Cisco Virtual Managed Services (VMS) Solution Overview Guide

First Published: 2018-02-14

Cisco Systems, Inc. www.cisco.com

Contents

Preface

Cisco Virtual Managed Services (VMS) solution changes 'how' managed services are instantiated by shifting the focus from device configuration to capturing the service designer intent through the use of service models. The decoupling of service design from service instantiation allows service providers the ability to deploy VMS services with the knowledge these services can be extended with future releases without the need of a redesign.

This chapter contains the following topics:

Document Objective and Scope

This guide provides a comprehensive explanation and guide to the design of the Cisco Virtual Managed Services solution that enables service providers to offer flexible and extensible services to their business customers. The scope of this release includes:

- VMS Service Packs, Service Ordering, and Service Orchestration
- VMS Service Interface—Front-end and Back-End
- VMS Platform Orchestration—NSO, ESC, and so on.

Cisco Virtual Managed Services Documentation

The Cisco Virtual Managed Services documentation set consists of:

- Cisco Virtual Managed Services (VMS) Installation Guide
- Cisco Virtual Managed Services (VMS) Cloud IWAN Service Pack Guide
- Cisco Virtual Managed Services (VMS) Cloud VPN and Cloud VCE Service Pack Guide
- Cisco Virtual Managed Services (VMS) Troubleshooting Guide

Cisco Virtual Managed Service documentation is available here:

http://www.cisco.com/c/en/us/support/cloud-systems-management/virtual-managed-services/tsd-products-support-series-home.html

Cisco Virtual Managed Services Solution Overview

Cisco Virtual Managed Services (VMS) is a software solution designed to separate the intent of the service designer from the underlying infrastructure that is destined to change with the latest increase in speed, features, and scalability. The VMS solution is designed to shift the focus of design from the configuration of devices to the end-to-end service, enabling Service Providers to focus on their customer requirements versus networking device limitations.

Cisco Virtual Managed Services solution shifts the deployment of managed services away from the manual configuration of the latest network devices to the creation of a software abstraction to represent the service definition. Through the advanced service orchestration capabilities of the Cisco Network Services Orchestrator (NSO), an abstraction of the service exists as a YANG based service model, which is than processed by specialized Fastmap code, resulting in the instantiation of the service atop a virtual infrastructure. This approach allows the service intent of the user, to be realized through the use of service models to automate the creation and customization of SD-WAN services.

Through the combination of a new VMS platform and service packs, the VMS solution offers a complete platform that allows the service providers to offer the next-generation Managed Services.

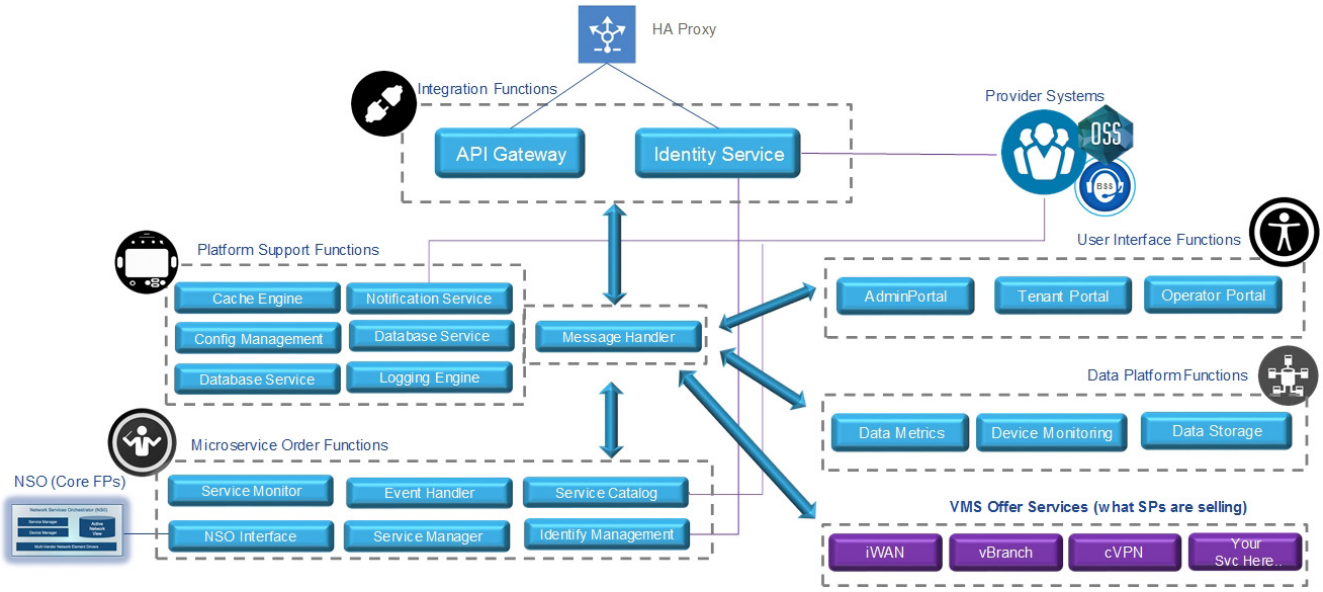
The chapter has the following sections:

VMS Platform

Cisco VMS is a service creation platform, implementing the different functions required to instantiate and provision virtual and physical elements in order to construct end to end managed services for service provider customers. The VMS solution uses software-defined networking (SDN), Network Functions Virtualization (NFV), Open APIs, and advanced orchestration capabilities to deliver a suite of business services through service providers' cloud infrastructure and over their existing network infrastructure.

The VMS platform is comprised into layers—each of which abstract the layer below it—and each layer may be scaled horizontally. A layered abstraction approach, with well documented API contracts between the layers, allows for modularization, a key tenet of the VMS platform. The modularization allows for the separation of concerns, independent scaling, development velocity, and ultimately component replacement, if necessary. The illustration below depicts the overall framework of VMS architecture.

Figure 1 VMS Functional Architecture



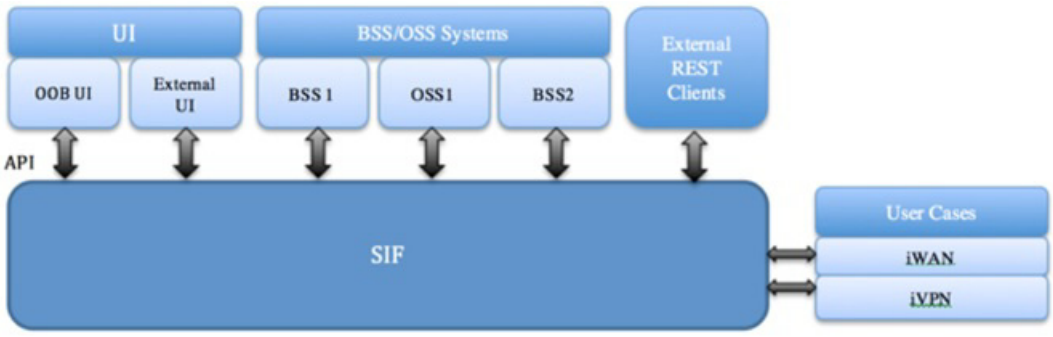
The key capabilities of VMS solution are:

- Automated end-to-end SD-WAN services managed from the service provider cloud.
- Secure multi-tenant cloud managed platform, simplified orchestration, and tenant self-service.
- SD-WAN created with Zero Touch Provisioning and validated service packs.
- Rapidly create new monetized services, modify existing services instantly from the cloud.
- Perfect solution for distributed customers looking for lower cost and self-managed SD-WAN options.

Service Integration Framework (SIF)

The SIF layer enables external entities to consume the visibility and control provided by platform functionality, optional capabilities, and use case specific logic in the format they require.

Figure 2 SIF Interactions



The SIF provide not only standard APIs required to operate the platform and use cases, but also supports the customization and extension of these APIs on per implementation or per customer basis. The ability of the SIF to be extended and customized to consume use case specific functions is critical to the ability for VMS to support multiple use cases deployed in a service provider setup with reduced operationalization required beyond the initial use case.

The consumers of the SIF include:

- Business Support Systems (BSS)/Operations Support Systems (OSS)
- User Interface (UI)
- External Clients
- Service Packs

Platform Integration Framework (PIF)

The VMS PIF is a thin layer that sits north of the SOP and Data Platform. It provides a consistent interface to the SIF layer and external systems that need to interact with the VMS Platform via APIs.

The VMS PIF provides a set of REST end points for the VMS platform, shielding the consumer of the APIs from being aware of the actual platform components that provides the implementation.

For the consumers of these APIs, integration is made easier as only a single set of credentials are needed to use the APIs of multiple platform components. These API are for public consumption and are secured via authentication methods such as OAuth2.

