

IBM Research

Openstack Networking

(formerly known as Quantum)

Plugin and Extensions for Cloud Applications

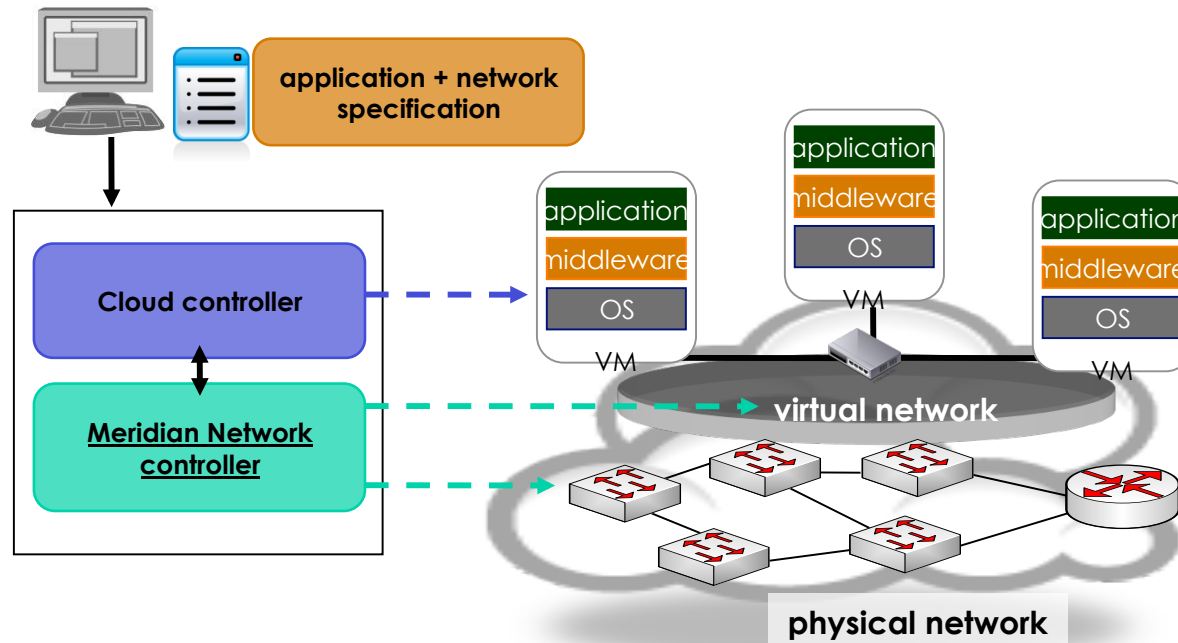
Mohammad Banikazemi

Outline

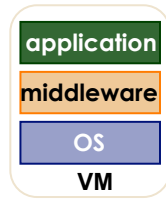
- Software Defined Networking
- Meridian, An OpenFlow Controller
- Quantum Plugin for Meridian
- Quantum Extensions
- Ongoing Work

Meridian

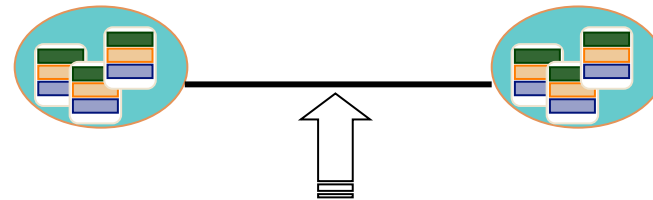
- An SDN Platform for Cloud Network Services



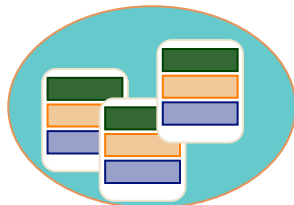
Meridian: The Model



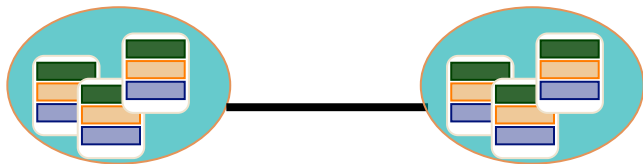
endpoint
(managed vNIC, VM, etc.)



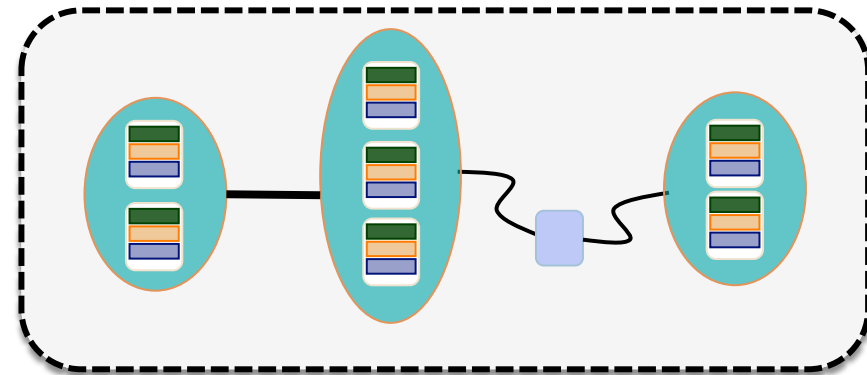
service
attach services to a segment
Filters/ACLs
middlebox
QoS



