



Is OPNFV just an OpenStack extension?

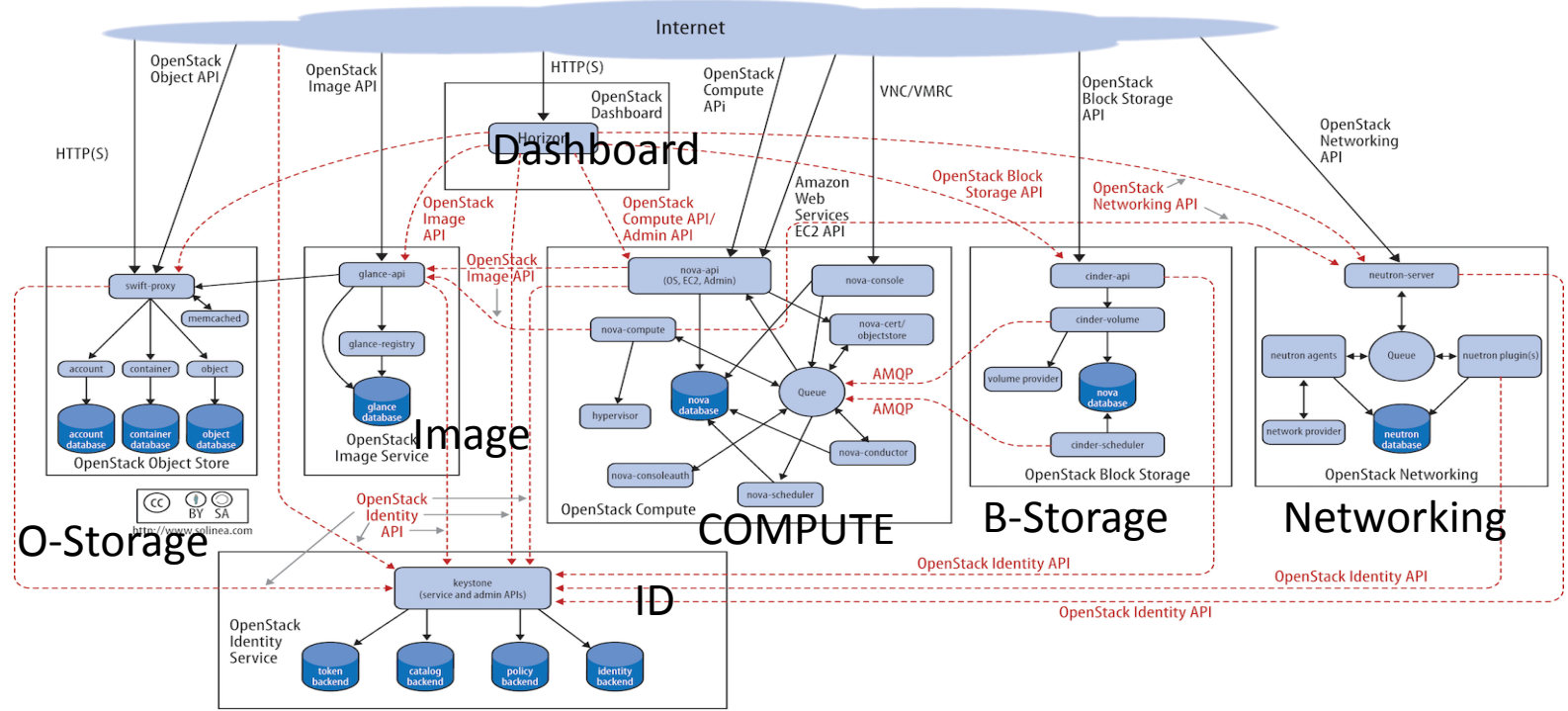
Ulrich Kleber

Prakash Ramchandran

OpenStack: Mitaka IaaS Service

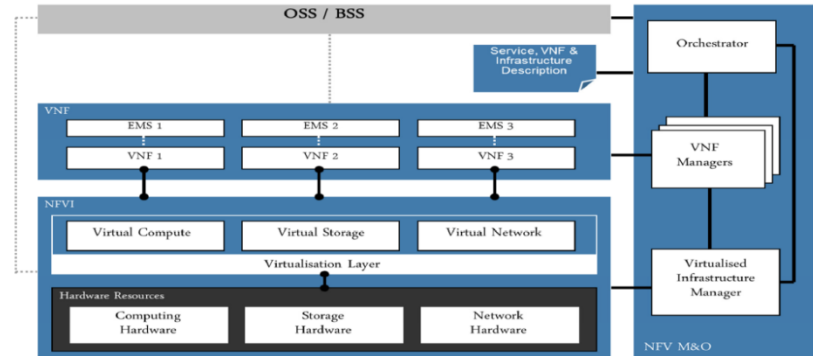
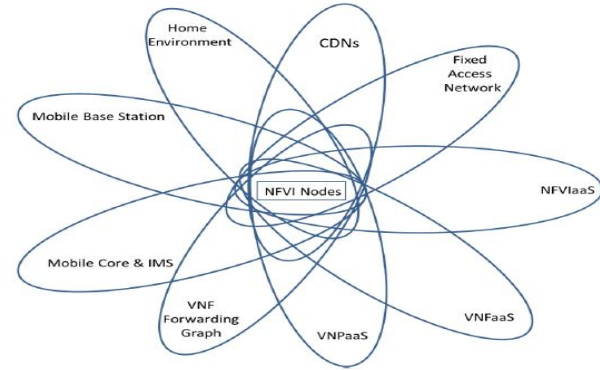