VMS Microservices

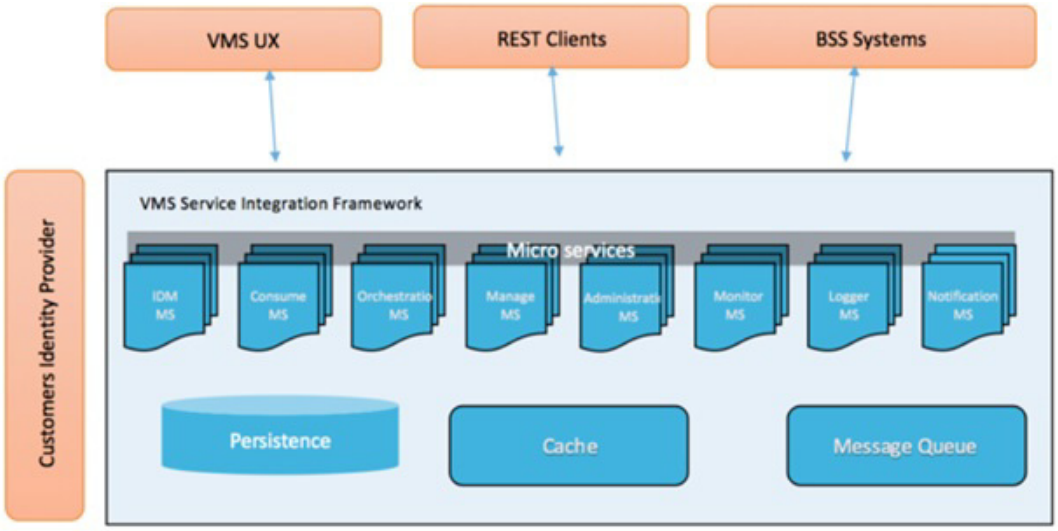
All of SIF functions and integrations are enabled as microservices that contain the logic to consume the functions exposed by the platform and use cases. There are a standard set of microservices that consume the base platform functionality required to operate the platform.

Each use case can also provide one or more microservices for any additional use case specific functionality that needs to be enabled.

The following functions are core microservices that exist in the SIF layer of the VMS platform:

- Identity Management
- Consume
- Manage
- Monitor
- Orchestration
- Notification
- Logger
- Administration

Figure 3 Microservice Architecture in VMS



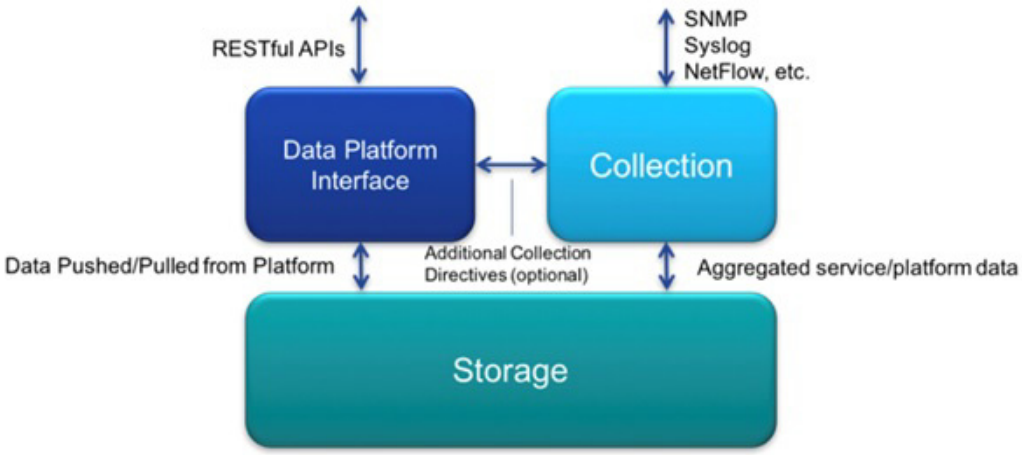
Data Platform

The data platform interface is the mechanism by which the data platform component interacts with the other platform components and applications. There are a number of actions that must be supported by the data platform component, and thus exposed via the service interface. These include, but are not limited to:

- Data Retrieval (Pull)
- Configurable Collection
- Data Subscription/Streaming (Triggers/Events)

The data platform component of the architecture is broken down into the following major sub-functions, as shown in the illustration below.

Figure 4 Data Platform Architecture - High Level



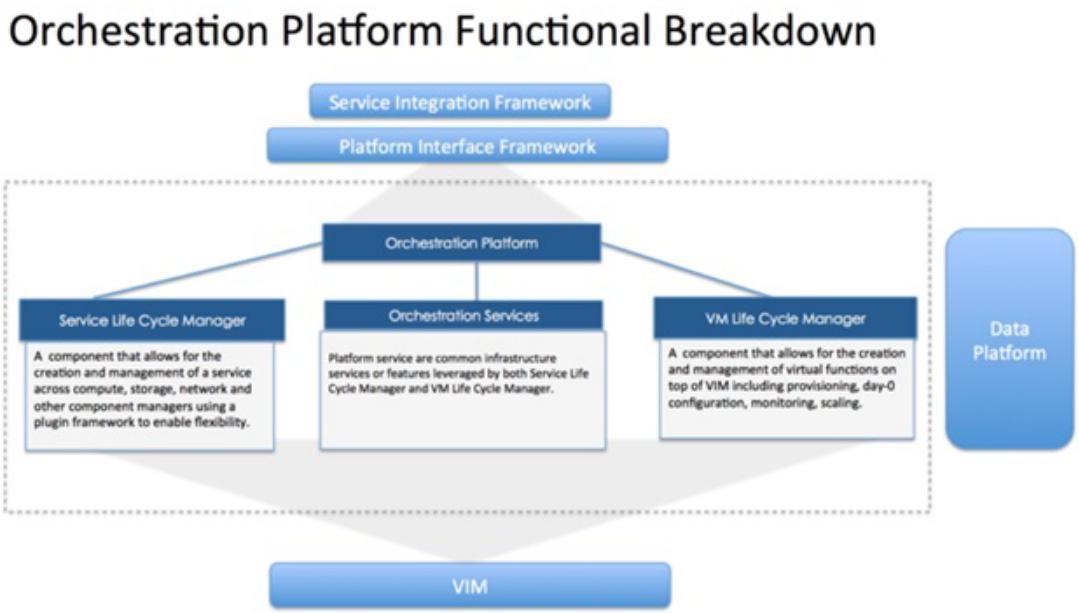
The major function performed by the data platform component is the collection of data. The collection interface allows a platform component or application the ability to instruct the data platform which data to collect, the frequency with this to collect, and how to store and correlate the data. This is to enable other platform components or external systems to augment the collection enabled by the SOP provisioning. For example, the instantiation of a typical service chain will program the devices participating in the chain to send data (Syslog/SNMP) to the data platform.

Service Orchestration Framework

The Service Orchestration Platform contains the logic that brings virtual and physical devices under management and deploys the required service configurations necessary to bring service and service topologies online and under management. The Orchestration platform contains both the Service Lifecycle Manager and Virtual Network Function (VNF) Lifecycle Manager layers and represents the resource facing services (RFS).

The illustration below depicts the orchestration platform for VMS along with the major roles of the components. Additionally, the interactions with the platform interface, VIM, and data platform are highlighted to further illustrate their relationships.

Figure 5 Orchestration Platform Functional Breakdown

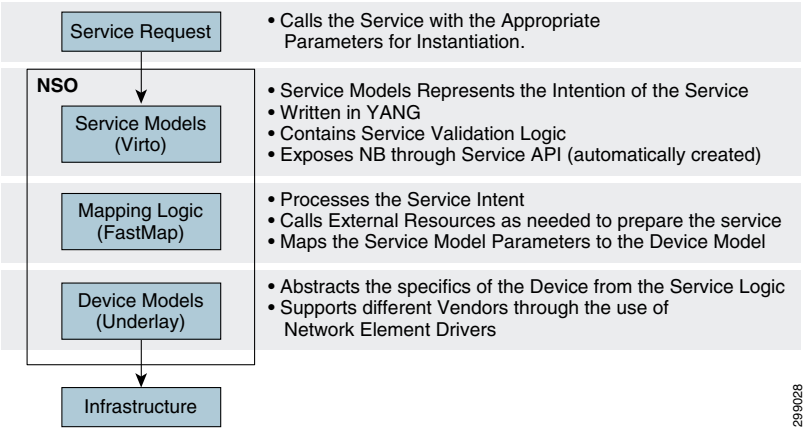


VMS Service Packs

VMS service packs contain all the elements necessary to orchestrate a service requested by the business customer or tenant operator. The operator initiates the service request to Cisco NSO through the API created as part of the software service pack at compile time. The service request may be passed through the Service Interface (self-service portal) or could be an open API call from the existing Provider OSS/BSS systems.

Figure 6 shows a detailed view of the service pack elements. The service definition model is written in Yang and describes the end-to-end service. In the current release, a service definition model is referred to as a 'Virto'. The Yang service model software has validation logic that validates the service requests to ensure that incorrect service requests are not completed.

Figure 6 Service Pack Fundamentals



The Service Model must be mapped to Device Models, which generate the service, and device configurations that are applied the physical and virtual infrastructure devices. Cisco NSO software uses innovative Fastmap functionality to handle this process. This mapping with Fastmap software could be accomplished through a template or using Java for more complex mapping applications.

Device models also use Yang modeling constructs but are intended to model the infrastructure rather than the service. This set of device models is what is referred to as the underlay and is an abstraction of only the physical and virtual infrastructure that must be configured to enable the requested service.

For more information on VMS 3.1 service packs, see Cisco VMS Service Packs.

