LET’S BUILD TOMORROW TODAY
NFV Architecture & Orchestration for Cloud based Virtual Managed Services

Sunil Cherukuri
BRKSDN-2065
Agenda

- Introduction
- vMS Architecture
- vMS 1.0 Use-cases & Benefits
- vMS Orchestration Details
- Future Use-cases
- Conclusion
Session Objectives

• Introduction to Cisco Virtual Managed Services (vMS) solution
• vMS 1.0 Architecture and model-driven Orchestration details
• Current and future vMS use-cases

• Session is oriented towards Managed Security Services, and Service Providers. But the concepts, usecases, and orchestration platform can be utilized for other usecases, and for Enterprises

• Knowledge of DC/Cloud platforms, technologies and orchestration are assumed
  • This session does not provide deep details on individual products like Network Service Orchestrator (NSO), CSR1000V, ASAv etc
  • This session does not delve into YANG/Netconf, preliminary knowledge is assumed
  • This session does not provide details on OpenStack
  • Session will mostly focus on service abstractions and logical designs, not configurations
Acronyms

- vMS – Cisco Virtual Managed Services solution
- NFV – Network Functions Virtualization
- VNF – Virtualized Network Function
- NSO – Cisco Network Service Orchestrator (based on Tail-f)
- ESC – Cisco Elastic Services Controller
- CSR1000V – Cisco Cloud Services Router
- ASA\text{v} – Cisco virtual Adaptive Security Appliance
- WSA\text{v} – Cisco virtual Web Security Appliance
- FlexVPN – Cisco converged IPsec VPN technology
- GETVPN – Cisco Group Encrypted Transport VPN
- LIS\text{P} – Locator ID Separation Protocol
Introduction
Network Functions Virtualization

NfV = Transition of network infrastructure services to run on virtualized compute platforms – typically x86

• NfV Initiative
  • Initiative announced at “SDN and OpenFlow World Congress”, Darmstadt, Oct 2012

• Key Enabler: using cloud technology to support network functions
  • Hypervisor and cloud computing technology
  • Improving x86 h/w performance
  • Network industry standardizing on Ethernet
  • Network automation / orchestration

• Value Proposition
  • Reduction in CAPEX and OPEX
  • Faster service provisioning
  • Service agility

• Not technically SDN but may use SDN technology – APIs, Controllers

Extract from “Network Functions Virtualisation – Introductory White Paper”
Network Function Virtualization

• Similar to compute moving to cloud, networking is virtualizing and moving to cloud: **Enabled by SDN and NFV**

• Capturing this transition is a top priority for nearly all service providers: Lower CAPEX, OPEX, truck rolls, & agility; Portal-based sales to SMB

• Enterprises: Cloud traffic driving the need for hybrid WANs and new internet-based services from service providers

• Drivers: Service velocity, Agility, Cost reduction, PAYG, Efficient Resource usage, Elasticity, On-demand Self Service

• Virtualized functions can reside On-Prem, in the POP, in the Cloud, or a mix.
Next Generation Business Services
Delivering business outcomes based on an integrated SDN network/cloud platform

Cross Domain Orchestration

- Automation - open platform for integration with open source, VNFs, & applications
- Built for integrated network and cloud services
- End customer visibility and control

Virtual Network Functions & applications

Distributed or Centralized NFV
Cisco VNF’s

• Vast and comprehensive set of VNF’s.
• Sample list:
  • Networking – CSR1000V, XRv
  • Security – ASAv, WSAv, ESAv, vFirePOWER, vSCE, vISE
  • Services – NetScaler1000v, vNAM, vWaaS, CUCM
  • vSwitch – Nexus1000v, VPP/VTS
  • Controllers – vWLC, vMSE, APIC, APIC-EM, OSC/ODL, WAE
  • Management – NSO (Tail-f), ESC, PNSC, PSC
  • Etc…

* Covered in this session
Service Chaining

- **Service Function**: Networking function that provides connectivity service, security service etc.

- **Service Chain**: Multiple functions linked together to provide a service

- **How are services chained**:
  - Chaining of features within a device – IOS ingress/egress feature (ACL+QoS+NAT)
  - Chaining via VLAN-stitching and routing within a device – Cat6500 with VPN, FW modules
  - Chaining via Routing across devices – hop-by-hop services
  - Chaining via Routing across VNF’s – hop-by-hop services
  - Chaining via diversion mechanisms like WCCP, Tunnels, PBR
  - Chaining via encapsulation mechanisms like vPath (Nexus1000v based) and Network Service Header (NSH)
Chaining across Cat6500 Service Modules

Slide I made in 2005 !!!
Security Services Chaining With vPath
Intelligent traffic steering through multiple network services (slide from 2013)

