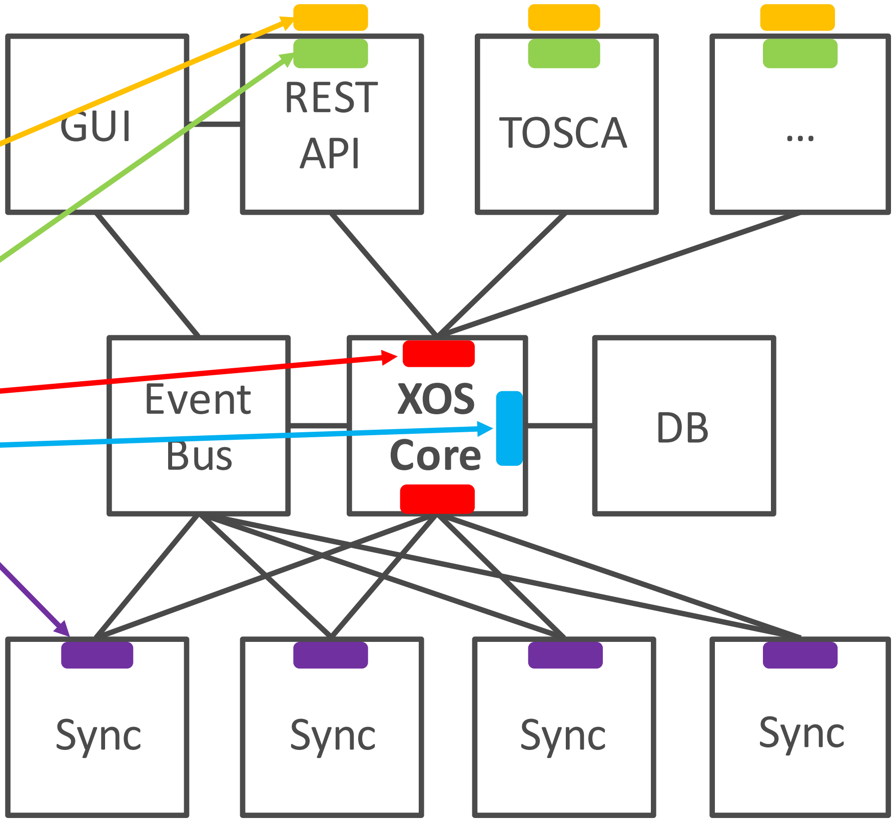
```
policy grant_policy < ctx.user.is_admin
      | exists Privilege:Privilege.object_type = obj.object_type
      & Privilege.object_id = obj.object_id
      & Privilege.accessor_type = "User"
      & Privilege.accessor_id = ctx.user.id
      & Privilege.permission = "role:admin" >

message Privilege::grant_policy (XOSBase)
      { required int32 accessor_id = 1 [null = False];
        required string accessor_type = 2 [null = False, max_length=1024];
        required int32 controller_id = 3 [null = True];
        required int32 object_id = 4 [null = False];
        required string object_type = 5 [null = False, max_length=1024];
        required string permission = 6 [null = False, default = "all", max_length=1024];
        required string granted = 7 [content_type = "date", auto_now_add = True, max_length=1024];
        required string expires = 8 [content_type = "date", null = True, max_length=1024]; }
```