group
logical grouping of endpoints

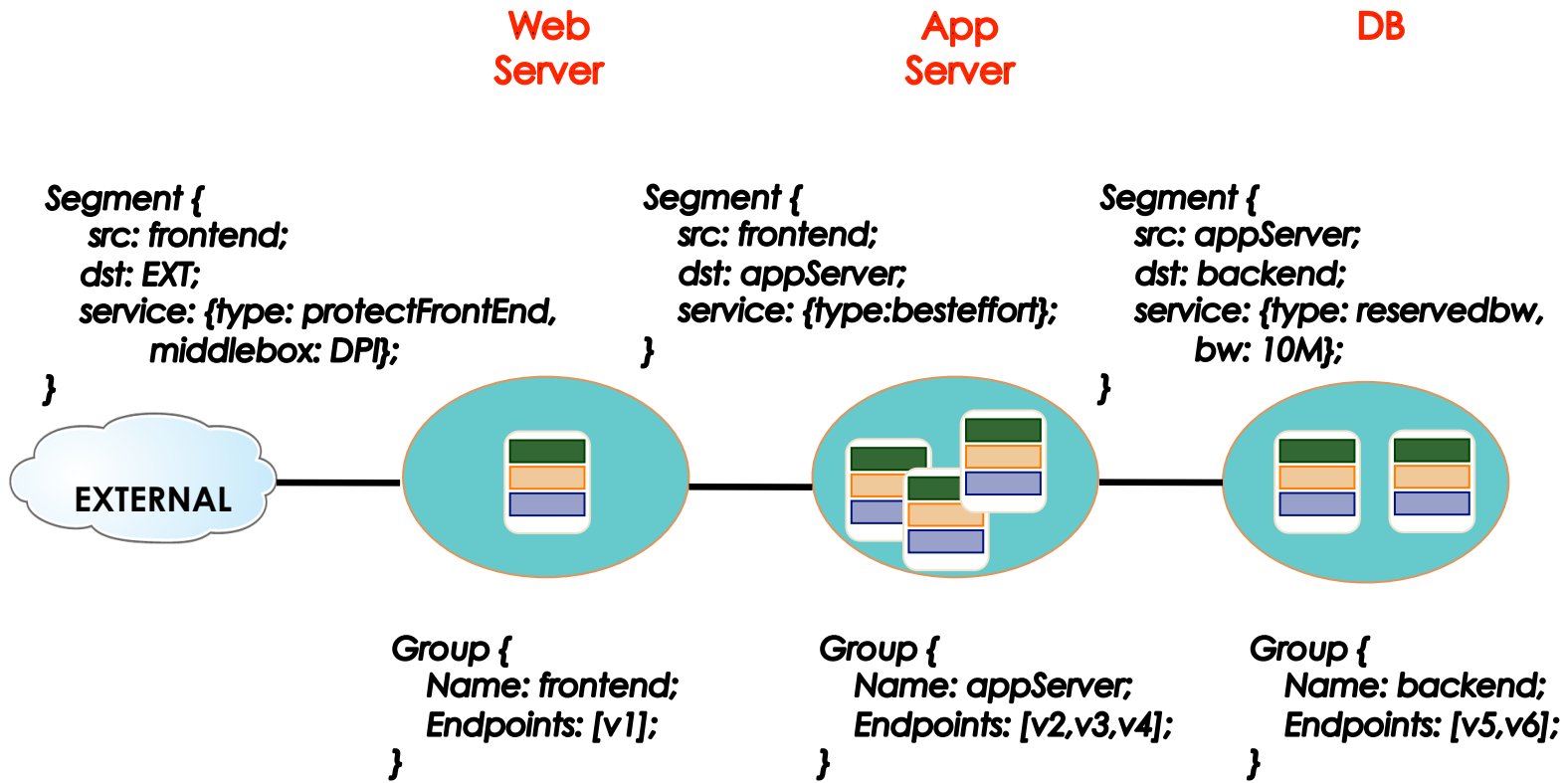


segment
bidirectional virtual link



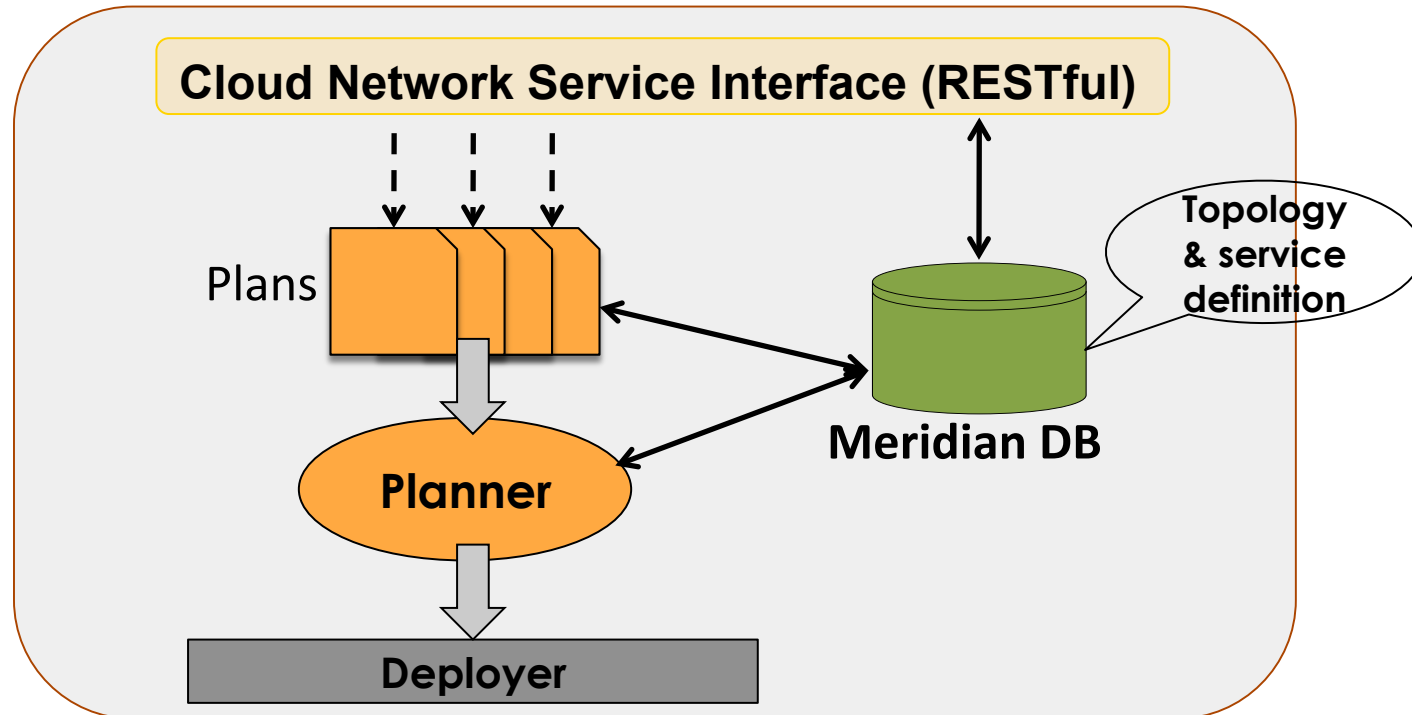
virtnet
application virtual network

3-tier Web Application



Meridian Controller Architecture

IBM Research



System components:

- Abstract network service interface (REST APIs)
- Planning architecture to support concurrent network installation and updates
- Deployment drivers to support OpenFlow/non-Openflow control channels
- Currently implemented on top of **Floodlight**

Meridian APIs

Category	API calls	Description
Endpoints	<ul style="list-style-type: none">• defineEndPoint• deleteEndPoints• updateEndPoints	Endpoints identify interfaces that are connected via segments / networks
Groups	<ul style="list-style-type: none">• defineGroup• deleteGroup• updateGroup	Groups are collections of endpoints that receive the same services or connectivity rules
Services / properties	<ul style="list-style-type: none">• defineService• deleteService• updateService	Services attached to virtual segments; Current services incl. routing type , filtering rules, and middlebox traversal
