We’re ready. Are you?
SDN and NFV for Service Providers
Rada Stanic, Consulting Solutions Architect
Agenda

• Standards and Open Source Landscape
  • Emergence of Containers and Microservices
• Cisco SDN and NFV Solutions
• NFVI
• Use Cases
• What’s Next for Programmability ?
• Summary and Conclusion
Software Defined Networking

- Decoupled Control and Data Planes
- Highly Centralised Control (aka SDN Controller)
- Greater application interaction with the network
- An opportunity to re-think the relationship between network hardware and software

**SDN Definition (ONF):** The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices.
Network Functions Virtualisation

- Transition of network services to run on virtualised compute infrastructure
- Decoupling the service from the device
- Elastic, consumption-based service model
Standards and Open Source Landscape
Evolving Network Stack – HW, OS, and Hypervisors

Industry Organisations

Application Software

Infrastructure Software (Orchestration & Control)

Network OS
(Control & Management Planes)

Custom Silicon
(Data Plane)

Custom Kernel
(Device OS)

Device OS
(VM or LXC)

Linux
(Host OS)

Virtual Forwarder
(Optional, SDN Overlay)

Network OS
(Control & Management Planes)

Merchant Silicon
(Data Plane)

Hypervisor or Linux Container

Linux
(Host OS)

Hypervisor or Linux Container

Linux
(Host OS)

Compute Server
(Cloud Platform)
Evolving Network Stack – HW, OS, and Hypervisors

- Custom Silicon (Data Plane)
- Merchant Silicon (Data Plane)
- x86 Compute (Cloud Platform)
- Linux (Host OS)
- Hypervisor or Linux Container
- Network OS (Control & Management Planes)
- KVM
- Application Software
- Industry Organisations
- Open Config
- ONF
- ETSI
- OPNFV
- Infrastructure Software (Orchestration & Control)
- OpenStack
- OpenDaylight
- ONOS
- Protocols & APIs
- DPDK
- Open vSwitch
- Network OS
- Virtual Forwarder (Optional, SDN Overlay)
- Compute Server (Cloud Platform)
- Custom Kernel (Device OS)
- Device OS (VM or LXC)
- Linux (Host OS)
- Infrastructure Software (Orchestration & Control)
Open Networking Foundation (ONF)

- A user-driven organisation dedicated to the promotion and adoption of SDN through open standards.
- ONF emphasises an open, collaborative development process that is driven from the end-user perspective.
- Signature accomplishment to date is introducing the OpenFlow standard, which enables remote programming of the forwarding plane.

Source: https://www.opennetworking.org/
ETSI Network Function Virtualisation (NFV)

- In November 2012 seven leading telecom network operators selected ETSI to be the home of the Industry Specification Group (ISG) for NFV.
- Specifications focus on movement of Network functions to the cloud
  - Control, services and data plane components
- NFV is an architecture rather than simply virtualising functions
  - Virtual services, compute
  - Service chaining, overlays
  - Orchestration and redirection

See also: [http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf)
OpenStack
Open Source Software for Creating Private and Public Clouds

Your Application

OpenStack Dashboard

Compute (Nova)
Self-service provisioning of virtual machines through a software API

Network Service (Neutron)
For tenant created, virtual isolated networks and subnets, and services

Object Storage (Swift)
Massively scalable, distributed object store

www.openstack.org
Open Daylight

- Open Daylight is an open platform for network programmability to enable SDN and NFV for networks at any size and scale.

- Open Daylight software is a combination of components including a fully pluggable controller, interfaces, protocol plug-ins and applications.

www.opendaylight.org
DPDK: Data Plane Development Kit

- DPDK is an Open Source BSD licensed project.
- DPDK is a set of libraries and drivers for fast packet processing.
- It was designed to run on any processors knowing Intel x86 has been the first CPU to be supported.
- DPDK is not a networking stack and does not provide functions such as Layer-3 forwarding, IPsec, firewalling, etc.