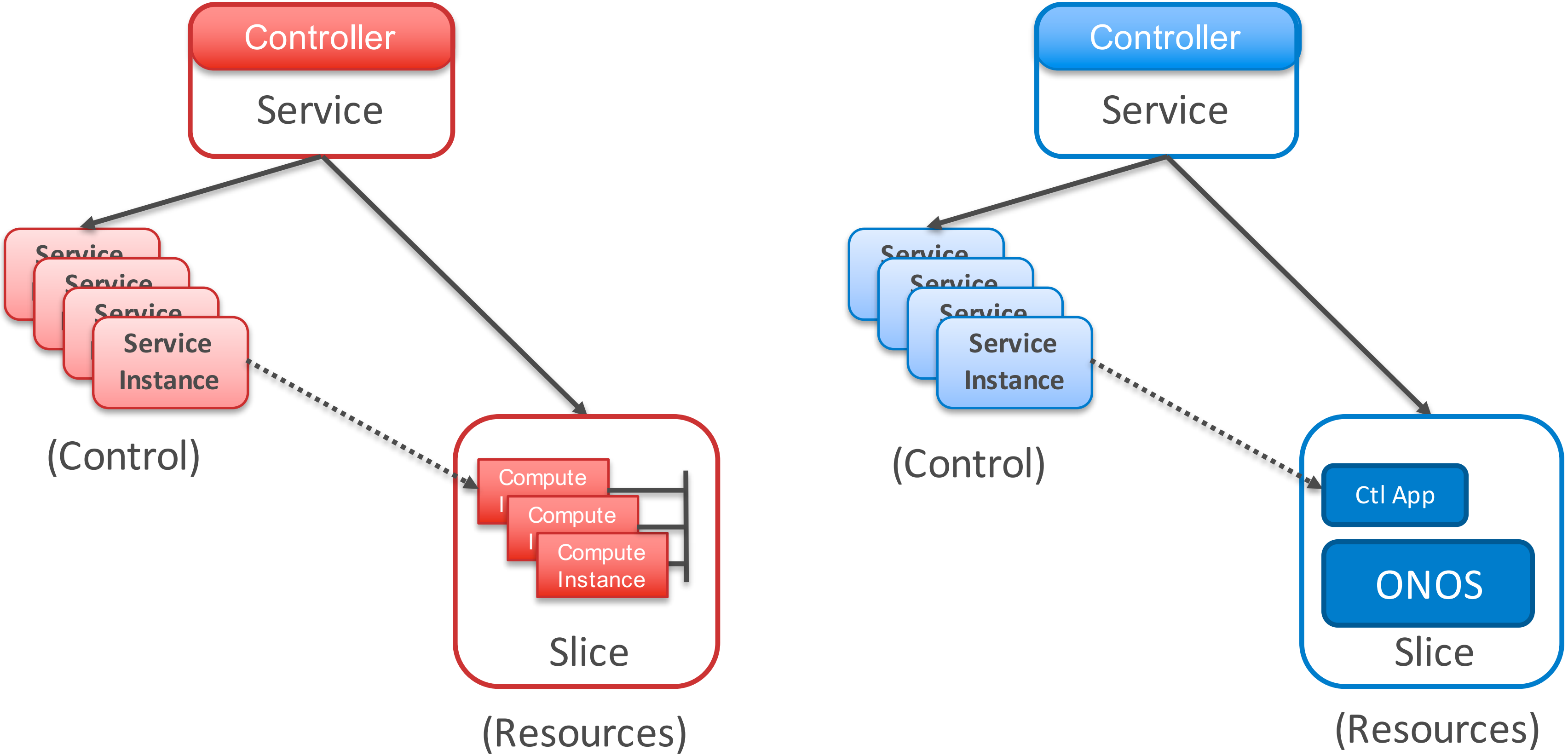
# XOS Generative Toolchain

Generated Code

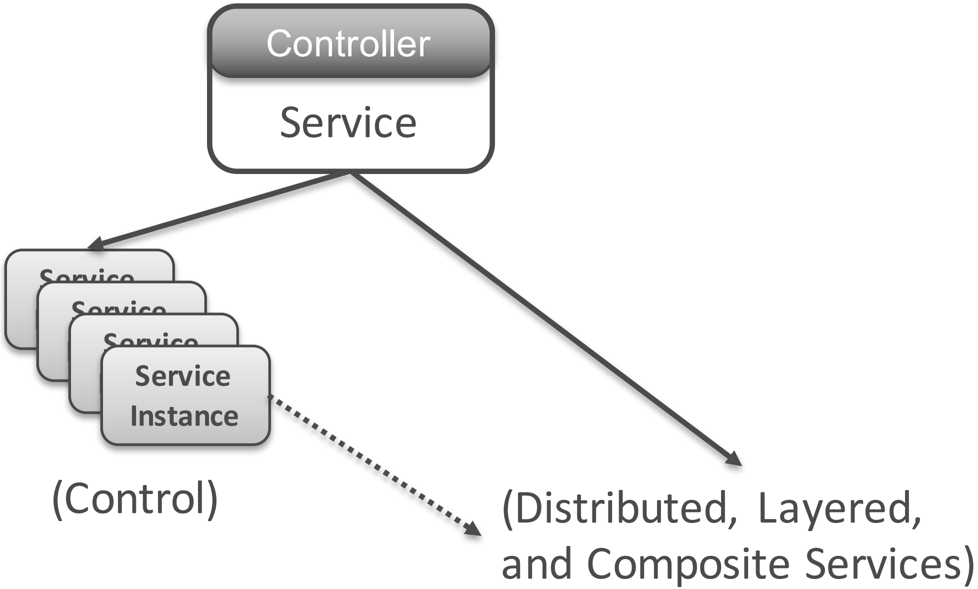
- API Tests
- Northbound Interfaces
- Enforce Security Policy
- Object Relation Mapper
- Synchronizer Framework