- Command-line interfaces (nova, neutron, swift, etc)
- Cloud Management Tools (Rightscale, Enstratus, etc)
- GUI tools (Dashboard, Cyberduck, iPhone client, etc)

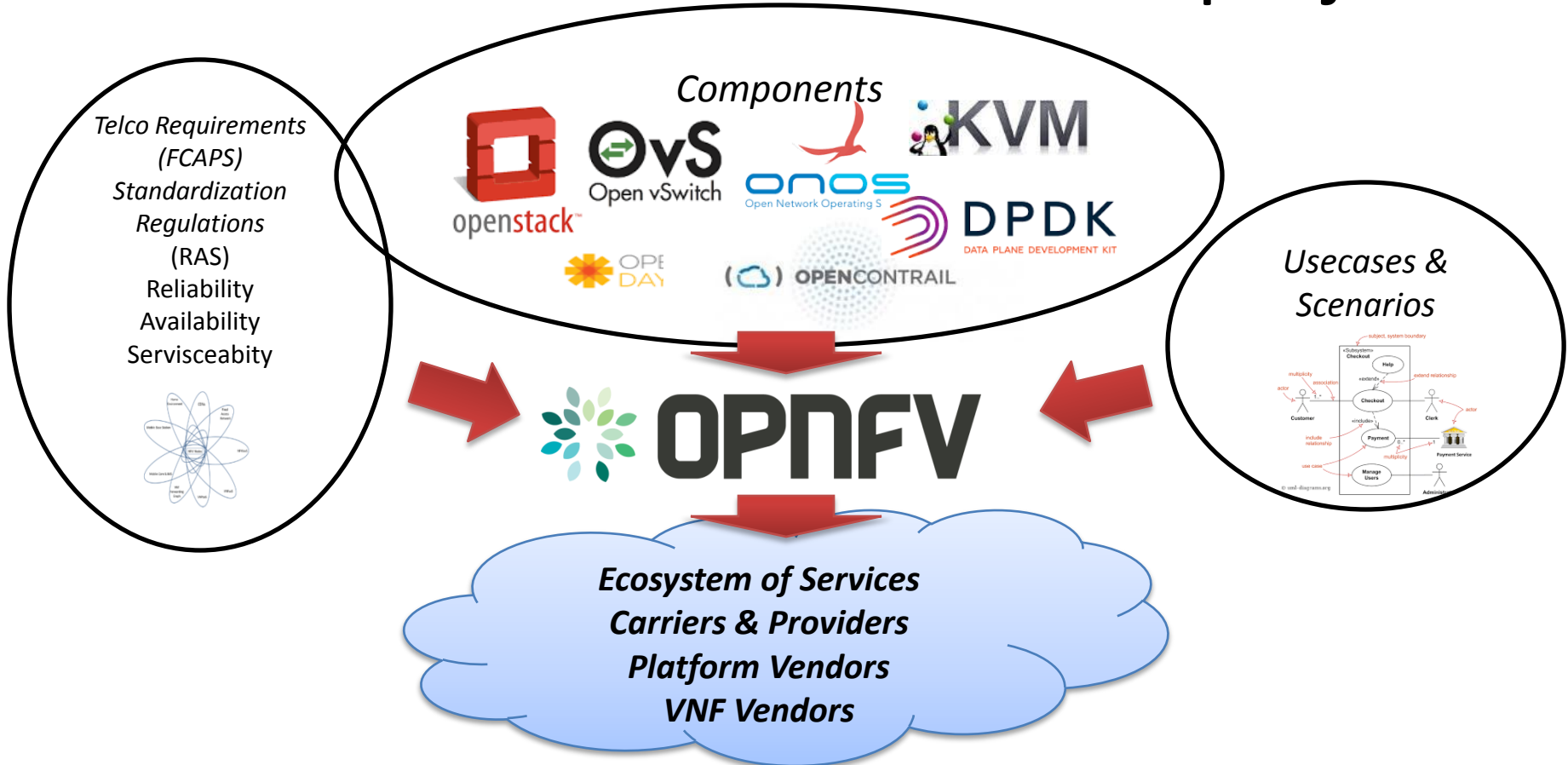


Telco Requirements (FCAPS)