Cisco VMS Solution Architecture

This chapter provides an overview of the VMS solution software architecture internals, provides insight to how the orchestration engine processes a service request, using service definition models to create and instantiate cloud based services.

This chapter contains the following sections:

VMS Modular Architecture

Service providers can automate the creation of cloud based business services by using YANG based service models in the orchestration engine software modules, and service function chaining of the required virtual network functions (VNFs) to enable those services.

As VMS evolves the architecture will continue to modularize components, creating a clear demarcation points between various layers in the solution allowing maximum flexibility in both commercial and deployment models.

VMS Modules

- Service Interface (SI)—Customer Facing Level

  The Service Interface consists of two modules, a customer facing Front-End and service intent to service request processing Back-End.

  The SI Front-End is the Self-Service User Interface; from here the end-customer can select the Service Package that meet their VNF requirements. At that point, the end-customer can construct VMS services consistent with the Service Package, in this example, the Advance Service Package is selected.

  The SI Back-End is separate module that constructs the service request based on the service intent determined by the end-customer choices in the From-End module.

- VMS Platform—Resource Facing Level

  The VMS Platform mirrors the ETSI MANO Standard for a virtualization orchestration model. The NFV Manager or Network Service Orchestrator provisions services based on service packs that logically mirror CFS level Service Packages. Service packs are internal software modules that house the service models and Fastmap execution logic that define specific Service Packages. Based on these service models, the NSO will use the internal Fastmap code to provision the service components through the Elastic Service Controller (ESC).

  The ESC is the VNF Lifecycle Manager, based in request from the NSO, this component will execute Create, Read, Update, and Destroy (CRUD) operations, resulting in operation request to the Virtual Infrastructure Manager.

Service Interface

The Customer Facing Service (CFS) role is VMS Solution customer experience, it is the responsibility of the Service Interface to capture the service intent of the customer and then form the service request to Cisco Network Service Orchestrator (NSO). The Service Interface is composed of two subsystems, the Front-End and the Back-End. The Front-End is the Web portal that supports several roles, admin, operator, and customer. Based on a Username/Password combination, the Service Interface determines the role of the user based on the log on authentication process. With a successful authentication, the user is directed to the Web based on the pre-defined role of the username. The Back-End is the composition of micro-services that together communicate with various components in the VMS Solution. From the customer perspective, the Back-End is responsible for processing their service intent, based on the interaction with the customer Web interface or self-service user portal, and creating a parameterized service request to send to the VMS Platform NSO.

Figure 7 shows a modular view of the Service Interface, the Front-End consists of several portal interfaces, that based on a user role, provides the following operations:

- Browse
- Buy and Activate
- Use and Consume
- Manage
- Support

The VMS Services that are available through the portals are dependent on the Service Packages made available by the Service Provider; currently VMS provides the Cloud VPN and Cloud IWAN service packages. Access to various interface portals is based on the role of the user, determined during the login process; all interface access is secured by a password. The VMS Service Interface utilizes an OpenAM ID system to determine customer identity; the user login information is available to the Front-End and Back-End modules.

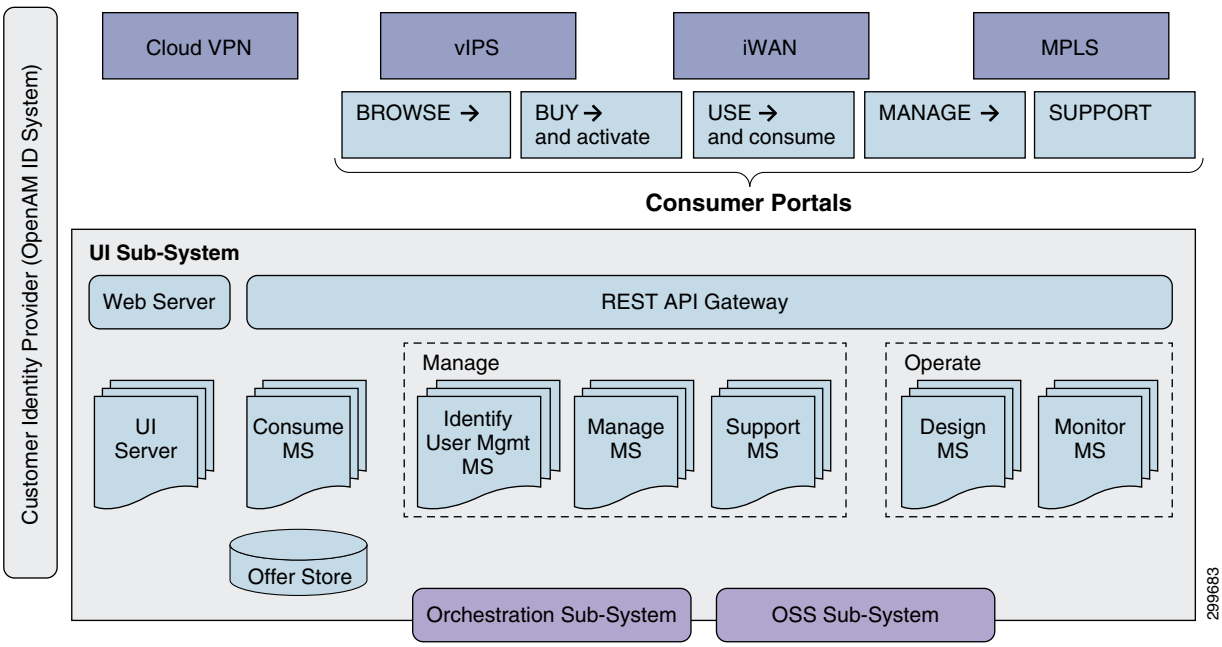
The Service Interface Back-End communicates with the Front-End through a REST API Gateway. The interface portals of the Front-End rely on Back-End micro-services to process user data entered in the various interface portal screens. Dependent on the type of interface portal and data entered, the information will be sent to/from the Back-End API and delivered to/sent from the micro-service responsible for processing the incoming data. The Back-End micro-services are responsible for multiple functions listed below; individual micro-services communicates with VMS modules or other OSS modules to fulfill their functions.

- Business functions microservices
  - Identity Management provides authentication and authorization. (Authorization is the component that defines what you have access to. Federation with other identity engines is possible, but requires integration.)
  - Consume (Catalog)
  - Manage (Service Provisioning and Life-Cycle)
  - Monitor (Service Status and Metrics)
  - Orchestration (NSO RFS Integrations)
- Technical Microservices
  - Notification (Email Templates)
  - Logger
- Microservices-based
  - Scalable, swappable and extensible (DevOps Enablement)

- Each service can be deployed independently of other services - easier to deploy new versions of services frequently
- Each microservice is relatively small (Easier for a developer to understand)
- Improved fault isolation. For example, if there is a memory leak in one service then only that service will be affected
- Each service can be developed and deployed independently
- Eliminates any long-term commitment to a technology stack

- Robust and well documented REST APIs to enable custom app development
  - Use our out-of-box UI or build your own
- Supports heterogeneous identity providers (IDM)

Figure 7 Service Interface Reference Architecture



The Service Interface and serves as the Customer Facing Service layer in the architecture. Designed as two separate modules, the Service Interface's Front-End and Back-End both play important roles in simplifying the provisioning of services.

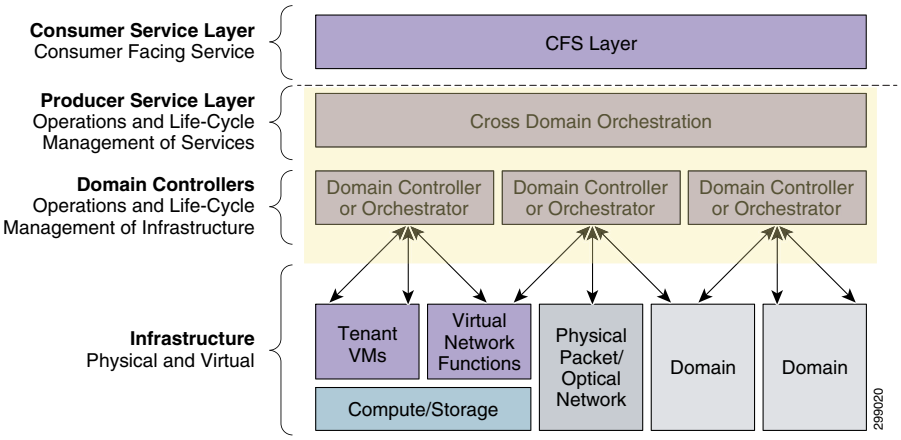
As part of solution simplification, the Front-End is designed as an all in one Web based solution GUI. Based on the user login type, the user will be presented with one of three service interfaces: administer, operator, or user. The Service Interface, based on user role, will allow the Service Provider administrator to provision tenant space for end-users, while the Service Provider operator can view the status of all services running, and lastly the user role is for the Service Provider end-customer to order the service based on their requirements.

The Service Interface also contains an independent Back-End module that is responsible for forming a user service definition or intent into a well-defined request to present to the VMS Platform for service provisioning. The Back-End is designed with a REST interface, allowing the Service Provider the choice of Front-Ends, the VMS Service Interface or another OSS front-end. The logic behind the inclusion of a Service Interface Back-End is to ensure a well-formed service request message to the VMS Solution.