Sources: [www.dpdk.org](http://www.dpdk.org), DPDK Summit (Sept. 2014)
Open Platform for NFV (OPNFV)

- Project originated out of Linux Foundation by a mix of vendors and operators to facilitate industry adoption of NFV through an Open Platform
- OPNFV will create a platform for NFV
- OPNFV will collaborate with existing upstream projects and initiatives such as open source and standards organisations
- Focus is on integration, closing of gaps, and testing with initial focus on Virtualisation, Controller and VIM layers of the platform

www.opnfv.org
Open Compute Project (OCP)

Goal: disaggregate Hardware and Software; create greater customer choice for each

• Applies to racks, servers, storage, switches, etc.
• Announced OS-agnostic ToR (Wedge), and modular chassis (6-Pack)
• Numerous hardware and software contributions from the vendor community

www.opencompute.org
Segment Routing

- IP/MPLS architecture that seeks the right balance between distributed intelligence and centralised optimisation and programming
- Drastic reduction of control-plane and hardware state
- Better utilisation of the installed infrastructure
- Wide applicability: DC, WAN, Metro, Peering (end-to-end)
- An architecture designed with SDN in mind
- Unleash application-network innovation
- Open IETF proposed standard (SPRING working group)

www.segment-routing.net
Segment Routing

- Distributed routing protocol used to compute shortest (or best) paths and advertise segments
- Segments identify forwarding resources within the topology and are encoded as labels
  - Global segments: nodes / prefixes
  - Local segments: peers, output interfaces
- Central SDN controller chooses explicit paths for flows and programs source (border router, VM, application) with forwarding policies (i.e., match flow → push segments / label stack)
- Downstream nodes switch based on label stack without carrying any per-flow state (reuses MPLS data plane)
- Implementations: IOS XR, IOS XE, NX-OS, WAE
NETCONF (or REST), YANG, and Open Config

Taking Service Automation to the next level: Declarative Configuration and Model-Driven Management

What is OpenConfig?

"An informal working group of network operators sharing the goal of moving our networks toward a more dynamic, programmable infrastructure...declarative configuration and model-driven management and operations. The initial focus of the effort is on the development of vendor-neutral data models for configuration and management that will be supported natively on networking hardware and software platforms."

source: www.openconfig.net

Programmatic Interfaces
Control Plane / Agent
Operating System
Data Plane

Implementations: IOS-XR 5.3.1, IOS-XE 3.17, NX-OS (XML), Open Daylight, Tail-F NSO
Path Computation Element Protocol (PCEP)

- Used between head-end router and PCE to:
  - Request/receive path from PCE subject to constraints
  - State synchronisation between PCE and router
  - Hybrid CSPF

- Two current modes based on Stateful PCE initiative:
  - PCE Initiated: App + PCE initiate tunnel setup
  - LSP Delegation: router initiates tunnel setup (e.g. via CLI or NMS) then delegates tunnel management to PCE

- Implementations
  - IOS XR 5.1.1, WAE, Open Daylight
BGP Flow Specification (aka Flowspec)

- Base specification defined in IETF RFC 5575
  - Various extensions defined in other IETF documents (see IDR working group docs)

- Provides the following key capabilities:
  1. Distribute ACLs via BGP, thereby, enabling rapid inter-domain distribution of flow-based traffic filters at large-scale (network wide)
  2. Flow-based traffic redirection, for example, to traffic scrubber for DDoS mitigation

- Open Daylight Lithium release will support origination of BGP Flowspec rules
  - Recent Cisco contribution
  - Enables centralised policy engine to dynamically program network wide traffic filtering and steering policies via Open Daylight SDN controller REST interface
  - Facilitates SDN-based DDoS mitigation

- Implementations: IOS XR 5.2.0, IOS XE 3.15S, Open Daylight (Lithium)
BGP Link State (BGP-LS)

- Allows BGP to push IGP topology (LSDB) and resource utilisation up to central SDN controller
  - New link state address family
- BGP provides a familiar operational model to aggregate topology information across domains
  - Multi-hop sessions
  - Need at minimum single BGP-LS speaker per domain
- Topology information distributed from IGP into BGP (only if changed)
- Implementations
  - IOS XR 5.1.1, WAE, Open Daylight
Emergence of Containers and Microservices
Cloud Native

An application designed to run in a cloud computing environment

• infrastructure agnostic – application may have resource and service requirements but it doesn’t care about the specific underlying hardware

• application components are designed as relatively simple, discoverable, re-usable services – e.g. microservices