Use Case: VSG & ASA 1000V

Cisco Nexus 1000V

Cisco vPath

1. Use Case: VSG & ASA 1000V

2. Cisco ASA 1000V

3. Cisco VSG

4. Cisco Nexus 1000V

5. VM
Service Function Chaining (SFC) & Network Service Header (NSH)

The Future

- SFC and NSH IETF Drafts
- Topology and Transport independent service chaining
- No hop-by-hop service/VLAN stitching
- NSH header consists of metadata and service path information
- NSH header added to packet, then encapsulated in outer transport header

- Service Classifier (SC) determines (based on Policy) which traffic needs which services
- Service Function (SF) is a physical or virtual device providing network functions
- Service Path is a forwarding path used to realize a service chain, composed of one or more Service Functions
- Service Function Forwarder (SFF) forwards to one or more Service Function nodes, based on the service path identifier, and the service forwarding table
- Metadata and context processing at each node, can trigger service forwarding changes
- Service Orchestrator programs the SC, SF, SFF
Overlay Networks

- Logical network built on top of another network. Overlay nodes connected via logical links

- Abstraction from physical network allows for change, mobility etc.

- Some form of Controller, Distribution, or Learning involved to discover the logical links and anchor points

- Some form of tunneling - encapsulation and header manipulation (MACinMAC, MACinIP, IPinIP)

- Some common overlays:
  - IPsec, GRE, VXLAN, OTV, LISP, MPLSoGRE, NVGRE, L2TPv3oIP etc.
vMS Solution
vMS = Virtualizing the Network Functions of Managed Services so they can operate in the Cloud (Public or Private)
Evolution of Managed Services – Premise to Cloud

**Cloud**
- Application Containers
- Applications from the Cloud
- Cisco Cloud
- Virtual Private Cloud
- Public Cloud

**Network**
- Network Functions from the Cloud
- Secure IP Overlays
- MPLS
- Layer 2 VPN
- Hybrid

**Premise**
- Network Functions
- Network Functions on the CPE
- L3 “classic” on premise
- L3 CPE + x86 on premise
- L3 Meraki Cloud Managed
- Simple L3 CPE
- vCPE on X86 on prem
- L2 NID
vMS 1.0 – First Step in a Journey

Cloud
- Cloud Application Containers
- Applications from the Cloud
  - Cisco Cloud
  - Virtual Private Cloud
  - Public Cloud
- Managed WAN
  - Site-to-Site VPN
- Managed Security
  - Firewall, RA VPN, Web Security

Network
- Secure IP Overlays
- MPLS
- Layer 2 VPN
- Intelligent/Hybrid

Premise
- Network Functions
- Network Functions on the CPE
  - ISR G2

Cisco live!
- L3 "classic" on premise
- L3 CPE + x86 on premise
- L3 Meraki Cloud Managed
- Simple L3 CPE
- vCPE on X86 on prem
- L2 NID
vMS Architecture
vMS Architecture Fundamentals
Model Driven Orchestration Approach

- Service Intent is done through Modeling Languages that Abstract out the “How” and “Where”
- The Service is looked at summarily across the implementation domains.
- The Orchestrator has both a Service and Device component. Each independent of the other. Answers the “How”.
- The Orchestrator is able to Instantiate a Service Across the derived Topology (Infrastructure). Answer the “Where”.

![Diagram of vMS Architecture Fundamentals](image-url)
vMS 1.0 Architecture

Customer Orders Service

Network Service Orchestrator (NSO)

Elastic Services Controller (ESC)

OpenStack

Establish VPN: IPSec tunnel, IP Overlay (L2TP, VXLAN, GRE, LISP)

CPE Shipped at Customer Site, connected & Powered ON

PnP Functionality Zero Touch Provisioning

SP’s OSS/BSS

PROVIDE DAY 1 CONFIGURATION

CPE

Tenant Portal

NETCONF/YANG

REST API

NC/YANG, RC/YANG

SP’s OSS/BSS

CPE

X86 Server

vSwitch

CSR1Kv

ASAv

vWSA

Internet Gateway

VNF Service chain

PnP server

Provides Day 1 Configuration

Customer Orders Service
vMS Architecture Components
Maps to ETSI MANO

Network Service Orchestrator (NSO) (VNF-O)
Elastic Service Controller (VNF-M)
OpenStack (VIM)

Service APIs
OSS/BSS
Operator Portal

Physical
Virtual N/W Functions
CSR
ASAv
WSAv

SSH
SSH
SSH

SDN Controller
OVS (DC Switching)
VTS (DC Overlay)

Physical

Customer Facing Services
Resource Facing Services

Infrastructure

Service Definition Layer
Example DC Infrastructure
Multi-Tenant Access
Zone Edge
Virtual Network
Tenant Web