Architecture and ETSI Mano Standards

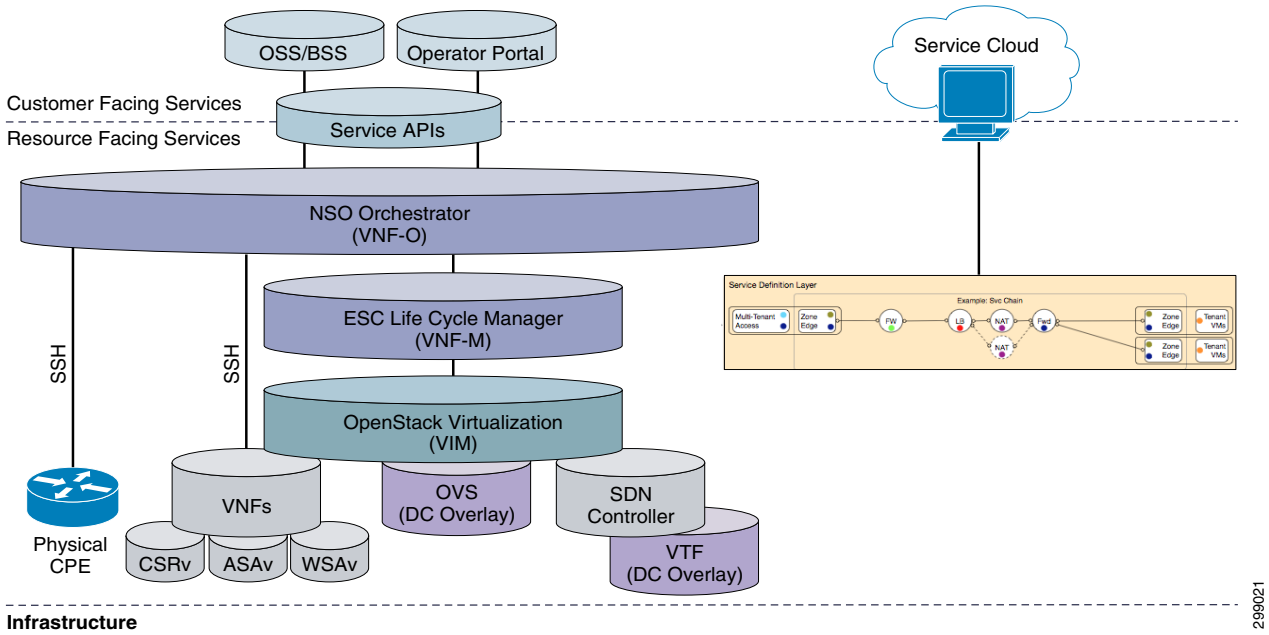
The ETSI NFV Management and Orchestration (MANO) standards define a method of orchestration whereby the components of orchestration of compartmentalized to specific functions. Figure 8 shows the ETSI MANO model. The layers are broken into Orchestrator (NFVO), Life Cycle Management (VNF-M), and virtual interface management (VIM). The NFV orchestrator is responsible for understanding the overall service being instantiated. The life cycle management is for the management and monitoring of virtual network functions (VNFs). Finally, the VIM handles interconnecting physically and virtual network and compute infrastructure.

Figure 8 VMS Uses ETSI Mano Architecture for Orchestration



Cisco Virtual Managed Services solution follows the ETS MANO model as show in Figure 9. The Cisco Network Services Orchestrator (NSO) orchestration engine software modules handle the NFVO functions. The Elastic Services Controller (ESC) software modules are responsible for VNF life cycle management (VNF-M). OpenStack networking software plug-ins modules provide virtual infrastructure management (VIM) functionality.

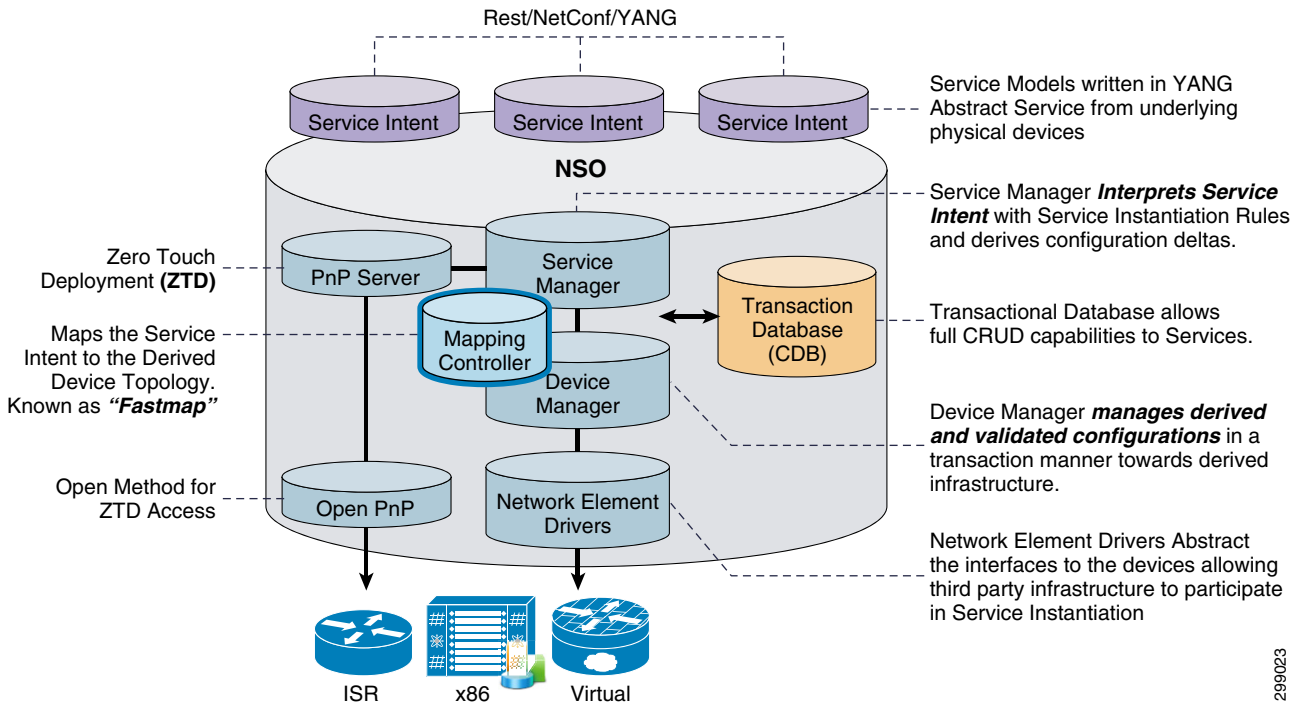
Figure 9 VMS Orchestration Component Mapping



Cisco Network Services Orchestrator (NSO)

Cisco NSO provides the automated services orchestration capabilities. Cisco NSO receives a service intent request through the open API interface presented northbound to the service interface (or customer OSS/BSS). For all services intended to be instantiated, NSO has a Yang service definition model loaded into the transaction database to handle such a request. These are shown in Figure 10 as individual service intent modules.

Figure 10 VMS VNF-Orchestrator–NSO



The Service Manager software subcomponent of NSO processes the service intent, corresponding to a particular service model. The purpose of the Service Manager is to analyze the service request, then interpret the request to a specific set of configuration asks. Similarly, the Device Manager is used to abstract out the infrastructure also based on a modeling language. A Mapping Controller is responsible for moving service models to device models. This will be discussed in more detail in later sections.

A transaction database known as the "CDB" stores all elements related to configuration and devices. This highly functional database can be used to derive all the transaction necessary to push a service intent to service instantiation.

Network Element Driver (NED) software modules are used to abstract out the different physical and network devices to which service configuration data may be pushed. As such the NED allows the service definition models to apply to equipment from different vendors.