• designed to survive failures

• designed for horizontally scaling up or down
# New Application Architectures

<table>
<thead>
<tr>
<th>Monolithic Apps</th>
<th>Cloud Apps</th>
</tr>
</thead>
<tbody>
<tr>
<td>server / hypervisor, IaaS</td>
<td>server clusters, containers</td>
</tr>
<tr>
<td>difficult to scale</td>
<td>easy to scale</td>
</tr>
<tr>
<td>high impact to component failure</td>
<td>built for failure, system resilience</td>
</tr>
<tr>
<td>challenging to upgrade</td>
<td>easy to upgrade</td>
</tr>
<tr>
<td>larger dev and ops teams</td>
<td>smaller, agile devops teams</td>
</tr>
</tbody>
</table>

- **Monolithic Apps**: These applications are typically built as a single unit and are deployed on a single server or hypervisor. They are difficult to scale, have high impact to component failure, are challenging to upgrade, and require larger development and operations teams.

- **Cloud Apps**: These applications are deployed in a cloud environment, often using server clusters and containers. They are easy to scale, built for failure, and system resilience, are easy to upgrade, and support smaller, agile devops teams.
Microservices

"The microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery."

5 Architectural Constraints of Microservices

1. Elastic – be able to scale, up or down, independently of other services in the same application.
2. Resilient – fail without impacting other services in the same application.
3. Composable – offer an interface that is uniform and is designed to support service composition
4. Minimal, and – only contain highly cohesive entities
5. Complete – be functionally complete

“Disruptor: Continuous Delivery with Containerised Microservices” – Adrian Cockcroft

Containers are Almost Like Virtual Machines

- Containers have their own network interface (and IP address)
  - Can be bridged, routed... just like with Xen, KVM etc.

- Containers have their own file system
  - For example a Debian host can run Fedora container (and vice-versa)

- Security: Containers are isolated from each other
  - Two containers can't harm (or even see) each other

- Resource Control: Containers are isolated and can have dedicated resources
  - Soft & hard quotas for RAM, CPU, I/O...
Hypervisors vs. Containers

Containers are isolated, but share OS and, where appropriate, libs / bins.

Type 1 Hypervisor
Hypervisor
Operating System
Hardware

Type 2 Hypervisor
App
Bins / libs
Operating System
Virtual Machine

Linux Containers
App
Bins / libs
Container
Hardware
Cisco SDN and NFV Solutions
Cisco SDN Strategy for SPs

- Management Plane: Programmable Platforms and Network Operating Systems
- Control Plane: Distributed Intelligence with Centralised Control
- Data Plane: Custom, Merchant, and Virtualised portfolio, MPLS forwarding
- Orchestration: Multi-domain and Multi-layer, and Model driven
  - E2E service lifecycle, and customer experience focus
  - Seamless integration with existing and future OSS/BSS environment
  - Modular architecture leveraging open APIs and standard protocols
- Commitment to Open Standards and Open Source

Multi Vendor End to End Management & Orchestration (Physical & Virtual)
Cisco SDN Strategy for SPs

- Management Plane: Programmable Platforms and Network Operating Systems
- Control Plane: Distributed Intelligence with Centralised Control
- Data Plane: Custom, Merchant, and Virtualised portfolio, MPLS forwarding
- Orchestration: Multi-domain and Multi-layer, and Model driven
  - E2E service lifecycle, and customer experience focus
  - Seamless integration with existing and future OSS/BSS environment
  - Modular architecture leveraging open APIs and standard protocols
- Commitment to Open Standards and Open Source

- Programmable Cisco Routers, Switches, Optical, Servers and Virtual Network Functions

Multi Vendor End to End Management & Orchestration (Physical & Virtual)
Cisco Routing and Switching Technology Options

- Data plane forwarding technologies include: Cisco custom silicon, merchant silicon and software-based forwarders
  - Each offers different levels of programmability, scale, performance, power consumption and unit cost
- Cisco invests in all data plane technology options to support different networking needs:
  - Example Cisco custom silicon –based products include but are not limited to:
    - NCS 6000, ASR 9000, CRS, Nexus 9000 (ACI mode) and Nexus 7000 series
  - Example Cisco merchant silicon –based products include but are not limited to:
    - Nexus 9500 (stand-alone mode) and Nexus 3000 series
  - Example Cisco virtual network function (VNF) products include but are not limited to:
    - IOS XRv 9000, CSR 1000v, ASA v etc…
Cisco Cloud Services Router (CSR) 1000V
Cisco IOS Software in Virtual Form-Factor