Resource Facing Services

Infrastructure

Service APIs
OSS/BSS
Operator Portal

Network Service Orchestrator (NSO) (VNF-O)
Elastic Service Controller (VNF-M)
OpenStack (VIM)

Virtual N/W Functions
CSR
ASAv
WSAv

SSH
SSH
SSH

SDN Controller
OVS (DC Switching)
VTS (DC Overlay)

Physical

Customer Facing Services
Resource Facing Services
Operator Portal

Exposing Service Blueprints to the Operator

- The Orchestration process can be kicked off through an Operator Portal.
- Super-user for SP Admin, RBAC for End-User Admin
- The Portal is aware of different Service Blueprints that can be exposed to an operator.
- The values that are selected in the Service Selection process result in the subsequent API call into NSO.
- The portal has 2 Modules.
  - Front-End: Skinned to the Customer’s Requirements
  - Back-end: Modified to support the Service Blueprints that can be orchestrated.
Network Service Orchestrator (from Tail-F)

VNF Orchestrator

- Service Intent
- Service Intent
- Service Intent
- Rest/NetConf/Yang
- Transaction Database (CDB)
- Service Models written in Yang
  Abstract Service from underlying physical devices
- Service Manager Interprets
  Service Intent
  with Service Instantiation Rules and
  derives configuration deltas.
- Transactional Database Allows full
  CRUD capabilities to Services.
- Device Manager manages derived
  and validated configurations
  in a transaction manner towards derived
  infrastructure.
- Network Element Drivers Abstract the interfaces
  to the devices allowing 3rd party infrastructure to
  participate in Service Instantiation

Zero Touch Deployment (ZTD)
Maps the Service Intent to the Derived Device Topology. Known as “Fastmap”

Open Method for ZTD Access

Open PnP

Service Manager

Device Manager

Network Element Drivers

ISR x86 Virtual Domain Controller

Network Models written in Yang
Abstract Service from underlying physical devices

Service Manager Interprets
Service Intent with Service Instantiation Rules and
derives configuration deltas.

Transactional Database Allows full
CRUD capabilities to Services.

Device Manager manages derived
and validated configurations
in a transaction manner towards derived
infrastructure.

Network Element Drivers Abstract the interfaces
to the devices allowing 3rd party infrastructure to
participate in Service Instantiation
Network Service Orchestrator (Tail-f)

Multi-vendor service orchestrator for existing and future networks

- L2-L7 networking
- Hardware Devices
- Virtual Appliances
- OpenFlow Switches

NSO provides abstractions based on:

- Data models
- Transactions

Yang/Netconf

- Run-time service reconfiguration
- Dynamic Reconfigurations
- Protocol-based (NETCONF)
- Client – server
- Fine-grained manipulations and data-model for the application
- Yang modeling allows for flexible customizations
Network Services Orchestrator
Model Driven Multi-Vendor Service Orchestration

Agility: Model-Driven Operations: Service Transactions

FASTMAP*
- CREATE SERVICE
- UPDATE SERVICE
- DELETE SERVICE
- REDEPLOY SERVICE

New Service Type: 2-4 days
New Device Type: 2-4 weeks
Elastic Service Controller
VNF Manager

Programmable Interface to ESC allows Functional Interaction to ESC Subcomponents.

VM Provisioning & Configuration Module

OpenStack
Public Clouds

VNS Bring-up & Initial Configuration Application. Multi-vendor Support.

ESC uses multidimensional approach to VNF Monitoring/Restartability

Allows Modular Communication with NSO. Data Model Driven.

Affinity Rules and Scale Requirements for the VNF components. Also manages the startup sequences.

Programmable Interface to ESC allows Functional Interaction to ESC Subcomponents.

VM Provisioning & Configuration Module

OpenStack
Public Clouds

VNS Bring-up & Initial Configuration Application. Multi-vendor Support.

ESC uses multidimensional approach to VNF Monitoring/Restartability

Allows Modular Communication with NSO. Data Model Driven.

Affinity Rules and Scale Requirements for the VNF components. Also manages the startup sequences.

Programmable Interface to ESC allows Functional Interaction to ESC Subcomponents.

VM Provisioning & Configuration Module

OpenStack
Public Clouds

VNS Bring-up & Initial Configuration Application. Multi-vendor Support.

ESC uses multidimensional approach to VNF Monitoring/Restartability

Allows Modular Communication with NSO. Data Model Driven.