Cisco NSO as the Plug-and-Play Server

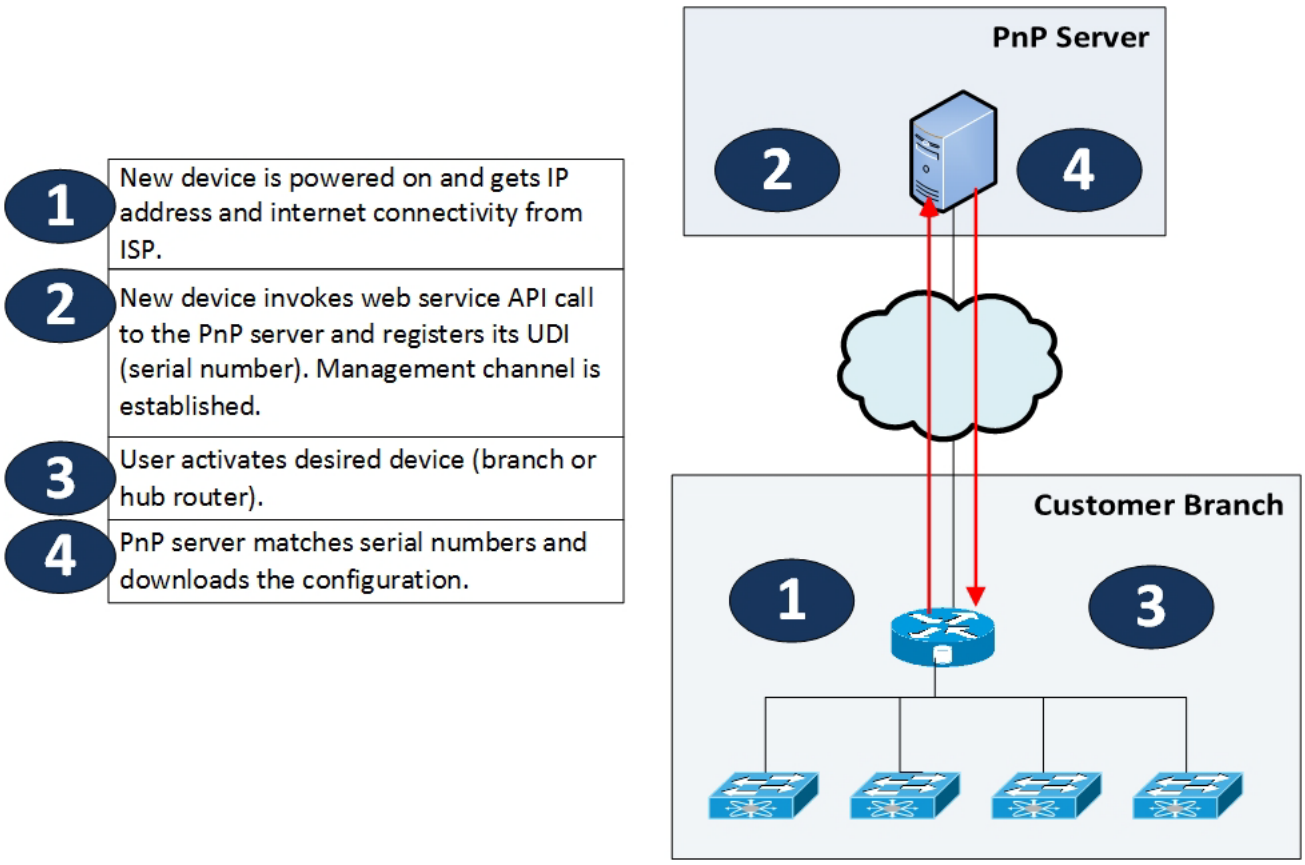
The Plug-and-Play (PnP) Server software element is used to handle and achieve zero touch deployment (ZTD) for CPEs coming online and wanting to utilize the services configured orchestration engine. In the VMS solution, NSO also functions as the PnP server. The site CPE deployments are supported in VMS by a PnP service, which is accessible as a NSO PnP Resource Facing Service (RFS) API. Once a CPE device is connected to the Internet, a "call home" protocol communicates with the NSO PnP RFS API via a secure PnP Configuration Manager located in the Service Provider VMS Cloud. The CPE device is fully configure using a four step process:

- Day-(-1) –Initial config of CPE to find PnP Server
- Day-0 –Configuration of the device management interface
- Day-1 –System configuration, basic interface and system configuration
- Day-2 –Service configuration, service specific configuration

Inclusion of the PnP service in the VMS Solution removes costly truck-rolls from the service deployment model, which are required to install and configure each CPE device. The removal of this activity greatly reduces the cost of service deployment, a cost that has long haunted traditional Managed Services.

The illustration below depicts the device registration process.

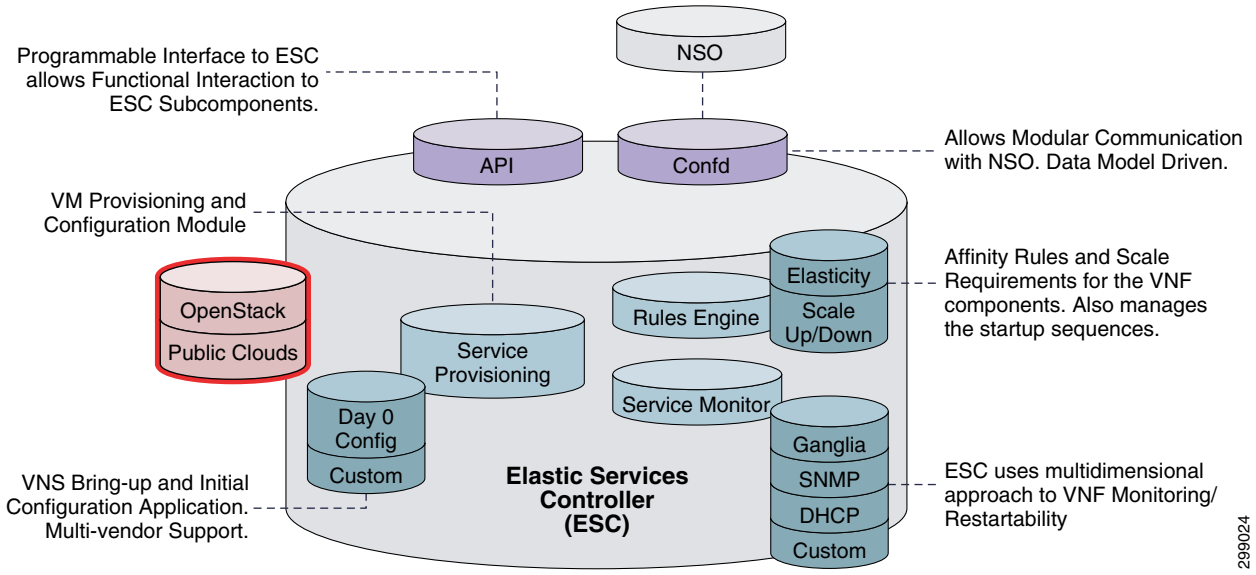
Figure 11 Registering a Device or CPE



Cisco Elastic Service Controller (ESC)

The elastic services controller software module provides the automation of life cycle management for any VNFs that a service needs to have instantiated–also known as the VNF-M in ETSI MANO terms. There are three main functions achieved through the ESC software module including service provisioning, a rules engine, and service monitoring (see Figure 12).

Figure 12 VMS VNF-Manager–ESC



The service provisioning function takes its VNF instantiation actions from NSO requests which are pushed down via a NED software process into the ESC. The purpose of ESC software module is to bring VNF devices online and be network reachable and configurable from the NSO software modules. The ESC software module use calls to the OpenStack networking software modules to accomplish the tasks of launching virtual machines. Specifically, the ESC software will call the OpenStack Nova software process for compute launch and the OpenStack Neutron software process to establish network connectivity to the newly created virtual machines. The OpenStack VIM software has no knowledge of the overall service being instantiated. OpenStack software is merely acting on the requests pushed by ESC software. All external networks must be created in OpenStack prior to the NSO instantiating the service requests.

The rules engine drives a critical feature of the ESC software, elasticity. Elasticity refers to the ability to spin up additional VNFs that may be required from increasing service load. Conversely elasticity also includes removing VNFs from service when they are no longer needed. In this release, the elasticity configuration is locked at “1” in the ESC software schema. This will prevent the VNFs from scaling up or down. The configurability of the ESC software rules engine elasticity factor will be enabled in subsequent releases.

Service monitoring is essential for detecting whether a VNF has stopped operating or is no longer functioning as expected per the service level agreement (SLA). The ESC software module uses various mechanisms such as SNMP and Ganglia, to detect that a VNF is acting out of profile. When this occurs, the ESC software restarts the VNF, maintaining the user license, and potentially being moved to a new compute resource if necessary. The VNF service monitoring information may also be sent as traps northbound via Open APIs to service assurance tools.

VMS Virtual Infrastructure Manager–OpenStack

The VMS platform assumes the Mitaka version of OpenStack as a Virtual Infrastructure Manager (VIM). The Elastic Service Controller will communicates with OpenStack Nova compute services and Neutron network service to provision Cloud VPN VNF types and create the a network connectivity. OpenStack is implemented as a series of controller services, these services are installed independent of VMS and requires dedicated servers.

For provisioning of VNF, the ESC communicates through a Nova REST Interface to provision a VM based on a particular VNF Type resources and image requirements. Provisioning of the virtual interfaces, Linux Bridges and OVS segments, the ESC communicates through a Neutron REST Interface. Lastly, VNF lifecycle operations utilize the same REST interfaces.

Cisco VMS Service Instantiation

The key differentiator of Cisco VMS is the orchestration and management of services using a deterministic and repeatable method, resulting in the consistent instantiation of a service. VMS, through the use of the service packs creates a consistent and well-formed service request, is able to instantiate a service based a well-defined service model and associated execution code. Each instantiated service will share common feature configuration and service topology.

This chapter contains the following sections:

Service Blueprints

The service pack services are made available at the Service Interface as a 'blueprint' or service definition model of an end-to-end service. For example, the service interface portal may offer a blueprint for a Virtual Router service connected to a CPE device. A service definition model 'blueprint' is essentially a set of intellectual property developed to render a customized service that is intended to operate over physical and virtual infrastructure. That intellectual property is referred to as a software service pack.