# Core Models


Controller
Service
Service
Service
Service
Service Instance
(Control)
Compute
I
Compute
I
Compute Instance
Slice
(Resources)
Controller
Service
Service
Service
Service
Service Instance
(Control)
Ctl App
ONOS
Slice
(Resources)

# Core Models



Controller

Service

Service

Service

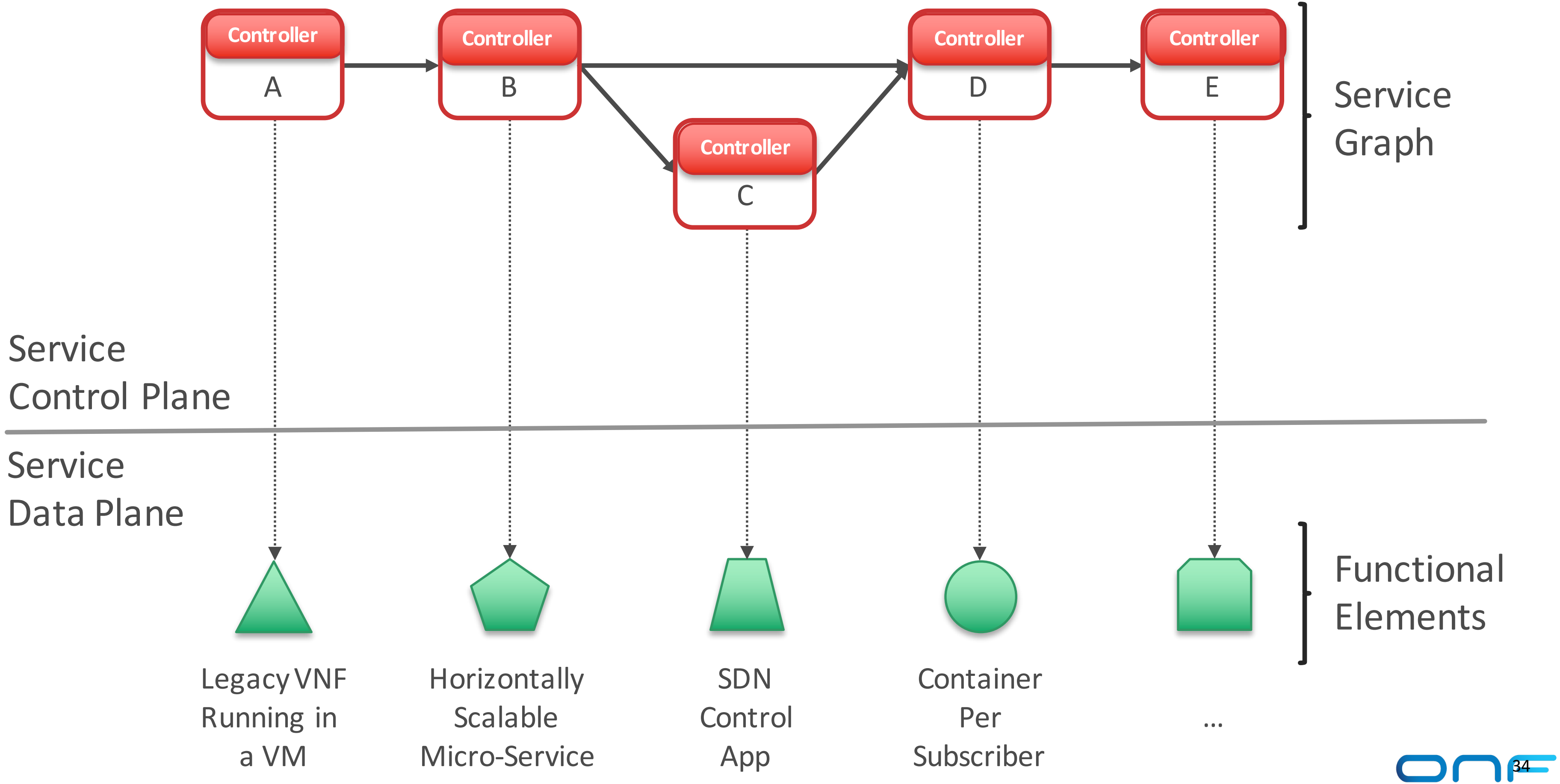
Service

Service Instance

(Control)

(Distributed, Layered, and Composite Services)