Affinity Rules and Scale Requirements for the VNF components. Also manages the startup sequences.
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 customizable 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
OpenStack, OVS, and SDN Controller

Virtual Infrastructure Manager

- Open SDN Controller (OSC) is Cisco distribution of Open Daylight Controller (ODL)
- OVS will be supported by OSC in upcoming vMS release. Common Neutron Plugin gives upgrade path on SDN Controller
- VirtIO drivers for OVS. DPDK vhost-user drivers for Cisco Virtual Topology Forwarder (VTF)
Maas and Juju are the managing tools for Openstack from Canonical, running on Ubuntu.
vMS Pod
DC Underlay

Production Nodes
- Small Pod = 4
- Medium Pod = 16
- Large Pod = 32

Control Nodes (3)
- OpenStack Nodes (Nova, Neutron, Ceph Mon)
- vMS Management Nodes (Portal, NSO, ESC)

Service Chains (CSR, ASAv, WSAv)
Cloud Services Router (CSR 1000V)

- **IOS-XE code**
  - Comprehensive feature set
  - 4 month release cycle

- **Infrastructure Agnostic**
  - Cisco UCS, Dell, HP, etc.
  - Runs on vSwitch, dVS, N1KV, etc. – no dependency
  - VMware ESXi 5.5, Citrix Xen Server 6.1, KVM – Ubuntu 12.04 LTS, RHEL, RHEV, MSFT Hyper-V 2012 R2 support
  - Amazon AMI support

- **Footprint**
  - 1, 2, 4, 8 vCPU
  - 2.5 GB / 1 vCPU [default]. 4, 8 vCPU / 4 GB
Cloud Ready Router

- Extending Enterprise WAN to Cloud

**Enterprise**

**Security**
- IPSec VPN, L2TP
- Route-based VPNs (DMVPN, ..)
- Firewall, ACL, AAA

**Integration**
- NAT, LISP, OTV
- HSRP, QoS, AVC
- IOS-XE CLI, Cisco PNSC

**User Experience**
- Routing (BGP, EIGRP, Multicast, ..)
- AppNav, WCCP, QoS, AVC
- IP SLA, NetFlow, SNMP

**Cloud Provider**

**Tenant Scalability**
- 802.1Q VLAN, VXLAN
- Routing (BGP, GRE, VRF-Lite)
- MPLS (MPLS VPN, VRF)

**Enterprise Services**
- DMVPN, EasyVPN, FlexVPN
- Firewall, NAT, WCCP, QoS, AVC
- IP SLA, NetFlow, SNMP

**Tenant Manageability**
- 10 Mbps to 1 Gbps
- RESTful APIs
- Term and Usage Licensing
## CSR 1000V Features per Technology Package

<table>
<thead>
<tr>
<th>Technology Package</th>
<th>Virtualization</th>
<th>IOS-XE Features</th>
</tr>
</thead>
</table>
| **STANDARD** (Routing) | - ESXi 5.1/5.5  
- XenServer 6.1  
- KVM (Ubuntu 12.04 LTS, RHEV 3.1, RHEL 6.3)  
- Hyper-V 2012 R2 | - **Basic Networking**: BGP, OSPF, EIGRP, RIP, ISIS, IPv6, GRE, VRF-LITE, NTP  
- **High Availability**: HSRP, VRRP, GLBP  
- **Addressing**: 802.1Q VLAN, EVC, NAT, DHCP, DNS  
- **Basic Security**: ACL, AAA, RADIUS, TACACS+  
- **Management**: IOS-XE CLI, SSH, Flexible NetFlow, SNMP, EEM, NETCONF |
| **ADVANCED** (Standard + Security) | | - **STANDARD +**  
- **QoS**  
- **Multicast**: IGMP, PIM  
- **Advanced Security**: Zone Based Firewall, IPSec VPN, EZVPN, DMVPN, FlexVPN |
| **PREMIUM** (Advanced + AppX + Hybrid Cloud) | | - **ADVANCED +**  
- **Advanced Networking**: L2TPv3, BFD, MPLS, VRF, VXLAN  
- **Application Experience**: WCCPv2, APPNAV, NBAR2, AVC, IP SLA  
- **Hybrid Cloud Connectivity**: LISP, OTV, VPLS, EoMPLS |

**Differentiated Tenant Services** – Basic RaaS, Adv Security, Premium DCI, Visibility

---

*Cisco live!*
Cisco ASAv Firewall and Management Features

Cisco® ASA 9 Feature Set

- 10 vNIC interfaces and VLAN tagging
- Virtualization displaces multiple-context and clustering
- Parity with all other Cisco ASA platform features

- SDN (Cisco APIC) and traditional (Cisco ASDM and CSM) management tools
- Dynamic routing includes OSPF, EIGRP, and BGP
- IPv6 inspection support, NAT66, and NAT46/NAT64
- REST API for programmed configuration and monitoring
- Cisco TrustSec® PEP with SGT-based ACLs
- Zone-based firewall
- Equal-Cost Multipath
- Failover Active/Standby HA model