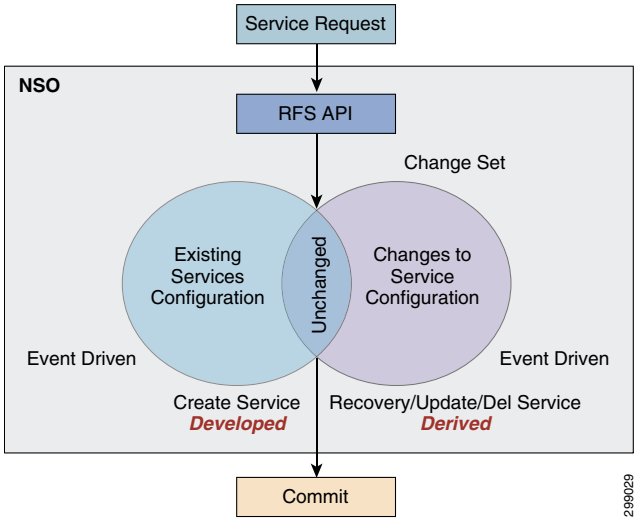
A software service pack lives in the Cisco NSO software modules and has several subcomponents. These subcomponents are the Service Model, Mapping Code, Device Model, and Network Element Drivers (NEDs). All of these components are required to instantiate service intention. When the elements of a service pack are compiled by the service developer, Cisco VMS solution automatically creates the APIs necessary for a northbound application (such as a portal) to request a given service definition model 'blueprint'.

Create, Read, Update, and Delete Configuration Optimizations

Create, Read, Update, Delete (CRUD) operations are at the heart of any services orchestration system. Typically these are achieved through somewhat complex workflow design. The Cisco NSO orchestrator software models provide a unique approach to solving this complexity issue. Hardened software processes enabled by Tail-F, service developers are only required to write the service create functions. Cisco NSO software automatically calculates all functions necessary to carry out the Read, Update, and Delete functions.

In Figure 13, a Venn diagram shows this operations concept. When a service request is made, Cisco NSO software will examine the requested service against any existing service currently deployed. A change set is then determined that represents the delta between the two service model definitions in the transaction database (CDB). Cisco NSO is capable of deriving all the actions necessary to move that new change of the service or device model set into operation. The transactional database (CDB) software allows the change sets to be unrolled either automatically if a service fails, or through operator request back to a previous stable state.

Figure 13 Cisco NSO Service Request Functionality



Service Device Mapping

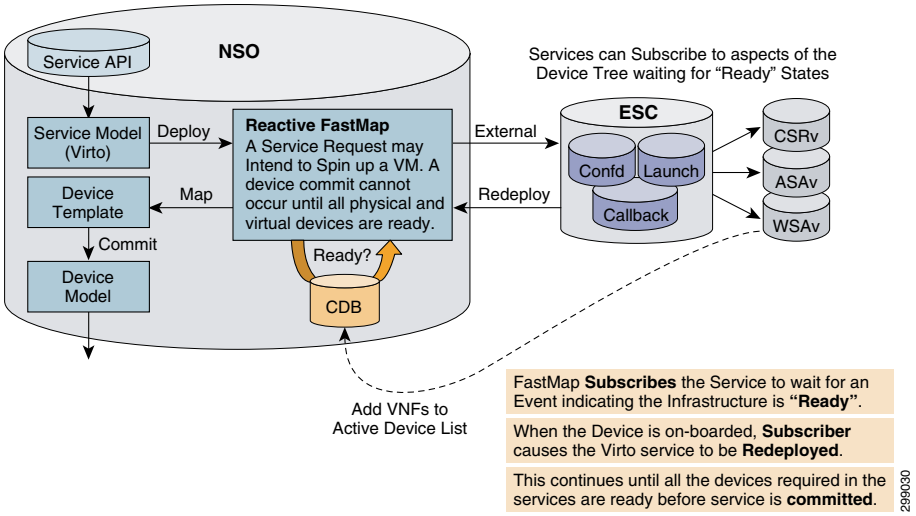
The purpose of the Fastmap software in the service pack of a service blueprint is to map service intent to the device infrastructure. The Fastmap software process uses Java logic resulting in the creation of a service template that is mapped to a device model. This process works very well for physical infrastructure. But what if the infrastructure does not yet exist, as is the case with virtual machines (VMs) running virtual network functions (VNFs). In these conditions, Cisco NSO can make use of the Reactive Fastmap software process.

The Reactive Fastmap software is capable of detecting when a service model requires a virtual network function (VNF) in the requested service model. Cisco NSO cannot complete service model mapping until all devices, both physical and virtual, are active. However, the Transactional Database (CDB) software cannot remain locked while the virtual devices are started.

In the case of the required VNF, NSO will call the Elastic Service Controller (ESC) software modules to handle the VNF life-cycle management. The Fastmap software process subscribes the service to an event in the transaction database indicating a VNF has been started and brought under management. When this occurs, Cisco NSO software will attempt to redeploy the service model requests. Here the entire service request process begins again with Cisco NSO software checking that all devices are in a ready state. If all physical and virtual devices are not ready, the Cisco NSO software modules will defer the service request again.

Eventually either the service deployment will fail because all devices cannot be brought into a ready state or all required components become fully available. In the case where the devices do not all appear in the ready state, the service request fails and all potential configurations are rolled back. When Cisco NSO detects that all devices are in a ready state, the service request process will proceed to map the service model to the appropriate device model. This entire process is shown in Figure 14.

Figure 14 Mapping Services to Non-Existing Infrastructure, Innovative Fastmap Functionality (Cloud VPN)



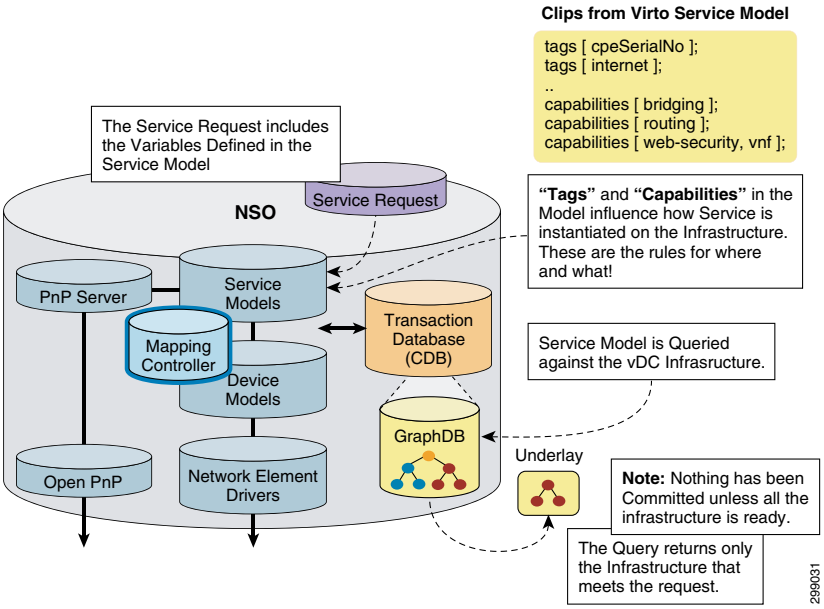
Configuring Infrastructure Elements

An underlay represents the physical and virtual infrastructure to which Cloud VPN services will be orchestrated. An underlay consists of physical and virtual devices, links, network bridges, and resource pools. Cisco NSO software is capable of supporting the loading of multiple infrastructure topologies into the transactional database (CDB).

The entire infrastructures is typically not necessary to instantiate every service. A service typically will require a subset of a given infrastructure. Cisco NSO software implements a mechanism known as the GraphDB, which is a tree representation of all the elements in the infrastructure. GraphDB software in Cisco NSO allows the Fastmap software processes to use queries based on the service model to find the applicable infrastructure required for the service.

To influence the result of the GraphDB query, Tags and Capabilities service requirements are programmed into the service request. Tags provide a description in the service model to which the Fastmap can parse that provides data to the mapping logic. Similarly, capabilities in the service definition model describe what is required of the infrastructure components for that particular service. The process of querying the GraphDB software modules is illustrated in Figure 15.