# Service Control and Data Planes



Service Graph

Service Control Plane

Service Data Plane

Functional Elements

Legacy VNF Running in a VM

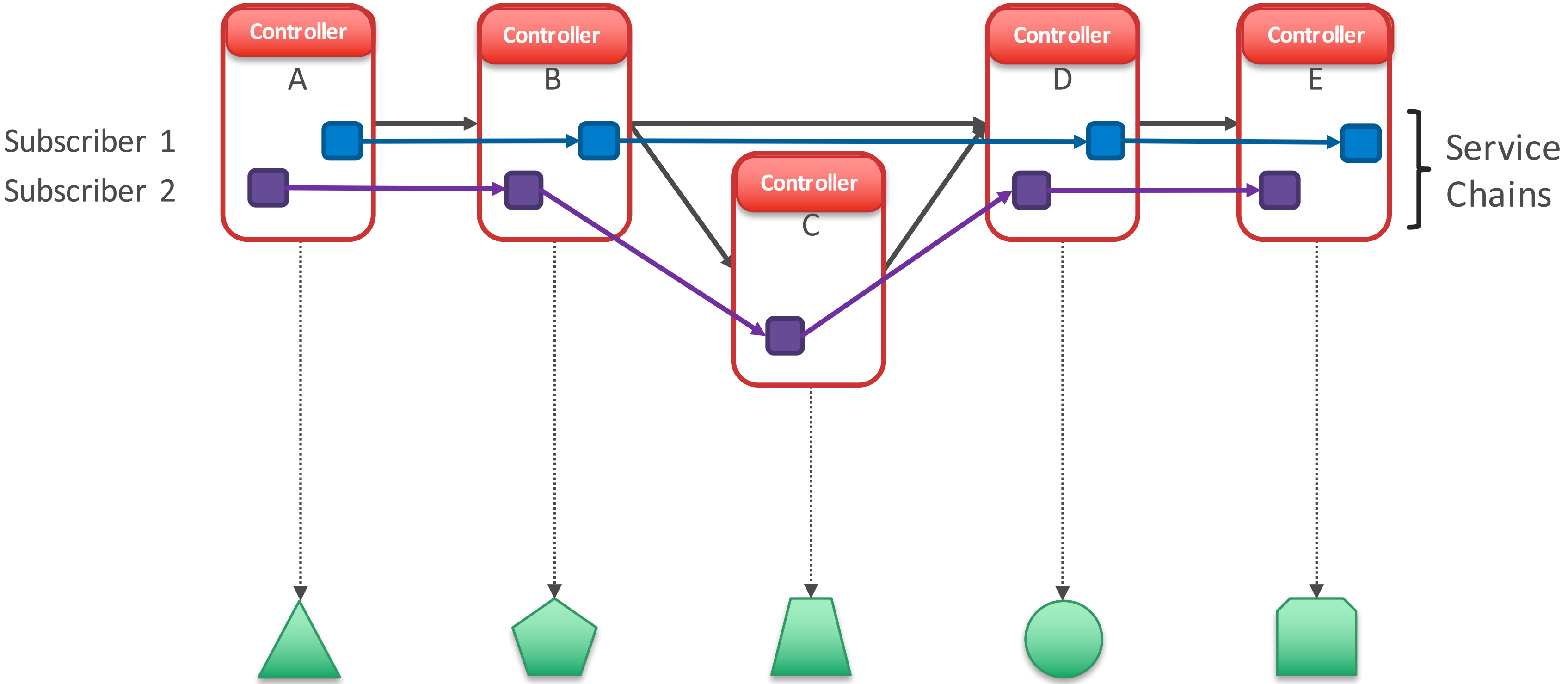
Horizontally Scalable Micro-Service

SDN Control App

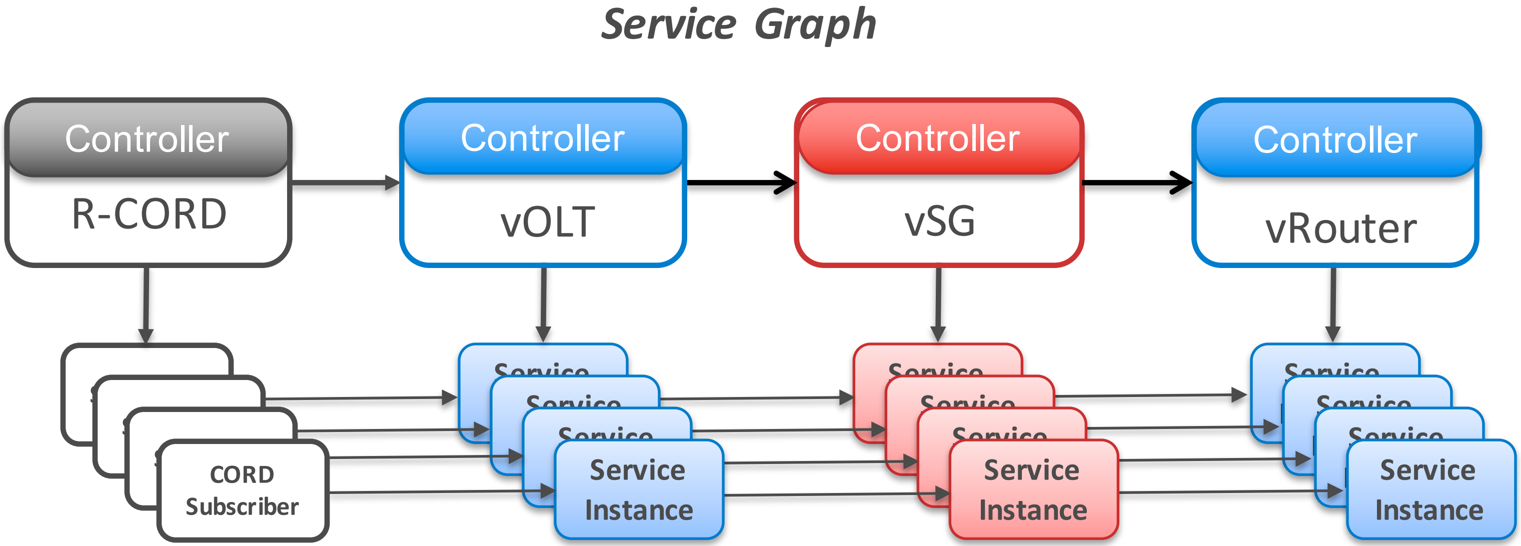
Container Per Subscriber