Removed clustering and multiple-context mode
Cisco ASAv Platforms

* Lab Edition license is built in with 100-Kbps throughput and 100 total connections allowed.
vMS 1.0 Use-cases
vMS 1.0 Use Case

Per-Tenant Services
- VPN termination for Hub-Spoke, or for Service Chaining
- Firewall for Internet/Extranet Services
- Additional Services like Web Security (others like IPS etc in future)
- UTM Service chain is currently per-tenant
vMS 1.0 Service Features

- Self-Service Portal
  - fully automated VPN provisioning
  - Service Monitoring

- Site-to-site IPSec connectivity (IPv4)
  - Hub & spoke FlexVPN
  - User-configurable LAN address ranges
  - Cisco ISR zero-touch deployment via PnP

- Configurable redundancy

- Application-aware QoS
  - Using Cisco NBAR

- SSL-VPN Remote Access
  - Cisco AnyConnect client
  - Password-based authentication

- Secured Internet Access - via NAT & FW

- Web Security
  - WEB URL filtering (http)
  - Anti Malware/Anti-Virus
  - Web-based Reputation Filtering
vMS 1.0 Full Service Chain
S2S IPsec + FW + NAT + SSLVPN + Web Security

* VNF Management Interfaces not shown
vMS 1.0 Medium Service Chain
S2S IPsec + FW + NAT + SSLVPN
vMS 1.0 Basic Service Chain
S2S IPsec

- Client
- CPE-1
- CPE-2
- Internet
- DCI (ASR9k)
- ToR (N9k)
- vR (CSR1kv)
- KVM Host

Options:
- L2 (VLAN)
- IPsec (Hub & Spoke)
- IP (Direct Internet Access from CPE)
### vMS Resource Usage

<table>
<thead>
<tr>
<th>Component</th>
<th>vCPU</th>
<th>Memory (GB)</th>
<th>Disk (GB)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portal</td>
<td>4</td>
<td>16</td>
<td>160</td>
<td></td>
</tr>
<tr>
<td>NSO</td>
<td>4</td>
<td>24</td>
<td>20 + 50</td>
<td></td>
</tr>
<tr>
<td>ESC</td>
<td>4</td>
<td>16</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>CSR 1000V</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>ASAv</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>WSAv</td>
<td>2</td>
<td>6</td>
<td>250</td>
<td></td>
</tr>
</tbody>
</table>

Further optimization planned in future vMS releases
## vMS Node Scale

<table>
<thead>
<tr>
<th>Service Chain</th>
<th>vCPU</th>
<th>Memory (GB)</th>
<th>Disk (GB)</th>
<th>Chains Per Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full (CSR + ASAv + WSAv)</td>
<td>6</td>
<td>14</td>
<td>266</td>
<td>7</td>
</tr>
<tr>
<td>Medium (CSR + ASAv)</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>11</td>
</tr>
<tr>
<td>Basic (CSR)</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>24</td>
</tr>
</tbody>
</table>

Scale is mostly CPU bound
Assuming 12-core dual socket CPU, and no CPU oversubscription
## vMS Pod Scale

<table>
<thead>
<tr>
<th>Pod Type</th>
<th>Service Chain</th>
<th>Scale (# Chains)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small Pod</td>
<td>Basic</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Full</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>Medium Pod</td>
<td>Basic</td>
<td>384</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Full</td>
<td>112</td>
<td></td>
</tr>
<tr>
<td>Large Pod</td>
<td>Basic</td>
<td>768</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Full</td>
<td>224</td>
<td></td>
</tr>
</tbody>
</table>
vMS Benefits
vMS 1.0 - Key Benefits

- Self Service – Catalog Driven
- Simplification and Control for End-users
- Agility, Service Velocity (creating new services)
- Elasticity and On-demand spin up of services (VNF service chains)
- Pay-as-you-Grow for Providers
- Zero-Touch Deployment of CPE with PnP
- Lower TCO (agility, automation, simplification) via Virtualization & Cloud Management
- Leverage existing SP Network Infrastructure
- Shorter Time To Revenue with NO upfront CAPEX
- Ability to bundle offers -> Mobile, Video, Security, Cloud Interconnect
End-user Control and Self-Service
End-user Control and Self-Service
CPE On-Boarding (ZTD)
CPE Zero Touch Deployment (ZTD)
Getting from Boot to Day 0 Configuration

CPE Day -1
pnp profile zero-touch
transport https ipv4 <pnp-ip> port 443 remotecert XYZ