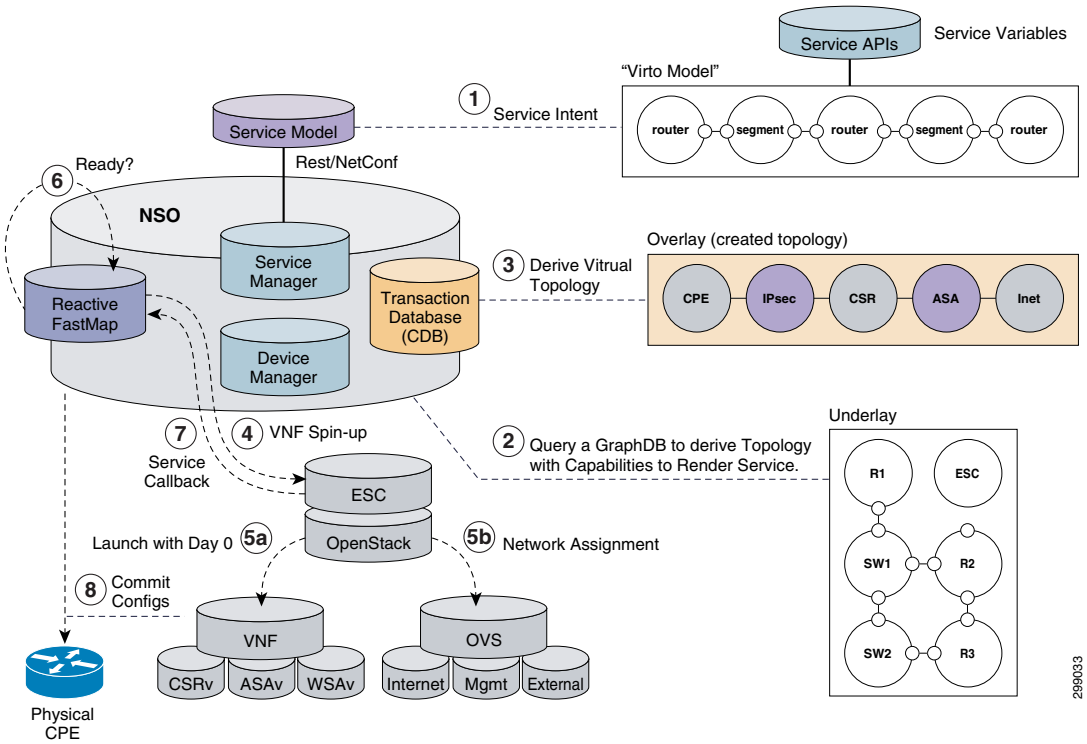
Figure 15 Mapping Service Intent to Specific Topology, Querying the Underlay



Putting the Pieces Together

Figure 16 is a complete representation of the advanced orchestration processes required to on-board a service. The first step is to ensure the Service Models required have been loaded into the Cisco NSO transaction database (CDB) The service model configurations that make up the service pack are written specifically to offer the required service. A service request received by Cisco NSO software will result in configuration checks against the service model which is handed off to the Fastmap process in Step 2. At this point GraphDB software is queried to retrieve only the infrastructure necessary to realize the specific service. Tags and Capabilities parameters in the service request will influence the service query. The output of the query request is the Service Overlay illustrated in Step 3. Based on the completion of these steps, Cisco NSO software will attempt to instantiate and create a Cloud VPN service comprised of both physical and virtual components.

Figure 16 Service Model Consumption



The reactive Fastmap software in Cisco NSO makes an external call to the Elastic Service Controller (ESC) software to spin up the required VMs in Step 4. ESC understands the affinity rules for launching the service, working with OpenStack VIM software (Nova, Neutron) to launch the necessary compute resources and bring up networking interfaces that enable the VMs to be manageable (Step 5). Cisco NSO software enables the initial minimal configurations of the newly launched VMs that are required to connect the VNFs to the management channel of Cisco NSO software.

During the process of launching the VMs, Cisco NSO software will release control of the database for other functions and wait (Step 6) until the VMs appear to be in the ready state. As with a CPE, a VM is online and ready once it completes a online is service call back process (Step 7) to put the newly started device into the Cisco NSO transactional database. Cisco NSO software will attempt to redeploy the service each time a service callback is made. However, no changes will be committed until all devices in the model register as ready in the transactional database (CDB).

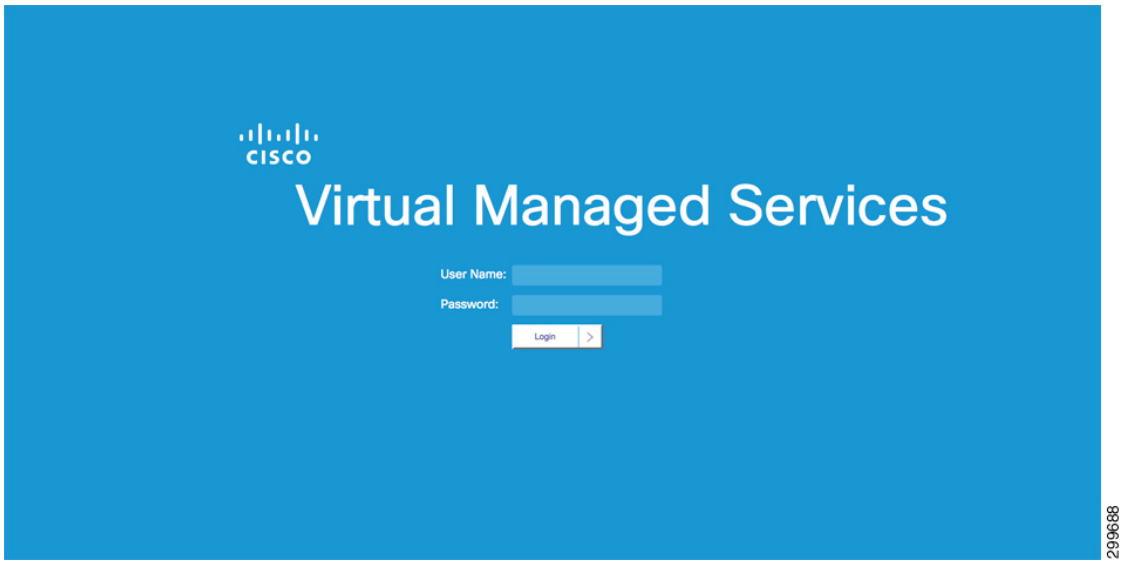
Cisco NSO attempts to reach the physical and virtual components in the service overlay via SSH connectivity over the management network interface (Step 8). The service model configuration applied by Cisco NSO is referred to as the Day 1 configuration. When all configurations to all physical and virtual devices are complete then the specific service is considered deployed. If any of the configurations fail, then the service and device model changes applied in the Fastmap or Reactive Map processes are rolled back to the last known working configurations in the transaction database (CDB).

Cisco VMS Service Interface

Service Interface—Front-End, The VMS Customer Experience

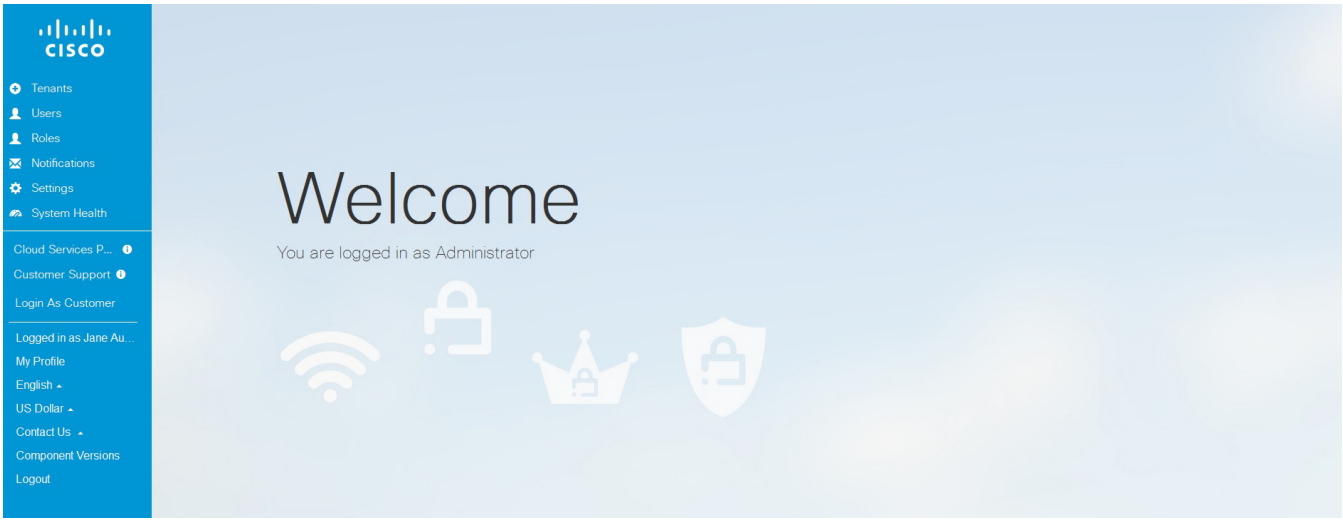
When accessing the Service Interface Web interfaces, the user will be presented with a login screen as shown in Figure 17, based on the user login, they will be present with interface portal for their defined role.