…

# Service Graph and Service Chains

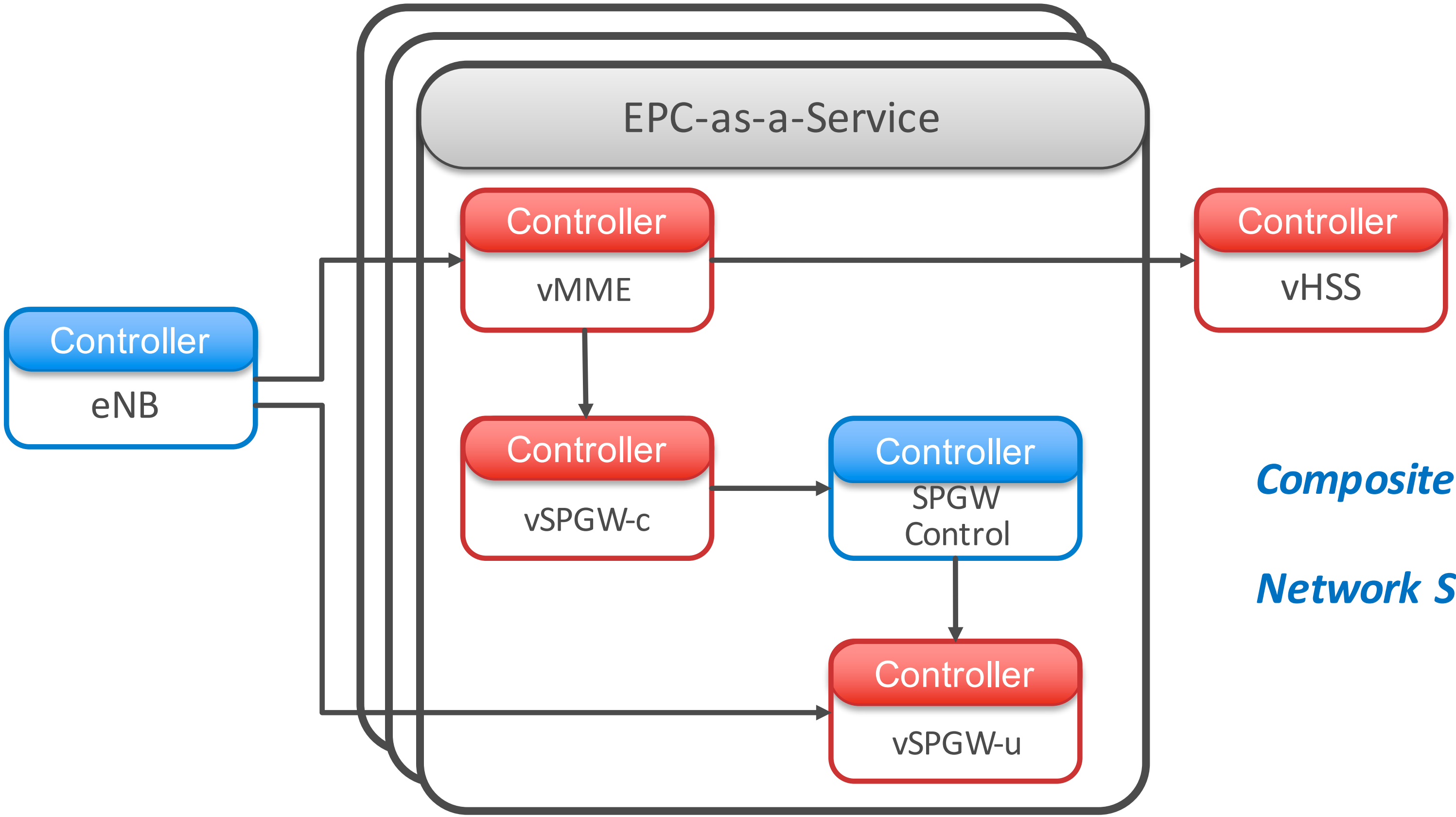


# Service Graph – Residential Case



*Service Chain = At the granularity of subscribers (or subscriber classes)*

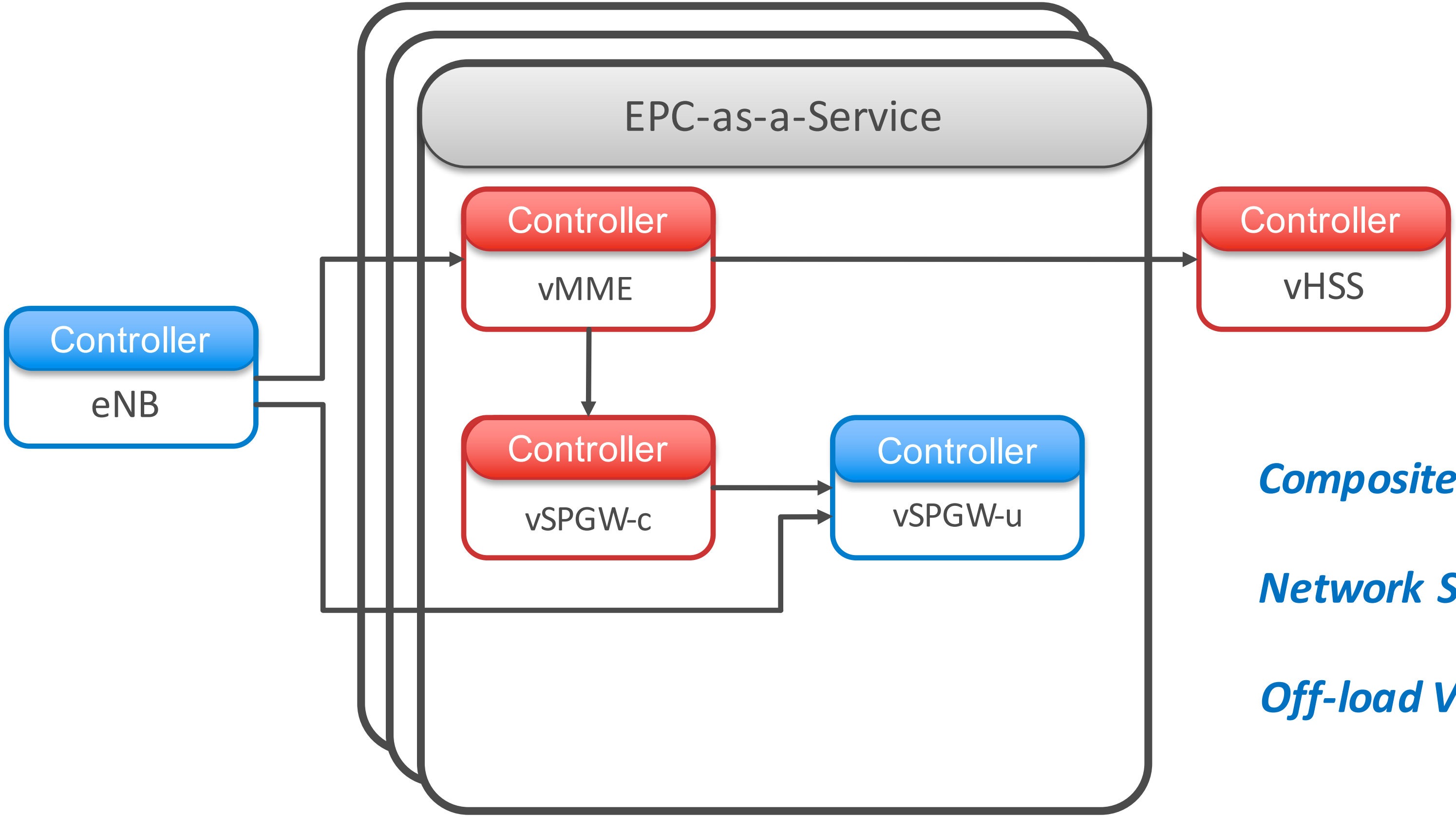
# Service Graph – Mobile Case



***Composite Services***

***Network Slicing***

# Service Graph – Mobile Case



***Composite Services***

***Network Slicing***

***Off-load VNFs (to fabric)***

# Conclusion

CORD is a Multi-Access Edge Cloud

- Includes both Access-as-a-Service and Software-as-a-Service
- Uses Merchant Silicon and Function Disaggregation

XOS is a Framework for Configuring and Operating a Cloud Platform

- Supports Services as a Unifying Abstraction (implementation agnostic)
- Decouples Service Control Plane and Service Data Plane
- Uses Declarative Models and Generative Toolchain to Specify & Enforce Behavior