Day -1 Config
- Pre-loaded with PnP profile
- Alternate use DHCP with Option 43

1. No Service has asked for this CPE yet. Leave it on an “Unclaimed” list.
2. CPE Reaches out to PnP/NSO via HTTPS. PnP/NSO pushes down Day 0 Config, for Mgmt Tunnel and Mgmt IP
3. FlexVPN Tunnel is setup to Mgmt-HUB. NSO sends Day 1 Config through this Tunnel
4. CPE is added to the CDB Device List.
5. FlexVPN Data Tunnel to Service Chain CSR
6. FlexVPN Mgmt Tunnel to Mgmt-HUB (NSO pushes Day 1)

NSO
- Transaction Database (CDB)
- CPE
- Service Manager
- Device Manager
- Network Element Drivers
- PnP Server

Virto
- Open PnP
- Associate Serial Number (SUDI) to Device Name using Virto Model
- Dual Tunnels to account for NAT. PnP is Pull, NSO is Push

Management IP
= HTTPS PnP (CPE requests Day 0)
WAN Port

Dual Tunnels to account for NAT. PnP is Pull, NSO is Push

CPE Day -1
pnp profile zero-touch
transport https ipv4 <pnp-ip> port 443 remotecert XYZ
Service Velocity
Achieved When Changes in Infrastructure do Not Require a Changed in Service Models.
Function Pack Model

YANG Business Service Model

**Constructed From**

Component YANG Service Models

**Consuming**

Capabilities Exposed in YANG device models / NEDs

- New function packs (usecases) will be released by upcoming vMS solution releases.

- Customers can utilize these function packs as-is, or tweak them per their usecase requirements.

- Cisco Services can be utilized to tweak existing function packs and service models; or build new ones to address customer usecases and business requirements.
PAYG with Cisco Smart Licensing & Call-Home
License Tokens configure on VNF as part of Day 0 config

## Options

1. **Direct cloud access**
   - Cisco product sends usage information directly over the internet. No additional components are needed.

2. **Direct cloud access through an HTTP proxy**
   - Cisco Products send usage information over the internet via a Proxy Server. Any off-the-shelf Proxy will work.

3. **Mediated access through an on-premises satellite – connected**
   - Cisco Products send usage information to a local Mediator such as the Smart Call Home Transport Gateway (Free VM Download), which automatically completes the transaction with the backend.

4. **Mediated access through an on-premises satellite – disconnected**
   - Cisco Products send usage information to a local disconnected satellite, which acts as a local license authority. Once a month, an exchange of human readable information will be performed to keep the databases in sync.

### Ease of use

- **Security Policy**

### Security Policy

- **Ease of use**
vMS Orchestration Details
Service Concept to Service Orchestration

- Virto (Intent)
  - Service Topology
  - SMEs on Service Requirements

- Orchestration (VNF-O)
  - Configurations
  - Config Rules
  - Derived Configurations
  - Derived Service Config Rules

- NSO Function Pack Concept
  - Yang Service Models
  - Mapping Code (Java/Templates)
  - Yang Device Models
    (Leverage Existing when Possible)

- Orchestrate
  - Physical
  - x86
  - Virtual

- Domain Controller
Function Pack (vMS Service Blueprint)
Mapping Service Intent to the Infrastructure

Service Request

Service Models
(Virto)

Mapping Logic
(Fastmap)

Device Models
(Underlay)

Infrastructure

- Calls the Service with the Appropriate Parameters for Instantiation.
- Service Models Represents the Intention of the Service
- Written in Yang
- Contains Service Validation Logic
- Exposes NB through Service API (automatically created)
- Processes the Service Intent
- Calls External Resources as needed to prepare the service.
- Maps the Service Model Parameters to the Device Model
- Abstracts the specifics of the Device from the Service Logic
- Supports different Vendors through the use of Network Element Drivers
Based on the draft: draft-clemm-netmod-yang-network-topo-00.
- Topology Comprises a set of Nodes and Links.
- Links are Pt-Pt and Unidirectional.
- Model is on-boarded to the Physical and Virtual Infrastructure.
Network Topology: “Virto”

- **Virto** is the Fundamental construct for understanding vMS Use Cases.

- A Network Topology is a Blueprint that will describes the intent of the service instantiation.

- This is modeled in **Yang** and is Abstracted from both the physical & virtual infrastructure below.