**Enterprise-class Networking with Rapid Deployment and Flexibility**

- **IOS XE Cloud Edition**
  - Selected features of IOS XE based on targeted use cases

- **Infrastructure Agnostic**
  - Not tied to any server or vSwitch, supports ESXi, KVM, Xen, AMI

- **Throughput Elasticity**
  - Delivers 10 Mbps to 10 Gbps throughput, consumes 1 to 8 vCPU

- **Multiple Licensing Models**
  - Term, Perpetual, Hourly

- **Programmability**
  - RESTful APIs for automated management
**IOS XRv 9000**

- **Virtualised ASR 9000 router including:**
  - 64-bit Linux kernel with KVM and Container based virtualisation for control plane
  - High performance, feature rich data plane based on x86 optimised code base

- **20Gbps+ Forwarder with features for IMIX traffic (with 8 core socket)**
  - i.e. 2×10GE ports at line rate
  - Multi-core scale-out for feature performance
  - Multi-socket scale out for control plane
  - x86-optimised emulated HW assists (QOS traffic manager, SW TCAM, PLU, Packet Replication)

- **Available since July 2015**
  - Hypervisor support includes Red Hat KVM, Ubuntu KVM and VMware ESXi (more to follow)
  - Operates as single VM → Linux containers used for data, control and admin planes
  - VM creation and deployment: OpenStack, VMware vCenter and VMware vCloud Director
Network Services Orchestrator (NSO)

• Enabled by tail-f

• Multi-vendor service orchestrator for existing and future networks
  o Includes distributed (multi-device) service configuration management, transaction integrity, validation and rollback

• Single pane of glass for:
  o L2-L7 networking
  o Hardware Devices
  o Virtual Appliances

• YANG Model Driven Orchestration
  o Service Data models (declarative)
  o Device Data Model (for auto config)
  o Fastmap engine translates models to device configuration including CLI

• Highly Scalable for large infrastructure
  o One of the existing deployment is managing 60K devices on the network
Virtual Topology System (VTS)

Management & Orchestration Plane

- DC Overlay SDN system consisting of:
  - Virtual Topology Controller (VTC)
  - Virtual Topology Forwarder (VTF)
  - VTF is highly optimised forwarding for x86
  - Guest OS as opposed to Host OS
  - VXLAN overlays
  - Service chaining

Data Plane

- Bare Metal Workload
- Virtualised Workloads with OVS
- Virtualised Workloads with Feature Rich & High Performance Cisco VTF Solution
- Virtualised Workloads with dVS
- Virtualised Workloads with SR-IOV
Virtual Topology System (VTS)

Management & Orchestration Plane
- VTS GUI
- 3rd Party VM Manager
- Elastic Services Controller (ESC)
- VCenter

Control Plane
- Virtual Topology System (VTS)
- MP-BGP
- BGP-EVPN
- RESTCONF/YANG
- WAN / Internet

Data Plane
- IP / MPLS WAN
- Bare Metal Workload
- Virtualised Workloads with OVS
- Virtualised Workloads with Feature Rich & High Performance Cisco VTF Solution
- Virtualised Workloads with dVS
- Virtualised Workloads with SR-IOV
Elastic Services Controller (ESC)
VNF Lifecycle Management, Monitoring and Elasticity

List of Events
- VM Alive
- Service Alive
- Upper load threshold crossed
- Lower load threshold crossed
- Service Dead
- VM Dead

List of Actions
- Notify (callback)
- Advertise Service
- Withdraw Service
- Restart VM
- Scale up (add a VM)
- Scale down (remove a VM)
- Individually customisable action(s) for every event

Simple Rules
- Service Alive => advertise
- VM Dead => withdraw
- Upper load => scale up

Complex Rules
- Service Alive => Advertise, Notify
- Upper load => Scale up, Notify, Advertise
- Service Dead => Withdraw, Notify, Restart
Cisco NFV Architecture Mapped to ETSI NFV Framework