- Fault Management
- Configuration Management
- Availability Management , Auto-Scaling , AAA
- Performance Management
- Security Management, Software Management



OPNFV's role as a midstream project

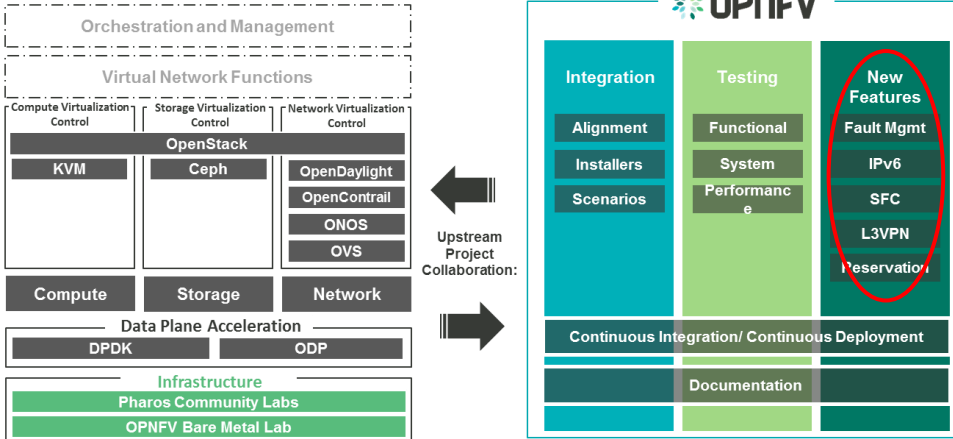


OPNFV Brahma Putra NFVaaS Service

- Release date: March 1st
 - Theme: Lab Ready
 - 30 Projects participated with > 165 developers
 - Based on Scenarios (combinations of components and features)
 - 4 different installers
 - Virtual or bare metal
 - 3 SDN Controllers (OpenContrail will be released later)
 - Optional Features (e.g. L3VPN, SFC)

OPNFV Brahma Putra provides:

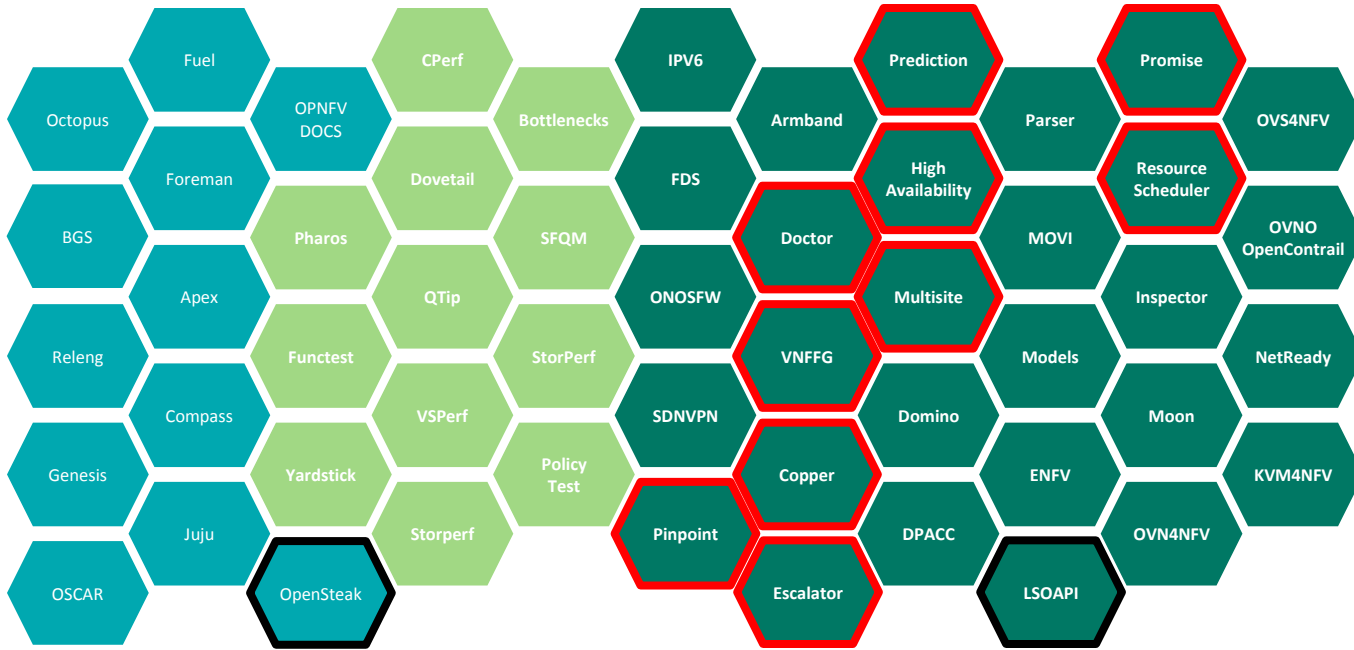
- » Target platform
 - NFVI and VIM according to ETSI reference architecture
- » Deployment tools
 - Installers: Fuel, Apex, Compass, Joid
- » Standard lab configuration
 - 2 payload nodes, 3 control nodes, jump server
- » Test Framework
- » Documentation
 - Brahma Putra Overview
 - Configuration Guide
 - User Guide
 - Release Documents