- The code snippet on the right is one portion of the overall Virto Model.

```yaml
termination-point inside;
termination-point outside;
function {
  security {
    policy ssi-vpn {
      rule 10 {
        match {
          match-type any;
          identity user@XYZ.com {
            user-name user@XYZ.com;
            user-password <removed>;
            admin-state activated;
          }
        }
      } [...]
    }
  }
}
node wsa-01 {
  capabilities [ vnf web-security ];
  node-type {
    cloudvpn-virto {
      vWSA;
    }
  }
  termination-point bridge;
  function {
    security {
      policy filtering-enhanced {
        default-action allow;
        rule 10 {
          match {
            match-type any;
            application adult {
              url-category [ adult ];
            }
          }
        }
      }
    }
  }
```
Mapping Separates Service Intent from Infrastructure
NSO Java & Templates through Fastmap

- Service Models define the exact attributes that can be configured for a service.
- Maps Service Variables to Device Variables.
- Adds Flexibility to Change the Device Configuration without Recompiling.
- Maps to Device Configurations

Java used when Simple Mapping from Service to Device Model is not Enough.

- Set of Validation Rules.
- Maps Service Model to Device Layer
- Declarative model of data-models; two or more stages.
Loading Infrastructure Into NSO (Underlay)
Modeling the Physical and Virtual Topologies

Multiple Infrastructure Topologies can be modeled in NSO.
These constitute the Topology Underlay.
The device models are stored in the CDB.
For rapid efficiency in parsing the device topologies, a GraphDB is used that emulates the device topologies but is optimized for lookups.
Mapping Service Intent to Specific Topology

Querying the Underlay

1. Service Intent

2. Query a GraphDB to derive Topology with Capabilities to Render Service. Selection based on Tags and Capabilities in the Model

3. Derive Virtual Topology

4. VNF Spin-up

5. Launch with Day0

5a. Launch

5b. Network Assignment

6. Ready?

7. Service Callback

8. Commit Configs

Physical

Service Model

Rest/NetConf

GraphDB

Transactional Database

Service Manager

Device Manager

OpenStack

ESC

VNF

OVS

CSR

ASA

WSA

Inet

Overlay (Created Topology)

Underlay

"Virto Model"

tags [cpeSerialNo];
tags [internet];

capabilities [bridging];
capabilities [routing];
capabilities [web-security, vnf];
Example:

- NSO Graph DB queries suitable underlays using tags as defined in the virto service definition
  - Only parts of the topology/components shown here

```plaintext
admin@nso-01> show configuration virto ...
node br-02 {
  tags [ br-internet ];
  capabilities [ bridging ];
  node-type {
    cloudvpn-virto {
      vBridge;
    }
  }
  termination-point firewall;
  termination-point gateway;
}
node br-03 {
  tags [ br-outside ];
  capabilities [ bridging ];
  node-type {
    cloudvpn-virto {
      vBridge;
    }
  }
  termination-point router;
  termination-point tunnel-01_cpe_1427875893928;
  [...]}
```

Bridge between firewall and gateway/Internet is required to use tag “br-internet” in the underlay

Likewise “br-outside” for bridge between firewall and the

```
node dc1-os-2~br-internet-xyz {
  tags [ br-internet ];
  capabilities [ bridging ];
  node-type {
    vswitch {
      ovs;
    }
  }
  supporting-node esc-01;
  termination-point tp-r2-br-internet-xyz {
    supporting-node dc1-os-2;
    tp-ref [ tp-r2-br-internet-xyz ];
  }
}
```

NSO graph search identifies these node as a possible candidate to implement the service

```
node dc1-os-2~net_d2_264 {
  tags [ br-outside ];
  capabilities [ bridging ];
  node-type {
    vswitch {
      ovs;
    }
  }
  supporting-node esc-01;
  termination-point tp-r2-net_d2_264 {
    supporting-node dc1-os-1;
    tp-ref [ tp-r2-net_d2_264 ];
  }
}
```
vMS Service Creation – Step 1

1. End customer signs on to vMS portal
2. Customer configures service on portal

3. Portal constructs service definition XML (“virtual topology” yang model) and sends netconf configuration to NSO

4. NSO compares target topology for the service and compares with current state to apply any changes.

5. **Assuming this is a new customer/service:** NSO identifies which network components it needs to deliver the service (virtual router/firewall/etc.) and creates them.

6. NSO maps target virtual topology to physical topology (i.e. DCs), and instructs the appropriate ESC to spin up the service

7. NSO configures Service on ESC (Netconf)

8. ESC calls OpenStack to create the required virtual resources

9. OpenStack spins up VMs

10. ESC monitors VM coming up

11. ESC signals VM and Service Alive to NSO
vMS Service Creation – Step 2

1. End customer receives the CPE on location and enters its Serial Number into the Portal

2. Portal constructs service definition XML (Yang model) which now includes a serial number assigned to a CPE to NSO

3. NSO compares target configuration for the service and compares with current state to apply any changes. Only change is the serial number assigned to one of the CPEs

4. CPE is connected to network, contacts PnP server

5. PnP server sends day0 configuration to CPE

6. NSO adds CPE device to its list devices

7. NSO builds the final day2 target configuration for all devices (virtual & physical) and configures the diffs on all devices