Segments (virtual links)	<ul style="list-style-type: none">• defineVirtNetSegment• deleteVirtNetSegment• updateSegmentService	Segments are virtual links that establish connectivity between groups of endpoints
Virtual networks	<ul style="list-style-type: none">• defineVirtualNetwork• deleteVirtualNetwork• updateVirtualNetwork• validateVirtualNetwork• installVirtualNetwork	Virtual network is a set of defined groups and segments for an application.

Openstack Networking

- Network
- Subnet
- Port

Quantum API

- Create Network
- Update Network
- Delete Network
- Create Subnet
- Update Subnet
- Delete Subnet
- Show Port
- Create Port
- Delete Port
- ...

Meridian Plugin

Quantum service

Quantum API

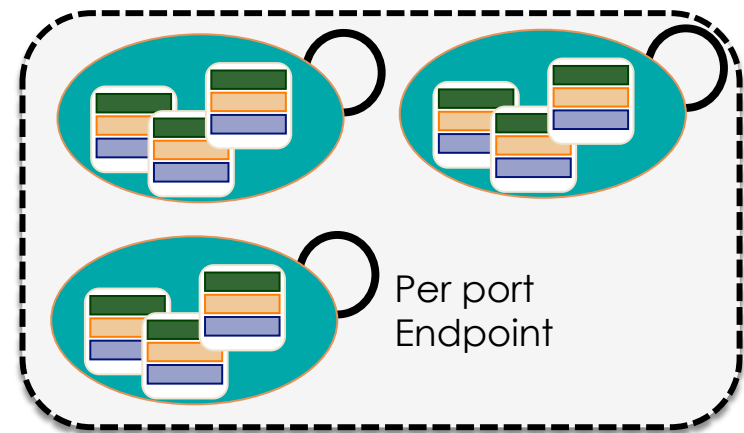
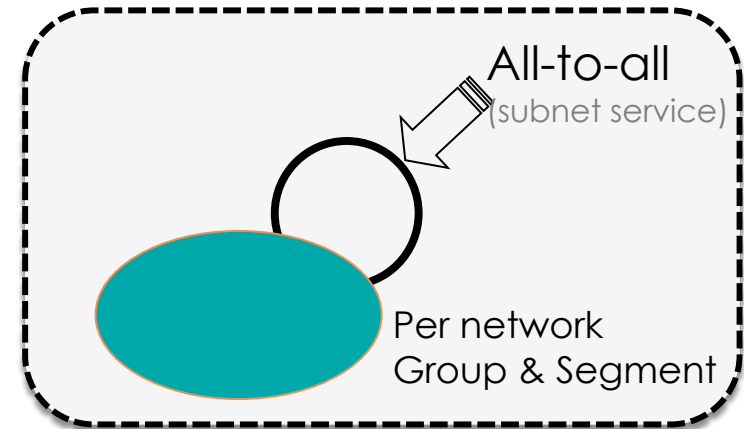
- Create Network
- Update Network
- Delete Network
- Create Subnet
- Update Subnet
- Delete Subnet
- Show Port
- Create Port
- Delete Port
- ...

Meridian plugin

- Create Network
- Update Network
- Delete Network
- Create Subnet
- Update Subnet
- Delete Subnet
- Show Port
- Create Port
- Delete Port
- ...
- **Extensions**