- Modular architecture that conforms to ETSI NFV framework
- Model driven design for declarative NFV orchestration
- Supports Cisco and 3rd party VNF Managers
- Supports Cisco and 3rd party DC SDN Controllers
- Supports Cisco and 3rd party VNFs
Cisco’s Open SDN Controller
Cisco’s Commercial Edition Of Open Daylight

Pre-Installed Apps

- **BGPLS Manager** – visualises network topology from BGP database
- **Inventory** – augmented OpenDaylight “Nodes” app identifies all connected devices
- **(YANG) Model Explorer** – exposes system models and previews JSON API body
- **OpenFlow Manager** – manages, visualises and troubleshoots flows + previews JSON API body
- **PCEP Manager** – creates, modifies and deletes MPLS LSPs


Centralised OA&M

- Robust user, application and feature administration
- Status monitoring: system, cluster, node
- Event logging
- Real-time CPU, memory, disk, heap size, load and network utilisation metrics

“One-Click” Install

- VMware ESXi and Oracle Virtual Box hypervisor ready
WAE: WAN Automation Engine

- Multi-Application Engine for the SP WAN
  - Enables applications to make queries about placing demands
  - Run demand placement and network failure simulations
  - Request demand placement or optimisation on the network

- WAE is really an advanced suite of network optimisation, planning and calendaring capabilities which can be leveraged by applications
  - Capabilities exposed via northbound REST/Java/Thrift APIs

- WAE uses topology and traffic abstraction
  - By collecting information from the network

- Multi-Vendor platform

- Compliments NSO and Open SDN Controller (ODL)

Traffic Management Applications

Traffic Management Applications

REST

REST/NETCONF

SDN Orchestration & Control

Configlet  NSO  ODL/OSC  EMS/NMS

NETCONF/PCEP/BGP-LS

Source

Destination

http://www.cisco.com/go/wae
NFVI
Cisco NFV Architecture

Virtual Network Functions
Cisco and 3rd Party

- CSR
- ASA
- vNAM
- vIPS
- vPC-DI
- vIMS
- Video Opt.
- 3rd party

GUI
Zenoss
Single Pane of Glass Mgmt. & Infrastructure Assurance

UCSD

Virtual Infrastructure Manager

- Mercury based on RHel OSP 7
- OpenStack

Network VIM
- APIC
- VTS
- OSC

Physical Infrastructure
Cisco & 3rd Party

- Compute (UCS)
- Network (Switches /Routers)
- Storage
- Ceph

NFV-O & Resource Orchestration
NSO – Network Services Orchestrator enabled by Tail-f

VNF Manager
Cisco ESC
3rd Party

OSS/BSS
Service Catalog

Service Catalog

NFVI Scope

GUI
Zenoss
UCSD
Virtual Infrastructure Manager
Network VIM
Physical Infrastructure

API
Distributed NFVI in Cisco’s Open Network Architecture

Enabling End-to-End Service Creation

<table>
<thead>
<tr>
<th>Apps</th>
<th>vMS</th>
<th>SP Mobility</th>
<th>Media Cloud</th>
<th>Cloud</th>
<th>Generic NFV</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ESP</th>
<th>Catalog</th>
<th>Orchestration</th>
<th>Controllers</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>EPN</th>
<th>Internet</th>
<th>Aggregation</th>
<th>Edge</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Carrier-E / Transport</th>
<th>Edge</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Data Centres</th>
<th>DC</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SP IP or MPLS Core</th>
<th>Cloud</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Public Cloud</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CCS/InterCloud</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Access</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Generic NFV</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Media Cloud</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Cloud</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SP Mobility</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Catalog</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Orchestration</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Controllers</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Apps</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>vMS</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SP Mobility</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Media Cloud</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Cloud</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Generic NFV</th>
</tr>
</thead>
</table>

| Cisco Confidential |

| Distributed NFVI in Cisco’s Open Network Architecture |

| Enabling End-to-End Service Creation |

<table>
<thead>
<tr>
<th>Apps</th>
<th>vMS</th>
<th>SP Mobility</th>
<th>Media Cloud</th>
<th>Cloud</th>
<th>Generic NFV</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ESP</th>
<th>Catalog</th>
<th>Orchestration</th>
<th>Controllers</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>EPN</th>
<th>Internet</th>
<th>Aggregation</th>
<th>Edge</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Carrier-E / Transport</th>
<th>Edge</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Data Centres</th>
<th>DC</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SP IP or MPLS Core</th>
<th>Cloud</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Public Cloud</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CCS/InterCloud</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Access</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Generic NFV</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Media Cloud</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Cloud</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SP Mobility</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Catalog</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Orchestration</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Controllers</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Apps</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>vMS</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SP Mobility</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Media Cloud</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Cloud</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Generic NFV</th>
</tr>
</thead>
</table>

| Cisco Confidential |
NFVI POD And Value to Service Provider

- Modular Configuration
- Multiple Sizes and Expandable
- Single Click Automated Install
- Easy to Manage and Operate
- Pre-Integrated and Validated
- Packaged NFVI