Additional provision of OPNFV

- » Requirement Documents
- » Community Labs

OPNFV Projects



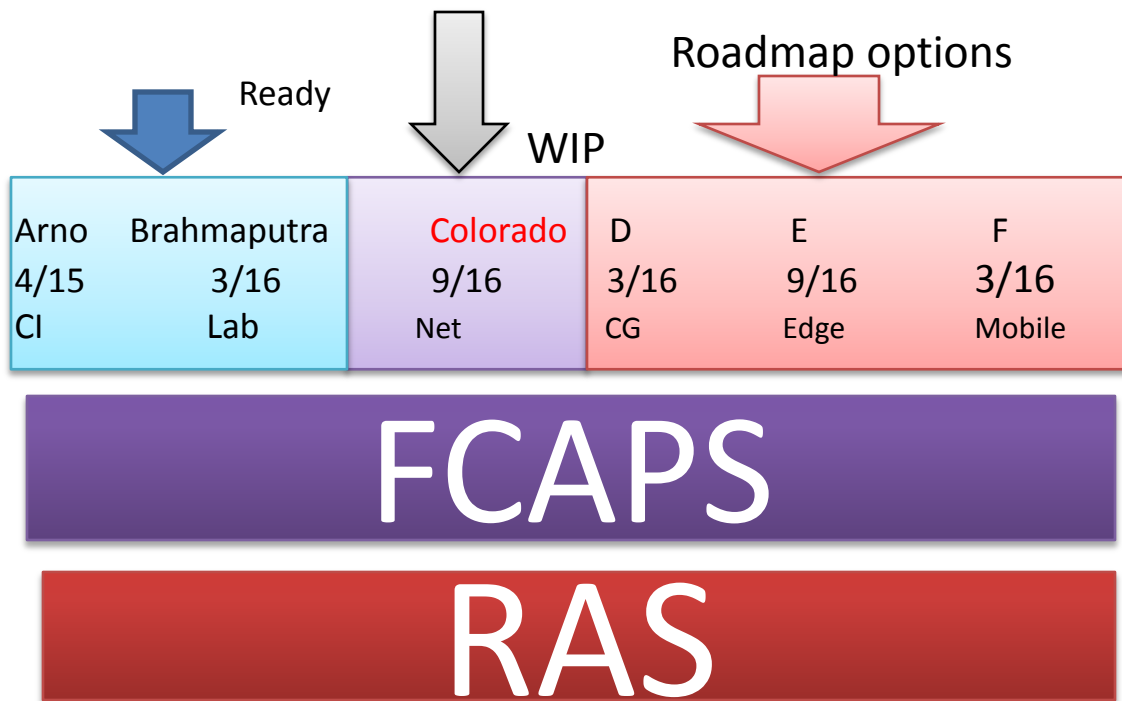
-  Build, Integration, Deployment, Documentation
-  Testing
-  Feature Requirements and Development

-  Terminated
-  Main upstream project OpenStack

- Projects have different way of working (categories)
- 10 Projects have OpenStack as main target
- But many project have different focus
- Some other projects take inputs from OpenStack, e.g. Fuel, Functest
- We are still learning (some projects already terminated)