Per tenant virtual network



Quantum Extensions - 1

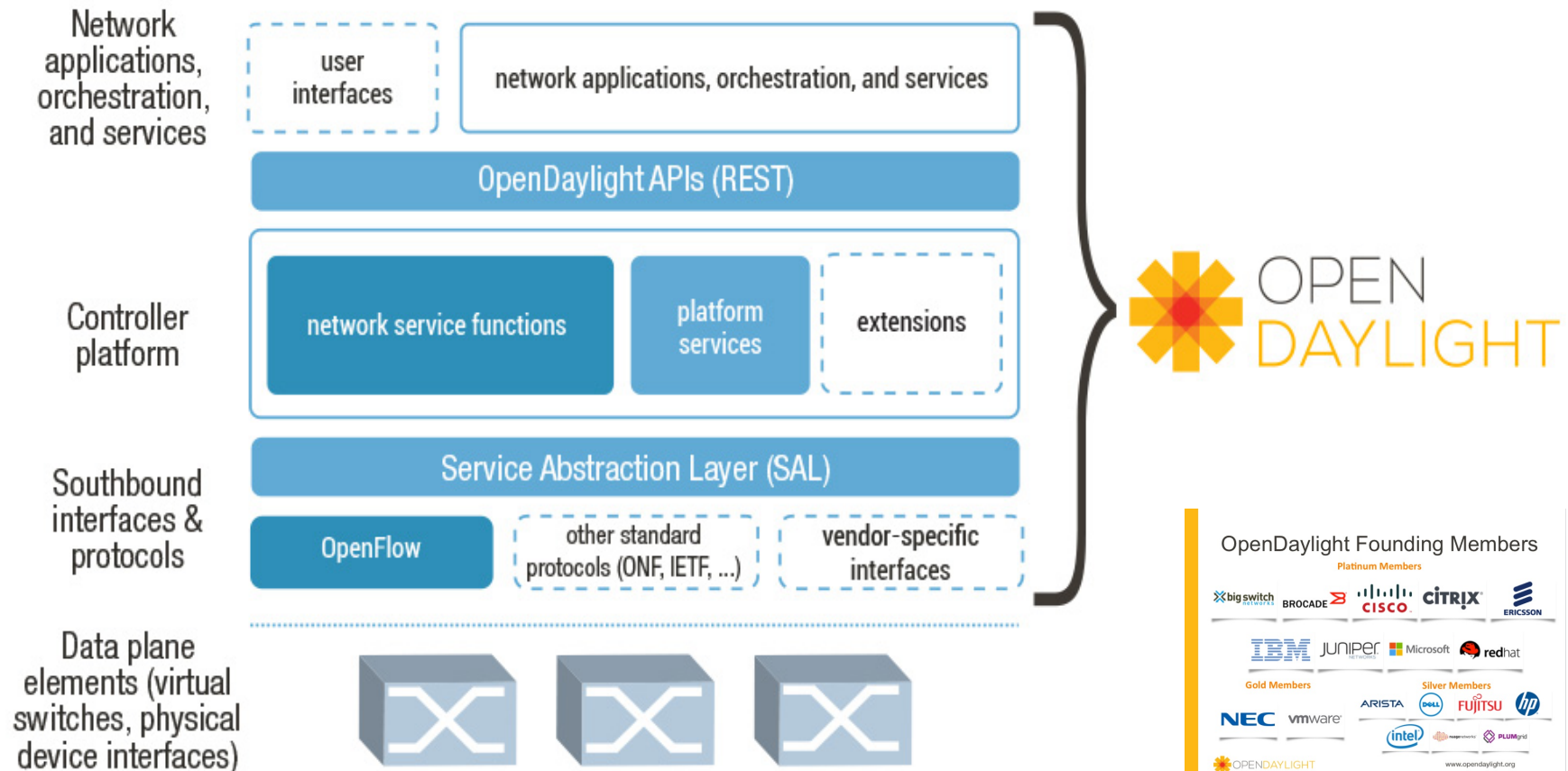
- ❑ Supports the Router and Floating IP extensions
- ❑ Exposes the underlying Meridian model through Quantum extensions
 - ❑ Virtnet
 - ❑ Groups
 - ❑ Segments
 - ❑ Services
 - ❑ Endpoints

Quantum Extensions - 2

- ❑ Provides a single method for providing new services for Quantum networks
 - ❑ Source Network
 - ❑ Destination Network
 - ❑ Ordered list of filters
 - ❑ Ordered list of actions
- ❑ Working towards using Quantum Service Insertion model

OpenDaylight

- An open, reference framework for programmability and control through an open source SDN solution



OpenDaylight Founding Members

Platinum Members

big switch | BROCADE | CISCO | citrix | ERICSSON

IBM | JUNIPER | Microsoft | redhat

Gold Members

NEC | vmware | ARISTA | DELL | FUJITSU | hp

Silver Members

intel | PLUMgrid

www.opendaylight.org

Team Members

IBM Research

- Mohammad Banikazemi
- David Olshefski
- John Tracey
- Guohui Wang

Links



- There is an open source implementation of the controller available as:

OpenFlow Technology Preview Controller (OFTPC)

<https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=0e8ee4de-b06e-4eb9-a1bc-c688314a5d61>