Starter Kit (POC / Lab)
½ Rack POD (Production)
Full Rack POD (Production)
Compute Expansion Module
Storage Expansion Module

CiscoLive!
## UCSD Single Plane of Glass Management (SPM) Feature Summary

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero Touch / Bare Metal Provisioning</td>
<td>SPM to provision Blades and install KVM and add the node in OpenStack &amp; VMware</td>
</tr>
<tr>
<td>GUI for system install</td>
<td>GUI for bare metal system install and OpenStack install</td>
</tr>
<tr>
<td>OpenStack support</td>
<td>OpenStack management added to UCSD</td>
</tr>
<tr>
<td>On Boarding of VNF</td>
<td>Complete onboarding of a VNF from SPM on OpenStack</td>
</tr>
<tr>
<td>Resource Management</td>
<td>Visualise resource groups created in OpenStack via SPM and provision VM’s in those resource groups, at the same time need the ability to see the utilisation of those resource groups. Ability to reconfigure resource groups from SPM.</td>
</tr>
<tr>
<td>Identity Management, Multi tenancy</td>
<td>Hierarchy in tenancy for resource allocation and access control, with micro segmentation for VM’s at TOR or VTF layer</td>
</tr>
<tr>
<td>Zenoss integration for Infrastructure monitoring and fault</td>
<td>Monitor the VM as well as underlying hardware components, and based on an event spin off a new VM on either same or different blade. The fault management and alerting should be embedded in SPM and cant be a separate GUI</td>
</tr>
<tr>
<td>Reporting for Capacity Management</td>
<td>Measuring Capacity for NFVI POD and provide metrics and reporting for capacity planning</td>
</tr>
<tr>
<td>Charge Back</td>
<td>Chargeback based on SLA and SLG for allocated resources</td>
</tr>
</tbody>
</table>
Mercury is a Cisco OpenStack platform built on top of an underlying OpenStack Distribution like Red Hat to build a carrier grade platform integrated with Cisco HW & SW.

Mercury provides a set of tools including an automated installer, containerised OpenStack services, logging/monitoring, health check tools and plugins for Cisco HW and SDN controllers.

Goal of Mercury is to provide a reliable, highly available & easily upgradeable OpenStack platform for SP deployment.
Use Cases
Use Case #1 – Service Automation (1/2)
End-to-End Service Provisioning Across Multi-Vendor Network

1. Customer adding new site to L3VPN
2. Portal or Partner REST API call to Tail-F NSO with new service parameters
3. NSO performs customer lookup and calls L3VPN service model
4. Service model contains multi-device logic which includes provisioning of backhaul circuit(s) and PE configuration
5. Service model parameters are mapped to the appropriate NSO Network Element Driver (NED) for each device in the service topology
6. NSO’ NED pushes configuration to devices via CLI or NETCONF/YANG
Use Case #1 – Service Automation (2/2)
End-to-End Service Provisioning Across Multi-Vendor Network

Benefits
- Massive reduction in provisioning errors
- Customer self-selection (and automation)
- Developed new Flexible-VPN service offerings
- 10x increase in transactions vs. previous manual process
- Scale Ops Staff: increased services-to-engineers ratio
Use Case #2 – Automated Traffic Engineering
Improve Capacity Management and Network Utilisation

1. WAE and ODL continuously collect topology and traffic information
2. PCE application queries WAE for topology and traffic info
3. PCE application computes optimal load placement for traffic matrix
4. PCE application programs re-optimised load placement via WAE
5. WAE creates, modifies and/or deletes RSVP-TE tunnels or SR-TE paths as needed via ODL
6. Go back to step 2 and repeat → Automated Traffic Engineering
Use Case #3 – Bandwidth on Demand / Calendaring
New Service and Revenue Opportunity