8. NSO triggers a creation of a DNS record for customer’s ASA’s outside address used for SSL-VPN
Future Use-cases
(possibilities)
MPLS vPE based Service Chain

**MPLS-RA-WSEC-FW-INET** – vMS 1.0 Full service chain, but with MPLS access

- vRR/RR can be provisioned by vMS, or can be done manually or by other OSS systems

---

**Inet-GWY can be P**

---

**VMS Orchestration Scope**

---

**MPLS VPN Network**

---

**Internet**

---

**SSL**

---

**RA**

---

**Inet**

---

**GWY**

---

**vRR**

---

**L3**

---

**L2**

---

**PE**

---

**P**

---

**vPE**

---

**vFW**

---

**vWS**

---

**vMS**

---

**ASA**

---

**WSA**

---

**CSR**

---

**MPLS**

---

**-**

---

**-**

---

**-**

---

© 2015 Cisco and/or its affiliates. All rights reserved. Cisco Public 69
Secure IP Overlay with LISP and GETVPN

Overlay Tunnels
- Overlays
  - LISP
- Secure Overlays
  - LISP + GETVPN
vBranch + vMS

- X86 based vCPE + Services in vBranch
- Additional services in vMS DC
vMS with SFC & NSH

Simplified Service Chaining

- SFC and NSH IETF Drafts
- Topology and Transport independent service chaining
- No hop-by-hop service/VLAN stitching
- NSH header consists of metadata and service path information
- NSH header added to packet, then encapsulated in outer transport header

- Service Classifier (SC) determines (based on Policy) which traffic needs which services
- Service Function (SF) is a physical or virtual device providing network functions
- Service Path is a forwarding path used to realize a service chain, composed of one or more Service Functions
- Service Function Forwarder (SFF) forwards to one or more Service Function nodes, based on the service path identifier, and the service forwarding table
- Metadata and context processing at each node, can trigger service forwarding changes
- Service Orchestrator programs the SC, SF, SFF
Cisco Virtual Topology System
Open, Multi-tenanted, Policy Based SDN Solution for SP NFV and DC Evolution

Management & Orchestration Plane

- 3rd Party VM Manager
- Cisco NSO
- OpenStack
- VCenter

Control Plane

- MP-BGP
- BGP-EVPN
- ToR
- MP-BGP
- BGP-EVPN
- ToR

Data Plane

- IP / MPLS WAN
- DCI
- Bare Metal Workload
- Virtualized Workloads with OVS
- Virtualized Workloads with Feature Rich & High Performance Cisco VTF Solution
- Virtualized Workloads with dVS
- Virtualized Workloads with SR-IOV

VTS GUI

REST API

Cisco NSO

64
Conclusion
Key Takeways

- Cisco Virtual Managed Services (vMS) is an architecture and orchestration solution for enabling NFV services using virtual appliances, overlay networking, and VNF service chaining.

- vMS 1.0 enables virtualised Managed WAN and Managed Security services in the Cloud

- vMS utilizes Network Service Orchestrator (NSO) from Tail-f, which provides Model driven service abstraction, separating Service Intent from Service Instantiation.

- vMS is a platform that can rapidly enable creating and deploying new services

- Multi-tenancy, elasticity, PAYG, automation, service agility and reduced OPEX can be provided by the vMS platform

- Cisco has a broad portfolio of VNF’s that can utilized in the vMS solution

- The Cisco vMS solution will keep introducing new Function Packs to enable new usecases, service topologies and VNF’s
Participate in the “My Favorite Speaker” Contest
Promote Your Favorite Speaker and You Could Be a Winner

• Promote your favorite speaker through Twitter and you could win $200 of Cisco Press products (@CiscoPress)

• Send a tweet and include
  • Your favorite speaker’s Twitter handle @SunilCherukuri
  • Two hashtags: #CLUS #MyFavoriteSpeaker

• You can submit an entry for more than one of your “favorite” speakers

• Don’t forget to follow @CiscoLive and @CiscoPress

• View the official rules at http://bit.ly/CLUSwin
Complete Your Online Session Evaluation

- Give us your feedback to be entered into a Daily Survey Drawing. A daily winner will receive a $750 Amazon gift card.

- Complete your session surveys through the Cisco Live mobile app or your computer on Cisco Live Connect.

Don’t forget: Cisco Live sessions will be available for viewing on-demand after the event at CiscoLive.com/Online
Continue Your Education

• Demos in the Cisco campus
• Walk-in Self-Paced Labs
• Table Topics
• Meet the Engineer 1:1 meetings
• Related sessions
Thank you
TOMORROW starts here.