- OpenDaylight
 - <http://www.opendaylight.org/>

Floodlight Controller Plugin with Openstack Quantum

Damian Igbe, PhD



Agenda

Overview

Quantum's Architecture

Floodlight with Quantum

Conclusion



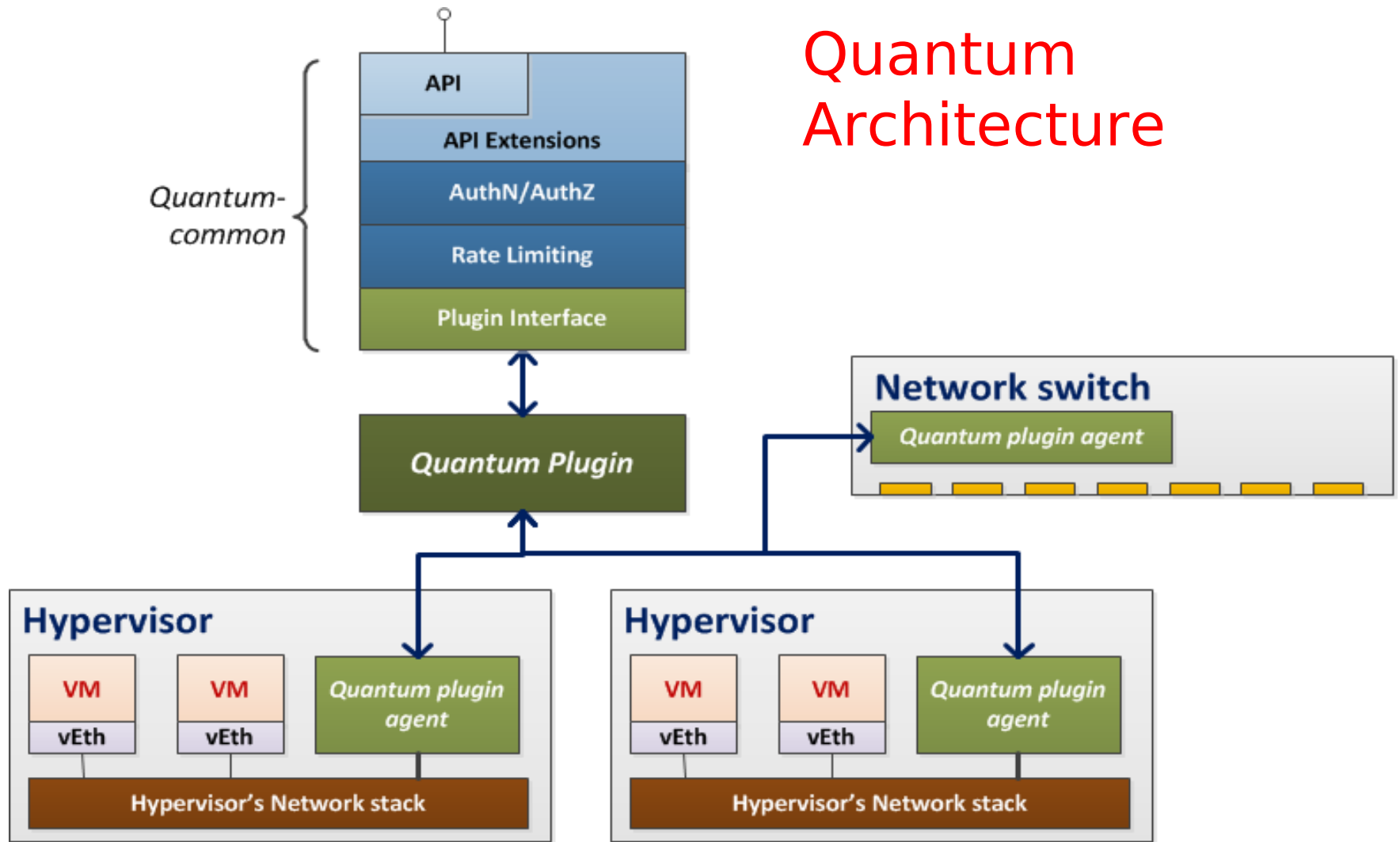
Floodlight is a completely
open, free , apache licensed,
java based openflow
controller

Floodlight Features

- Developer friendly Apache license
- Easy to use, extensible Java development environment
- Enterprise grade Core engine used and Supported by Big Switch Networks (running in Production today)
- Supports a broad range of physical and virtual OpenFlow switches
- OF 1.0 compliant today - future OF versions on the way



Quantum Architecture



Floodlight Plugin



REST Proxy for Floodlight - uses OpenFlow and Floodlight controller

Cisco UCS

Isolation is realized by VLAN and net-profiles – related to the Cisco UCS products

LinuxBridge

Isolation with VLAN interfaces and Linux bridge – pure Linux solution

Nicira NVP

Proxy for the Nicira platform - related to Nicira products

RYU

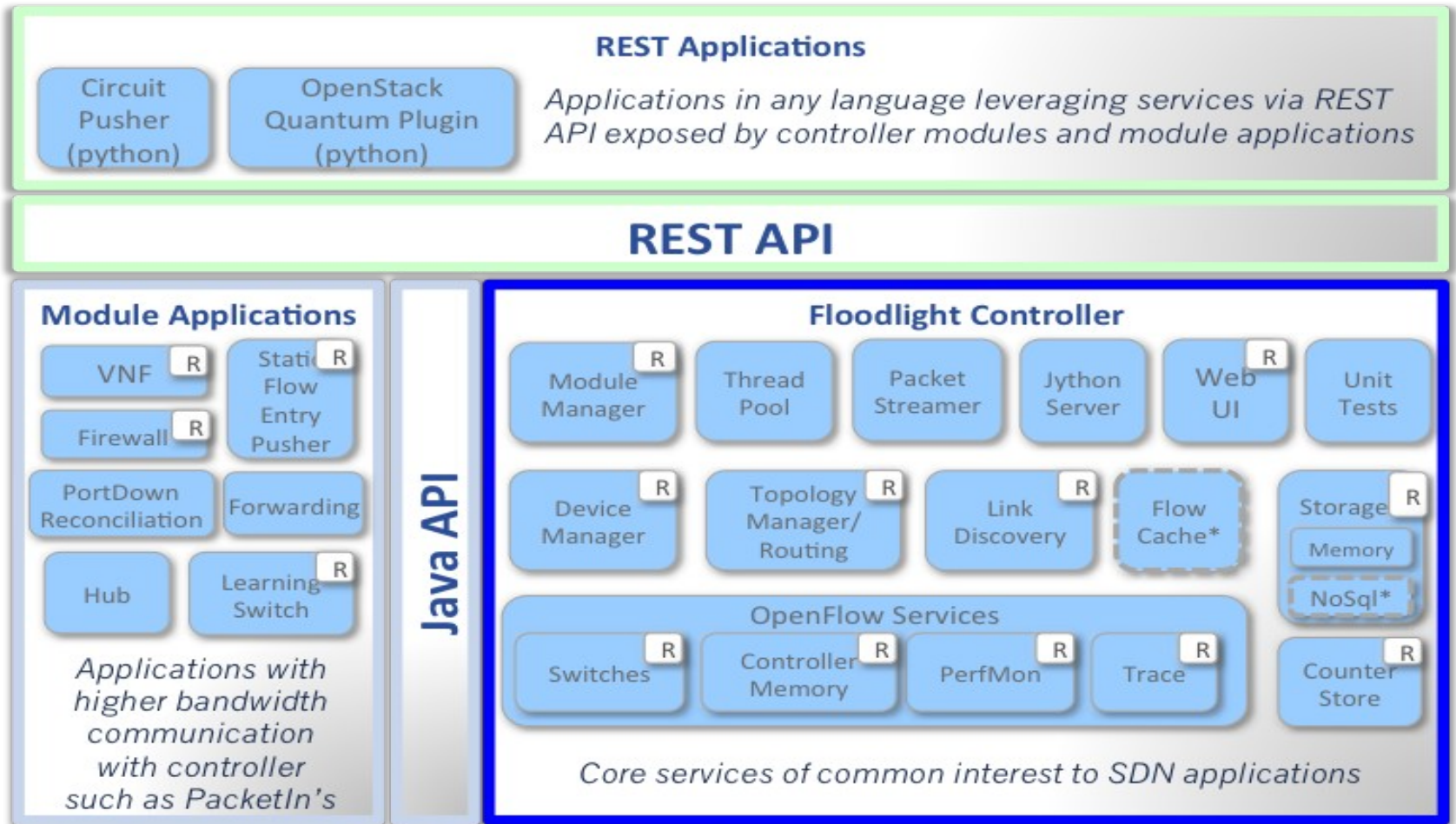
Proxy for the RYU network OS - uses OpenFlow and RYU controller