1. WAE and ODL continuously collect topology and traffic information
2. Customer uses SP web portal to reserve bandwidth between its sites (effective immediately or at a future date and time)
3. PCE application queries WAE to verify if traffic demand can be admitted and, if so, to identify optimal paths
4. PCE application decides load placement for traffic demand
5. PCE application admits traffic demand via WAE
6. WAE admits traffic demand onto network topology which may include creation, modification and/or deletion of RSVP-TE tunnels or SR-TE paths as needed via ODL
Use Case #4 – Application Engineered Routing
Allows Applications To Get What They Need From The Network

1. SDN controllers are capable of collecting data from the network – topology, link states, link utilisation, … (ODL + WAE + PCE app)
2. Application expresses demand requirements (bandwidth, latency, live-live, SLA) to SDN controller
3. SDN controller programs forwarding policy for application while preserving network scale and simplicity (via Segment Routing)
4. Application traffic is mapped to path defined by a list of segments as determined by SDN controller
5. Application Engineered Routing enables application interaction with the network
Use Case #5 – Egress Peer Engineering (1/2)
BGP Segment Routing for Peering and Traffic Steering

1. Multiple egress paths to user 9.9.9.9 in AS4
2. Peering Routers X & Y configured for Segment Routing
3. X & Y advertise Prefix-SIDs for their loopback addresses
4. X & Y advertise PeerNode SIDs for each external Peer to controller with BGP-LS
5. BGP-LS has full understanding of fabric topology
Use Case #5 – Egress Peer Engineering (2/2)
BGP Segment Routing for Peering and Traffic Steering

6. Controller calculates path, selects EPE path to destination via Egress Node Y, AS2 Peer
7. Controller programs Ingress node (CDN or first hop router) with BGP-LU update
9. Controller monitors path: auto-update of label-stack during congestion or outage
10. Controller: full Internet table. Peering Routers: may carry full table or only local prefixes in case of direct peering. Fabric and DC-Edge: LSR only

Future: SR-TE – support for custom SR path through fabric to Peering Routers (aka, tunnel steering mechanisms and imposition of 3 or more labels)
Use Case #6 – Virtualised Managed Services (1/3)

Using NFV to Add New Revenue Generating Services

1. ETSI NFV Management Framework
2. Tail-F NSO maintains data-models for CFS (Customer Facing Services) and RFS (Resource Facing Services)
3. CFS request received from portal: add security service to L3VPN
4. CFS maps to RFS data-model: vFirewall with service attributes (BW, HA, monitoring, etc.)
5. NSO Orchestrator sends service request to VNF Manager (Elastic Services Controller)

6. ESC maps attributes (BW, HA, etc.) to VM image and resource requirements (CPU, memory) and sends build request to VIM (Openstack)

7. Openstack builds VM(s)
Use Case #6 – Virtualised Managed Services (3/3)
Using NFV to Add New Revenue Generating Services

8. NSO Orchestrator sends VNF network service request to VTS Controller
9. Network service request mapped to tenant VRF
10. VTS Controller provisions the following:
    - VNF attachment points to V-Forwarder
    - MPLS-over-GRE or VXLAN overlay tunnels to other V-Forwarders or to Data Centre PE Node
    - VRF config on Data Centre PE Node
Use Case #8 – DDoS Mitigation
Mitigate Security Attacks using SDN Controller

1. Network telemetry provides visibility to detect DoS attacks
2. Once detected, DDoS mitigation application programs attack mitigation policy using REST interface of Open Daylight SDN controller
3. Open Daylight SDN controller originates BGP flowspec rules to either filter attack or redirect attack traffic to scrubbing centre
4. BGP flowspec rules are propagated to wider network (e.g., border routers) using BGP route reflector (physical or virtual)
5. Border routers install security policy to mitigate attack
What’s Next for Programmability?
Problem Statement

- Virtualisation / automation / orchestration has made real-time service provisioning possible
- Open source big data / analytics technologies now being widely applied outside of big web companies
- OSS architectures simply not keeping pace with the rest of industry
- No coherent industry direction on how OSS needs to change in the presence of these new technologies
Relationship Between Orchestration and OSS Analytics

Orchestration is responsible for service provisioning and pushes state to the infrastructure.