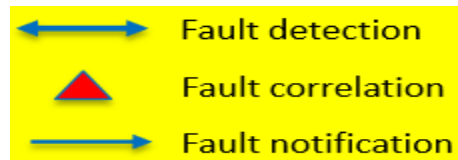
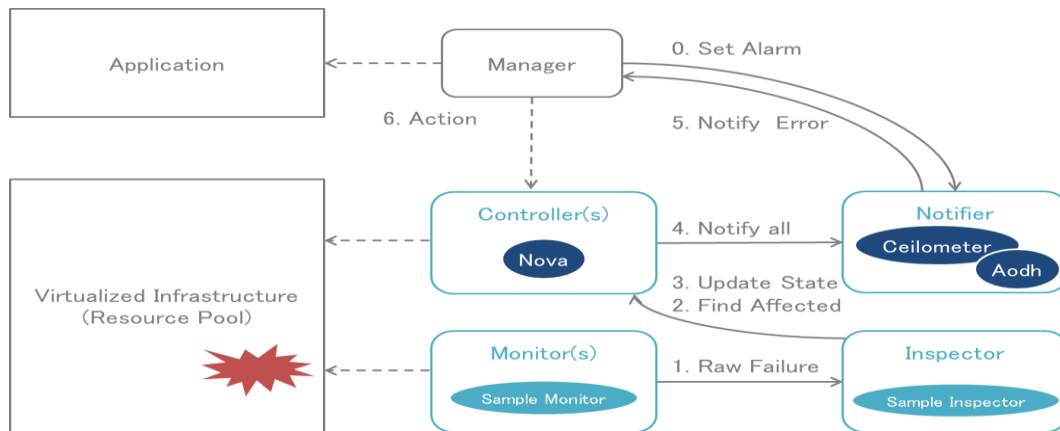
Targets of OPNFV's Requirements of OpenStack

- Maintenance
 - Fault Management
 - Resilience
 - Upgrade
- Network
 - SDN and vSwitch Integration
 - Telco Network Features (e.g. SFC, VPN)
 - Acceleration on data plane
- Orchestration & Interfaces
 - Modeling
 - Automation (e.g. Policies)
 - MANO
- Multiple Locations
 - Multisite/Tricircle
 - Policy Distribution
 - Network between the locations



Maintenance

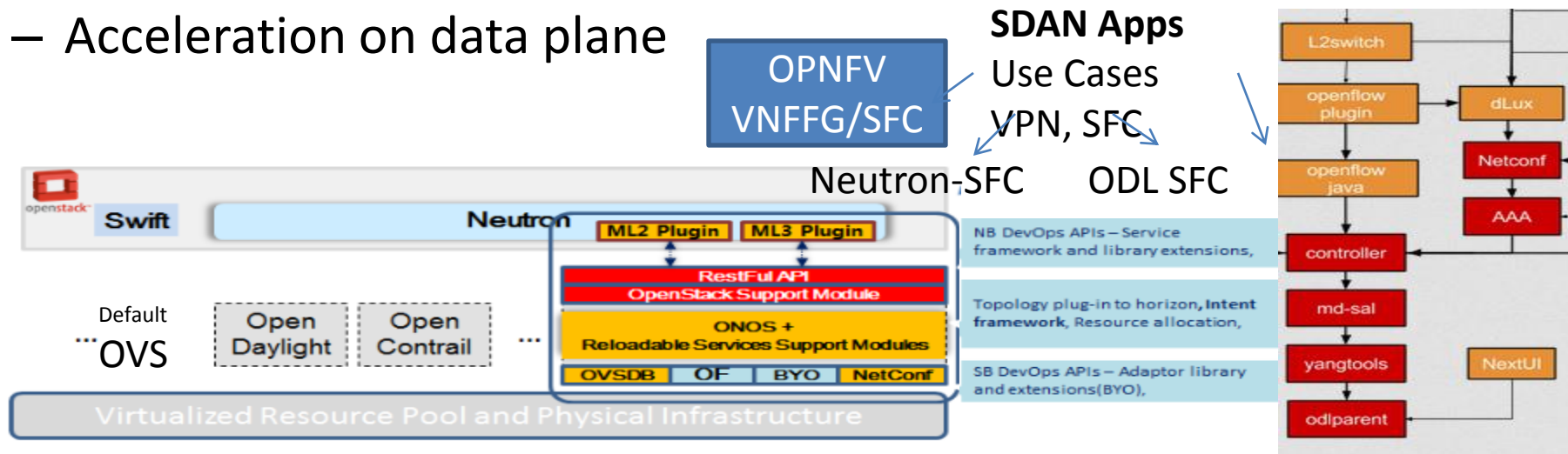
- Fault Management Architecture
(e.g. Immediate Notification, Consistent resource state awareness, ..)
- Resilience (Cloud Control, Network, Services)
- Upgrade (seamless for Services)



Network is Computing!!!