# Conclusion

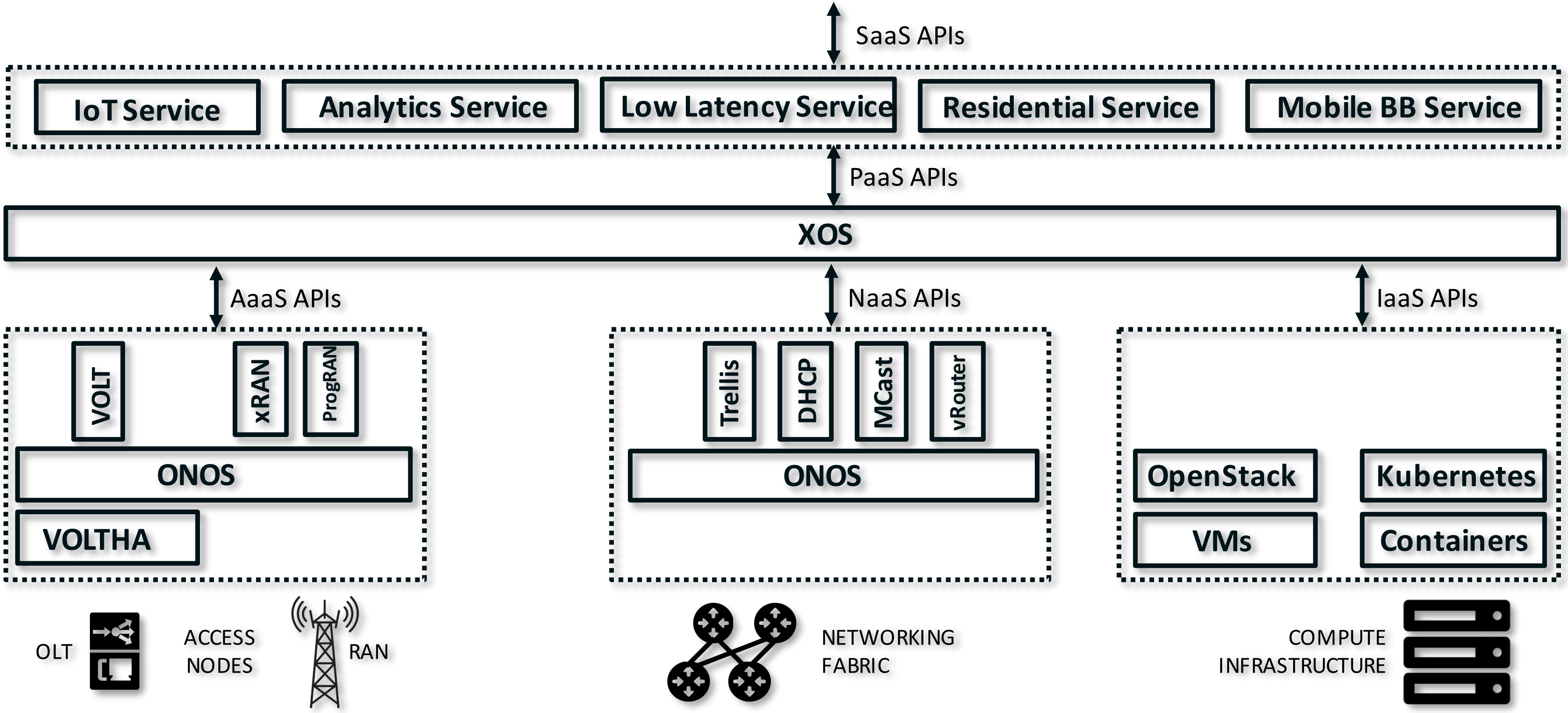
CORD integrates *Access-as-a-Service* into a multi-tenant cloud platform

- Disaggregated functionality with a mix of server- and switch-based implementations

XOS integrates the disaggregated components into a coherent whole (PaaS)

- Programmable framework with visibility and control at the granularity of subscribers

# Conclusion


SaaS APIs
IoT Service
Analytics Service
Low Latency Service
Residential Service
Mobile BB Service
PaaS APIs
XOS
AaaS APIs
NaaS APIs
IaaS APIs
VOLT
xRAN
ProgRAN
ONOS
VOLTHA
Trellis
DHCP
MCast
vRouter
ONOS
OpenStack
Kubernetes
VMs
Containers
OLT
ACCESS NODES
RAN
NETWORKING FABRIC
COMPUTE INFRASTRUCTURE