OSS analytics is responsible for collecting data from the infrastructure, monitoring and analysis.

The “C” in FCAPS

Orchestration

OSS Analytics

State

Data

Related as loosely coupled but tightly integrated systems

The “F_APS” in FCAPS
Relationship Between Orchestration and Analytics

Orchestration Platforms: NSO, ODL, Openstack…

Orchestration is responsible for service provisioning and pushes state to the infrastructure.

The “C” in FCAPS

OSS analytics is responsible for collecting data from the infrastructure, monitoring and analysis.

The “F_APS” in FCAPS

Related as loosely coupled but tightly integrated systems

Platform for OSS Analytics

User

Access & Aggregation

Core

Data Centre

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Public
PlAtform for Network Data Analytics

**Vision:**
- Simple, scalable, open big data / analytics platform
- Forms a generic big data analytics platform supporting different types of analytics applications for cloud based networks and services
  - Operational Intelligence, e.g. OSS
  - Business intelligence, e.g. BSS

---

**Context:**
- Inventory
- Topology
- Geography

**Data Sources**
- Customer-level data
  - Customer
  - QoE Monitoring
- Infrastructure and service-level data
  - Applications
  - Orchestration
  - Controllers
  - Devices
- Network Telemetry

**Producers:**
- Event Data
- Log Data
- Metric Data
- Network Telemetry

**Data Distribution**

**Data Store & Processing**
- Horizontally Scalable Data Platform Applications

**Consumers:**
- Data analysis Applications

---

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Public
• Principles
  • Decouple data aggregation (publishers) from data analysis (consumers) – allow any OSS app the potential to access any data source
  • Simple, scalable, open data distribution platform
    • Scale-out architecture with support for horizontal scale in all core components
    • Very highly available core platform
    • Low and predictable latency
  • Immutable dataset with minimal filtering/processing on ingress
  • Analytics based approach to analysis functions
  • Support for streaming apps, real-time queries and batch processing
Benefits

• An open system architecture

• Collect data once – allow any analysis application to mine any data source, leveraging the full value of the OSS dataset

• Extensible – add new OSS analysis functions quickly and seamlessly with minimum of development cost

• Leverage rapid innovation in Big Data analytics space
PlAtform for Network Data Analytics

Technologies:
- Open Source
- Proprietary

Data Sources: Network Telemetry, SNMP

Producers: Data aggregation
- Metric Data
- Event Data
- Log Data
- Network Telemetry

Context:
- Inventory
- Topology
- Geography

Data Distribution: Logstash

Data Store & Processing:
- Kafka
- Spark Streaming
- Hbase
- Impala
- HDFS
- Spark

Data Analysis:
- Fault Analysis
- Perf. Analysis
- ELK
- OpenSoc
- Capacity Analytics
- Inventory
- Billing (Mediation)
- Business Intelligence

Consumers: Data analysis Applications
- Inventory Topology Geography
- SNMP Monit, Collectd, Logstash, Ceilometer
- PMACCT
- Logstash
- Collectd
- Logstash
- Netflow

Technologies:
- Netflow
- Technology Stack: ELK, OpenSoc, Spark, Hbase, HDFS, Impala, Spark Streaming
Summary and Conclusion
Summary

- Lots of industry activity related SDN and NFV
- Cisco is investing in all SDN and NFV technology options
  - Including but not limited to virtual network functions, merchant hardware, compute servers, SDN controllers, programmatic interfaces, cross domain orchestration, northbound applications and open source projects
- Cisco offers many deployable SDN and NFV solutions for SPs today that …
  - Enable network optimisation, increased utilisation and operational efficiencies
  - Enable new service and revenue opportunities
Q & A
Complete Your Online Session Evaluation

Give us your feedback and receive a **Cisco 2016 T-Shirt** by completing the Overall Event Survey and 5 Session Evaluations.

- Directly from your mobile device on the Cisco Live Mobile App
- Visit any Cisco Live Internet Station located throughout the venue

**T-Shirts can be collected Friday 11 March at Registration**

---

**Learn online with Cisco Live!**
Visit us online after the conference for full access to session videos and presentations.
[www.CiscoLiveAPAC.com](http://www.CiscoLiveAPAC.com)
Thank you