SDN-C Diversity

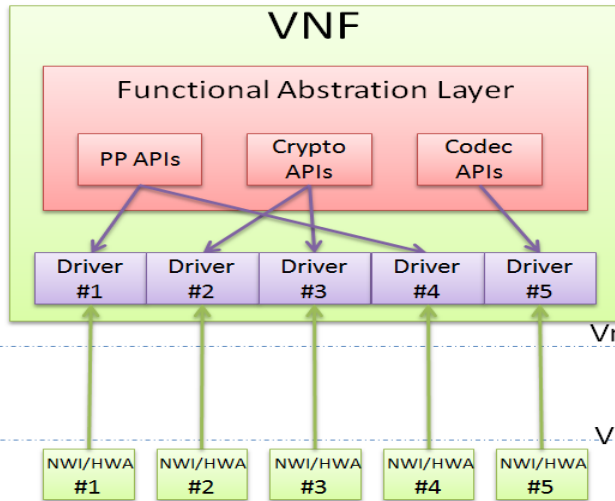
- SDN and vSwitch Integration
- Telco Network Features (e.g. SFC, VPN)
- Acceleration on data plane



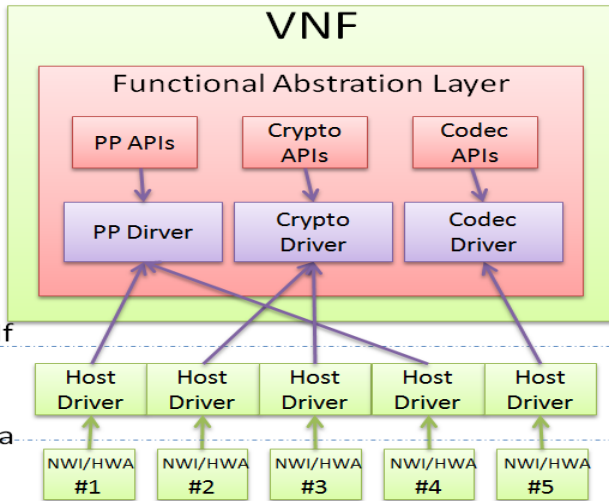
Fast Forwarding - DPACC & FDS

- Packet Processing Abstractions & Fast Data Stack

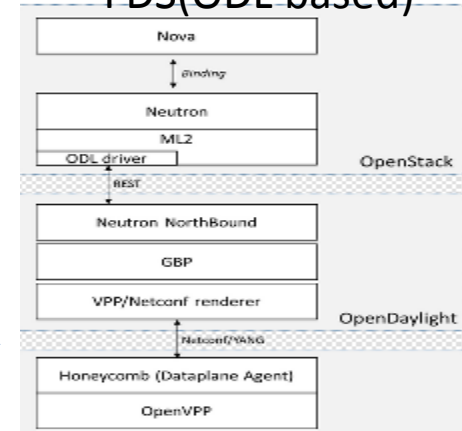
"Pass-Through" Model



"Fully Intermediated" Model

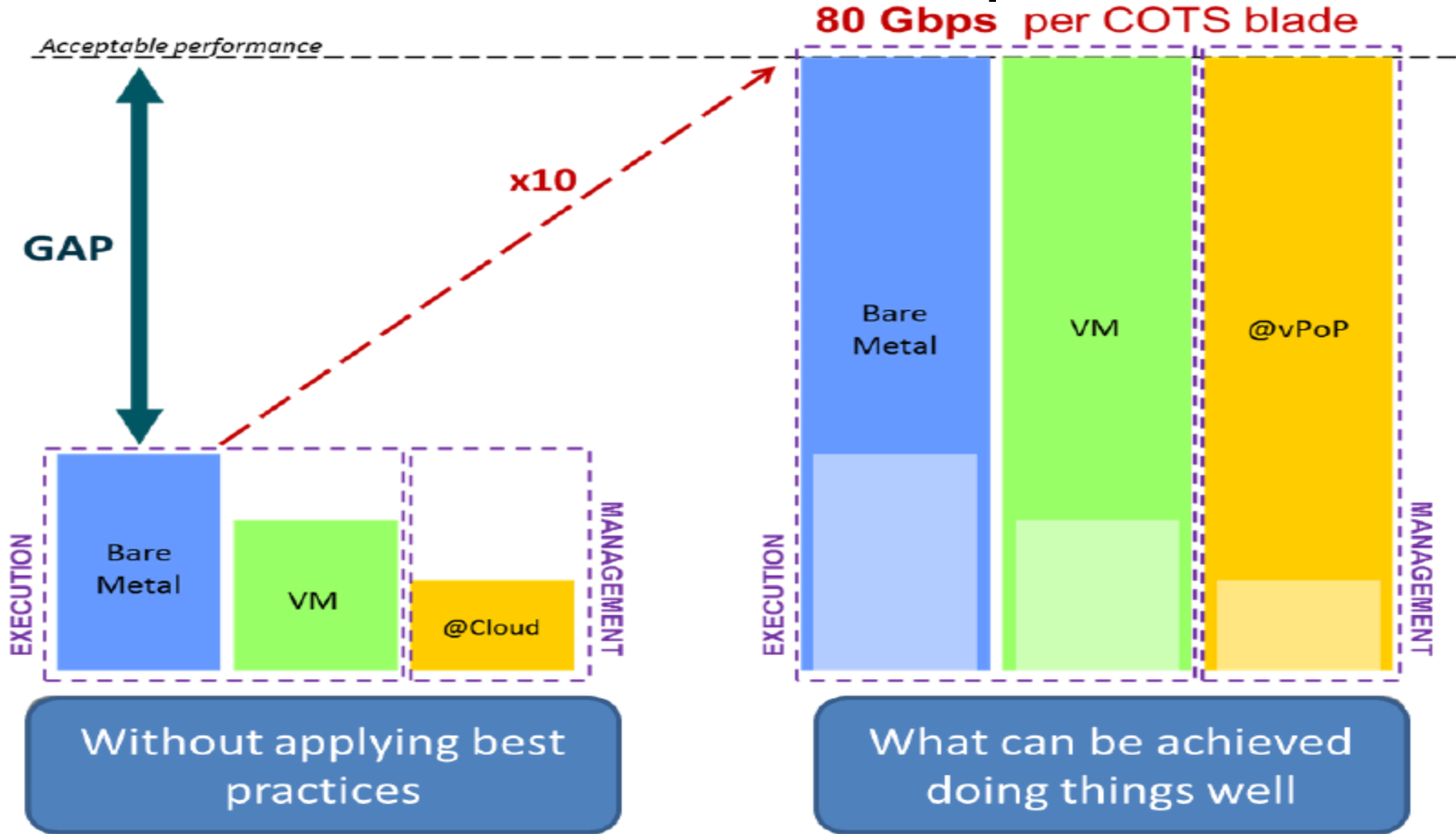


FDS(ODL based)