Open vSwitch

Isolated networks with OVS and L2 in L3 tunneling



Floodlight Architecture



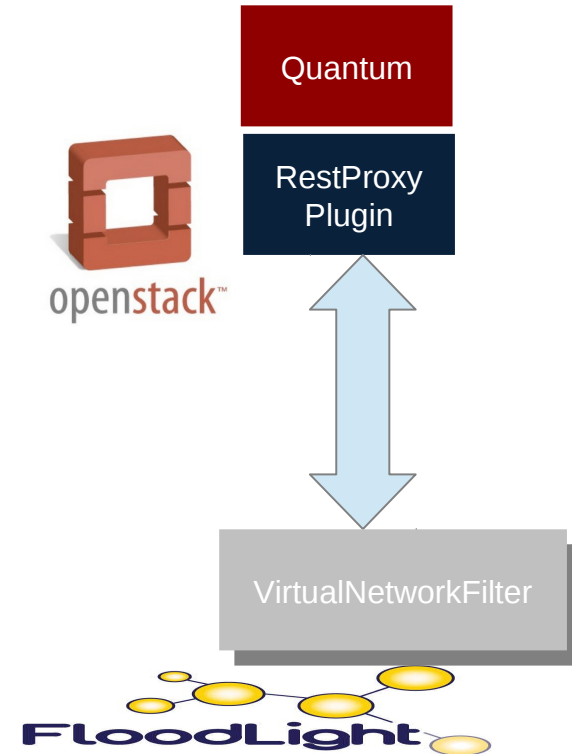
* Interfaces defined only & not implemented: FlowCache, NoSql

Floodlight- Quantum Integration

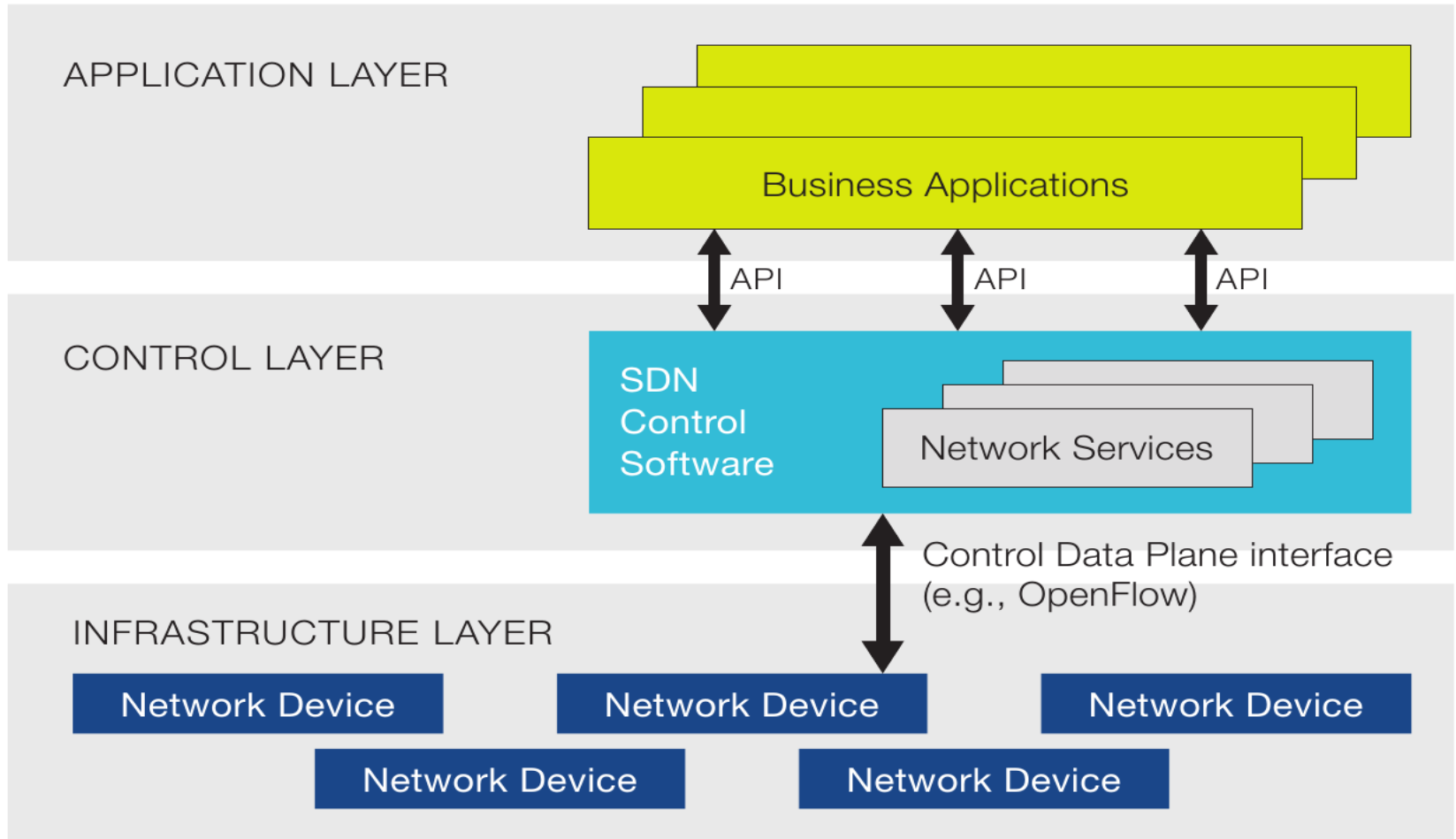
2 Components:

First: QuantumRestProxy plugin runs inside Quantum module in OpenStack

Second: VirtualNetworkFilter implements layer 2 isolation based on MAC Floodlight



SDN Architecture



Creating Networks/Adding Hosts with NVF

Creating a virtual network named "VirtualNetwork1", the ID is "NetworkId1", the gateway is "10.0.0.7", and the tenant is "default" (which is currently ignored).

```
curl -X PUT -d '{ "network": { "gateway": "10.0.0.7", "name":  
"virtualNetwork1" } }' http://localhost:8080/network  
Service/v1.1/tenants/default/networks/NetworkId1
```

Adding a host to VirtualNetwork1 with the MAC address "00:00:00:00:00:08" and the port "port1".

```
curl -X PUT -d '{"attachment": {"id": "NetworkId1", "mac":  
"00:00:00:00:00:08"}}'  
http://localhost:8080/networkService/v1.1/tenants/default/networks/NetworkId1/hosts/port1
```

```
curl http://localhost:8080/networkService/list/json
```



FloodLight CORE Restful APIs

- 1) VirtualNetwork Filter (VNF)
- 2) Static Flow Entry Pusher
- 3) Circuit Pusher
- 4) Firewall API and Load balancer

Floodlight Controller Lab Setup



2 VMs in use:

First VM: An OpenStack Quantum in a virtual machine with REST-Proxy Plugin and open vSwitch managed by floodlight

Second VM: Independent floodlight controller running VNF

```
touch /opt/floodlight/floodlight/feature/quantum  
java -jar target/floodlight.jar -cf  
src/main/resources/quantum.properties
```