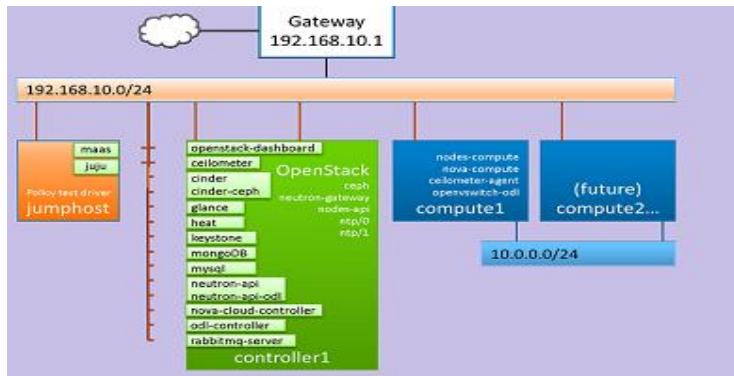
VPP over Linux FDIO/DPDK

OPNFV– Performance possibilities



Models & Policies

- **Model-Driven and Policy based NFV addresses automation in large OPNFV installations**
 - Use case driven.SFC, Clustering, L3VPN, Scaling, Policy /Intent Grouping
 - Based on Standards defined for modeling like UML, TOSCA,YANG with Templates and APIs
 - Testing in OPNFV Pharos PODs with Modeling tools