Q & A

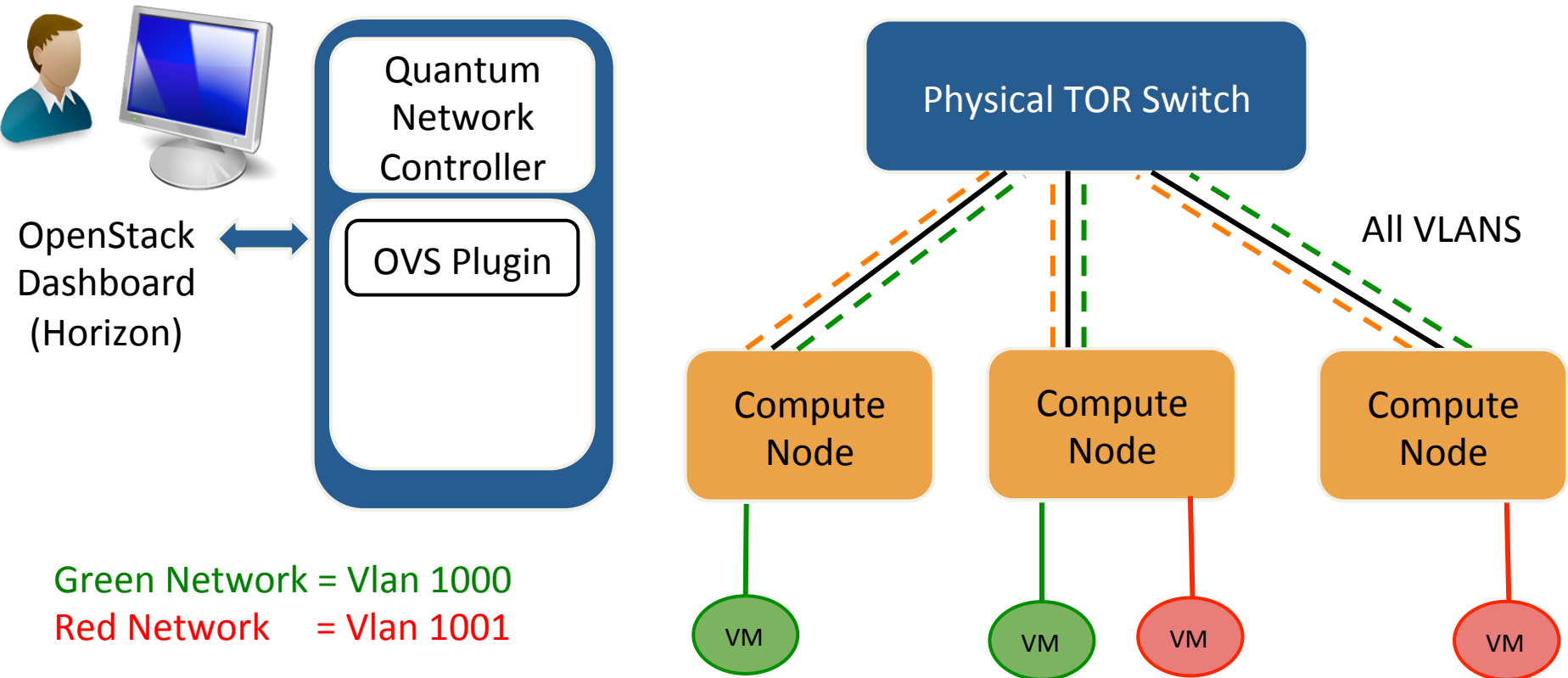
ARISTA

Extending Quantum for the Physical Network
Andre Pech

Current State of the World

- Quantum is great at orchestrating virtual tenant networks
 - But the physical network is left to the user
- Some physical network vendors have developed their own quantum plugins
 - But the model is wrong - your choice of virtual switch should not be slaved to your physical network infrastructure
 - Result is a proprietary, single-vendor solution

Problem



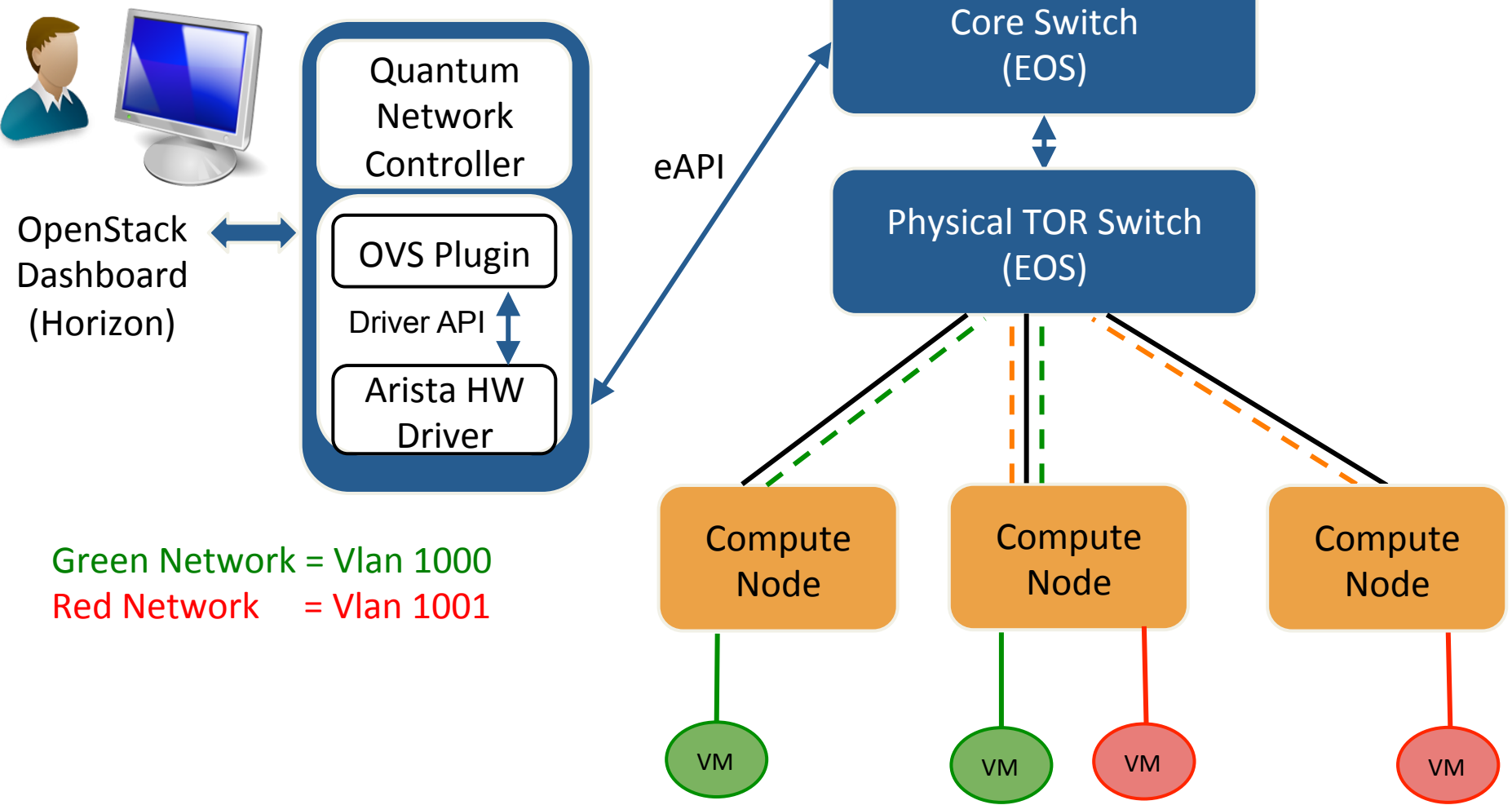
Result for the physical network is either:

- suboptimal network design,
- manual provisioning of tenant networks,
- or need an outside system to automate

Extending Quantum and the OVS Plugin for the Physical Network

- Goal - open, multi-vendor solution for seamless virtual and physical network provisioning
- Extended the OVS Quantum Plugin with an abstract API for calling into one or more hardware driver plugins
- Re-empower the networking team by giving them visibility into how virtual tenant networks map onto the physical network

Solution



Summary

- Extending Quantum and the OVS Plugin for physical networks and integrating with EOS enables automated provisioning of network segments across both the virtual and physical infrastructure in an open and multi-vendor solution
- Lots of potential for future improvements by more tightly integrating the physical network into Quantum and OpenStack