Yang tools

Tosca Library

Heat Translation Library

ETSI NSD, VNFD, VDU, VLINK, VN, NODES

VNFFG, SFC, GBP, NEMO

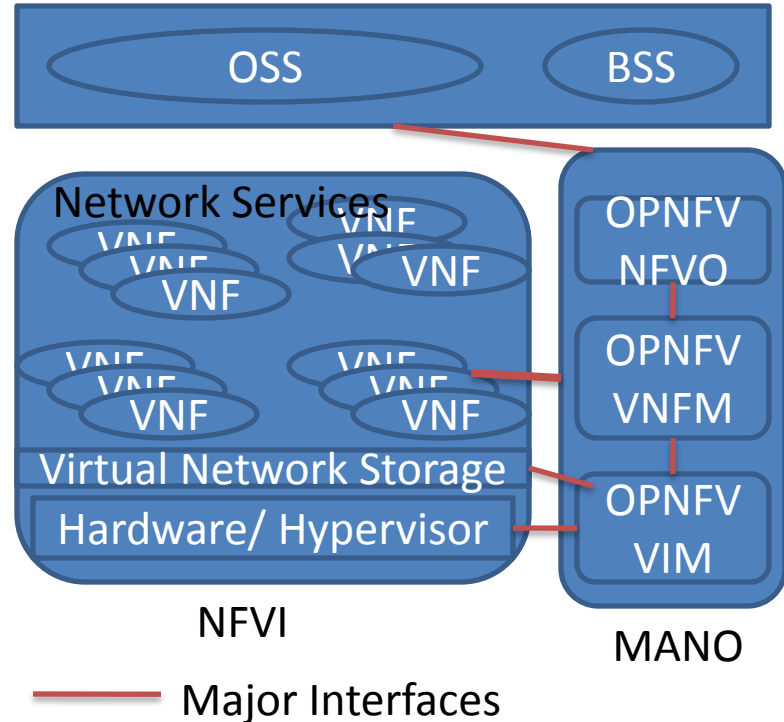
Virtual or Logical and Physical Models

Policies – Scale UP/DOWN, IN/OUT

Orchestration for OPNFV is evolving

- OPNFV Orchestration starts at OSS/BSS
- Follows to NFVO for Service Orchestration using VNF Life cycle management
- Finally lands on VIM for Resource Orchestration & Network Overlay including Virtualization
- MANO Options
 - OPEN-O, OSM, Tacker
 - Automation Tools (Puppet, Chef, Ansible, Salt, Domino...)

- Overview of Orchestration



OPNFV provides to the Ecosystem

- Preintegrated scenarios for deployment (installer, Linux-flavor, SDN, optional Features)
 - 21 Scenarios are in pipeline for „colorado“ release
 - Eg. Os-noSDN-nofeatures-ha, Os-onos-sfc-ha
- Testing framework-(Tempest, Rally, Robot)
- Community Labs (Huawei, Ericsson, Intel, Orange,CMCC and many more)
- Plugfests : Interoprability and Portability
- Compliance and Certification Program

Continuous Integration in OPNFV

- Automation (CI) / Octopus <https://build.opnfv.org/ci/>
- Automation Build/Release :Jenkins workflow: Releng
- Liuxfoundation LF Lab : Master Jenkins
- Community Labs: Slave Jenkins & Jumpserver + PODs

The screenshot shows the Jenkins web interface. At the top, there's a search bar and a 'log in' button. Below the navigation menu, a warning message states: 'JOBS IN THIS VIEW ARE USED FOR OPNFV PLATFORM CI. ONLY JOBS RUNNING ON CI PODS SHALL BE INCLUDED IN THIS VIEW. LF POD1, ERICSSON POD2, INTEL PODS, HUAWEI SC POD1, LF POD2, INTEL POD2, INTEL PODS, INTEL PODS'. Below this, there are several tabs for different job categories. The 'OPNFV Platform CI' tab is selected, showing a table of build results for the 'stable/brahmaputra' branch.

S	W	Name	Last Success	Last Failure	Last Duration	Console
		apex-daily-brahmaputra	13 hr - #97	5 days 13 hr - #92	13 hr	
		apex-deploy-baremetal-ovs-odl-l2-nofeature	11 hr - #79	5 days 11 hr - #74	37 min	

Installers

Apex (RDO)

JOID (JuJu)

Fuel(OpenStack)

Compass (Ansible)

Scenarios

os-nosdn-nofeature-ha

os-onos-sfc-ha

os-odl_l2-sfc-ha

Compliance & Certification (Dovetail)

Work-in-Progress

- Certification/verification framework
- Capability like in RefStack
 - By Units (NFVI:Hypervisor),(VNF:L3VPN) etc.
 - By Modules (OpenStack: Nova, Neutron, Cinder)
 - By Levels (Core, Mandatory, Optional etc)
- Verification By OPNFV/Third Party/Vendors
 - Identify needs/requirements of telcos as documented in requirements projects
 - Assurance (interoperability/portability)



Q&A

For general info refer

<https://wiki.opnfv.org>

<http://www.opnfv.org>

Or Contact us through email

Ulrich.Kleber@huawei.com

Prakash.Ramchandran@huawei.com