Figure 17 VMS Service Interface: Login Screen



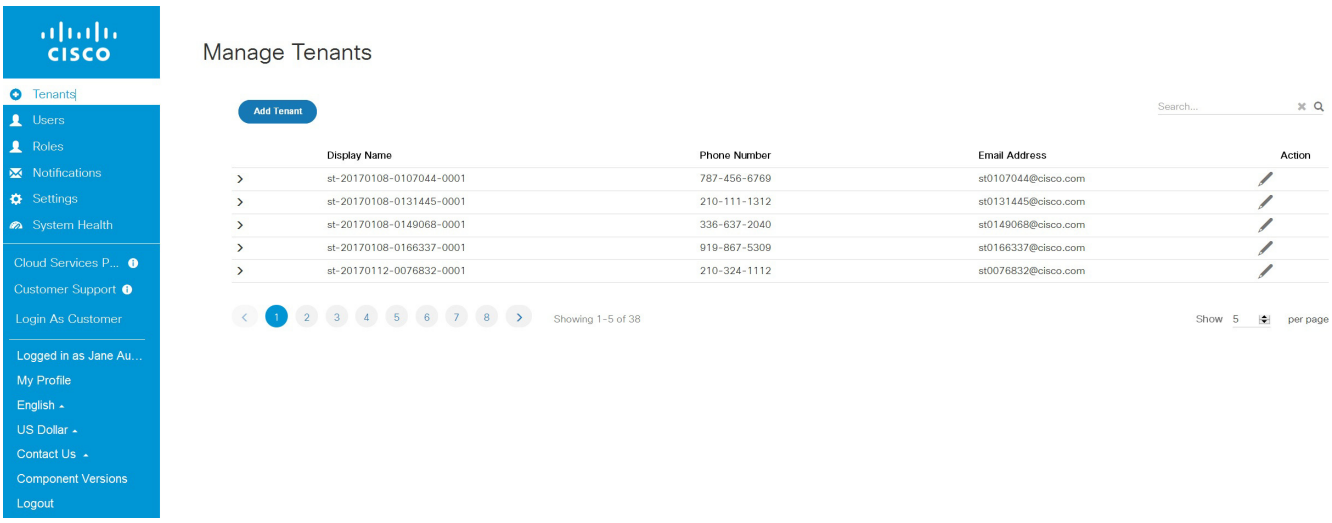
When logging in as the administrator, the user is presented with the screen in Figure 17. As the administrator, VMS service management and customer/tenant domain management are managed from this service interface, the administrator is a Service Provider role.

Figure 18 VMS Service Interface Administrator: Welcome Screen



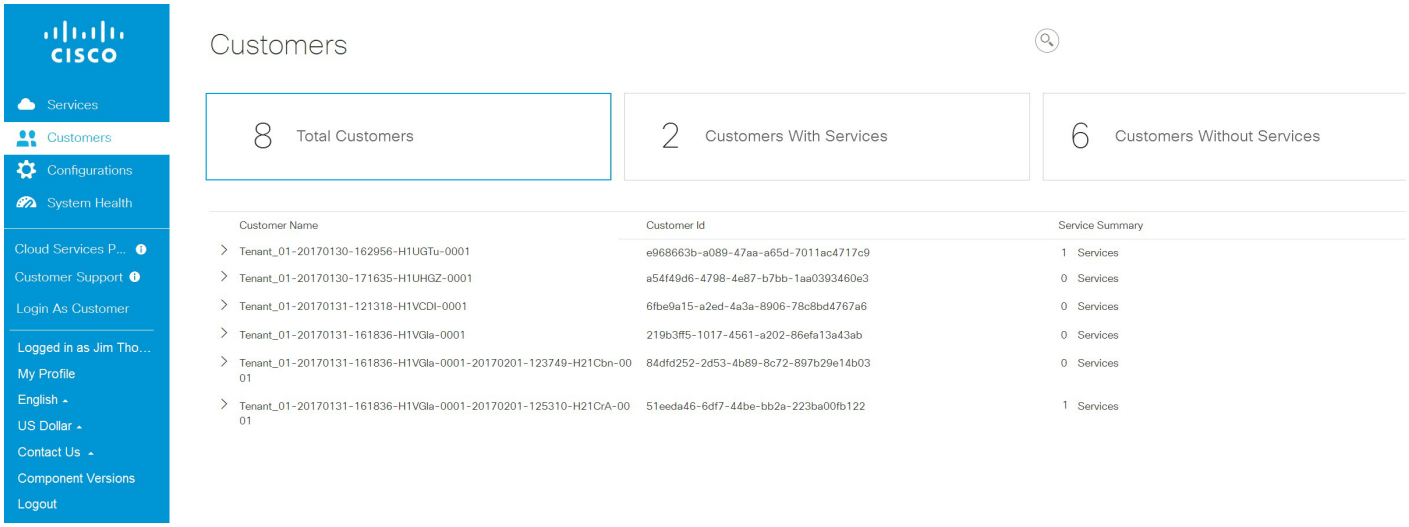
To create a new Tenant domain, the administrator would select Tenants, as Figure 19 shows the Managed Tenants screen is displayed based on this action. A Tenant is an enterprise who has contracted for VMS Managed Services from a Service Provider.

Figure 19 VMS Service Interface Administrator: Mange Tenant Screen



Once a system is active, the Service Provider can utilize the operator role to monitor the services, as shown in Figure 20. This service portal is used by the Service Provider to monitor customer (tenant) services.

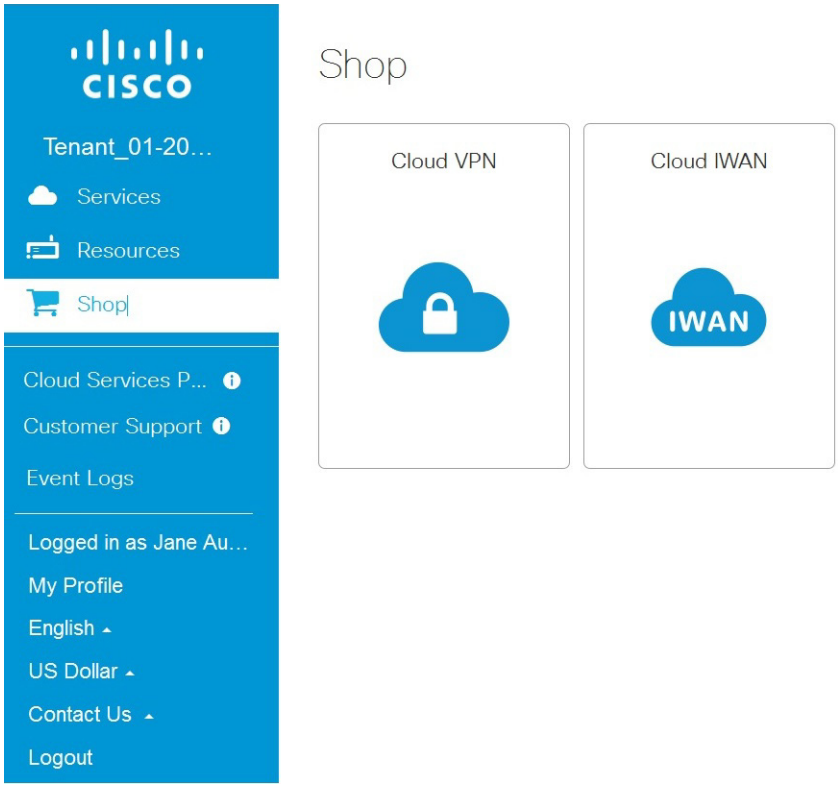
Figure 20 VMS Service Interface: Operator Screen



For tenant/customer accounts, the Service Provider can reduce their operational costs by providing access to the Service Interface self-service user portal, allowing customers to dynamically order and provision their service. The Service Provider has the option of ordering and provisioning services or they can optimize this operation by allowing customers the ability to manage their own Cloud VPN Service. With this option, the customer has the capability to preform the following actions on a VMS Service: create, read, update or downgrade. Figure 21 depicts the interface screen presented to the user, when the customer choose to 'Shop', a list of Service Packages is provided.

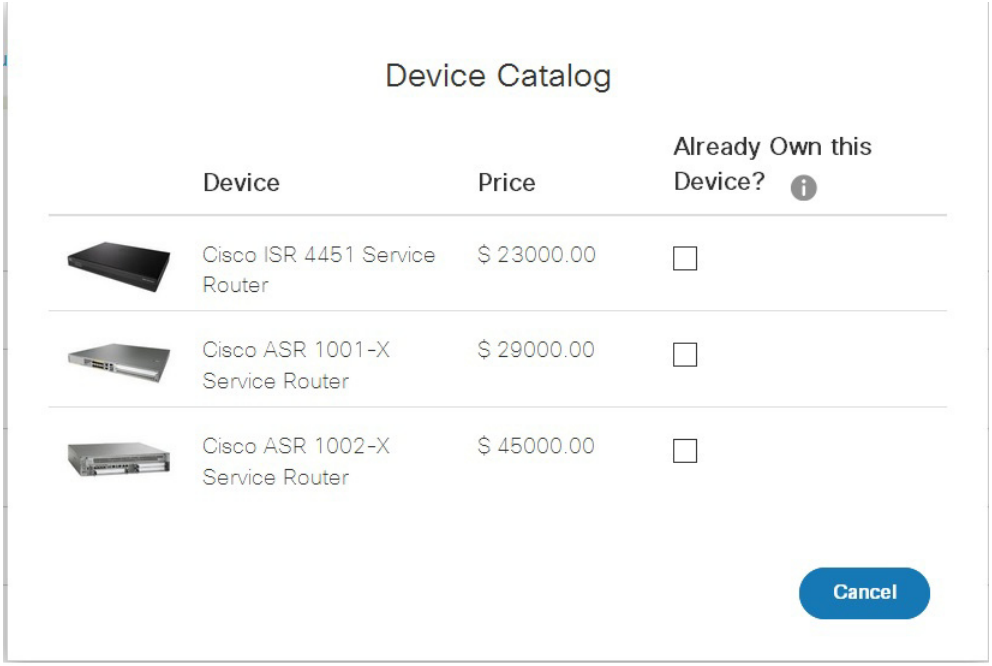
Note: The list of Service Packages is based on the types of services offered by the Service Provider and supported by the VMS release.

Figure 21 VMS Service Interface: Select Service Screen



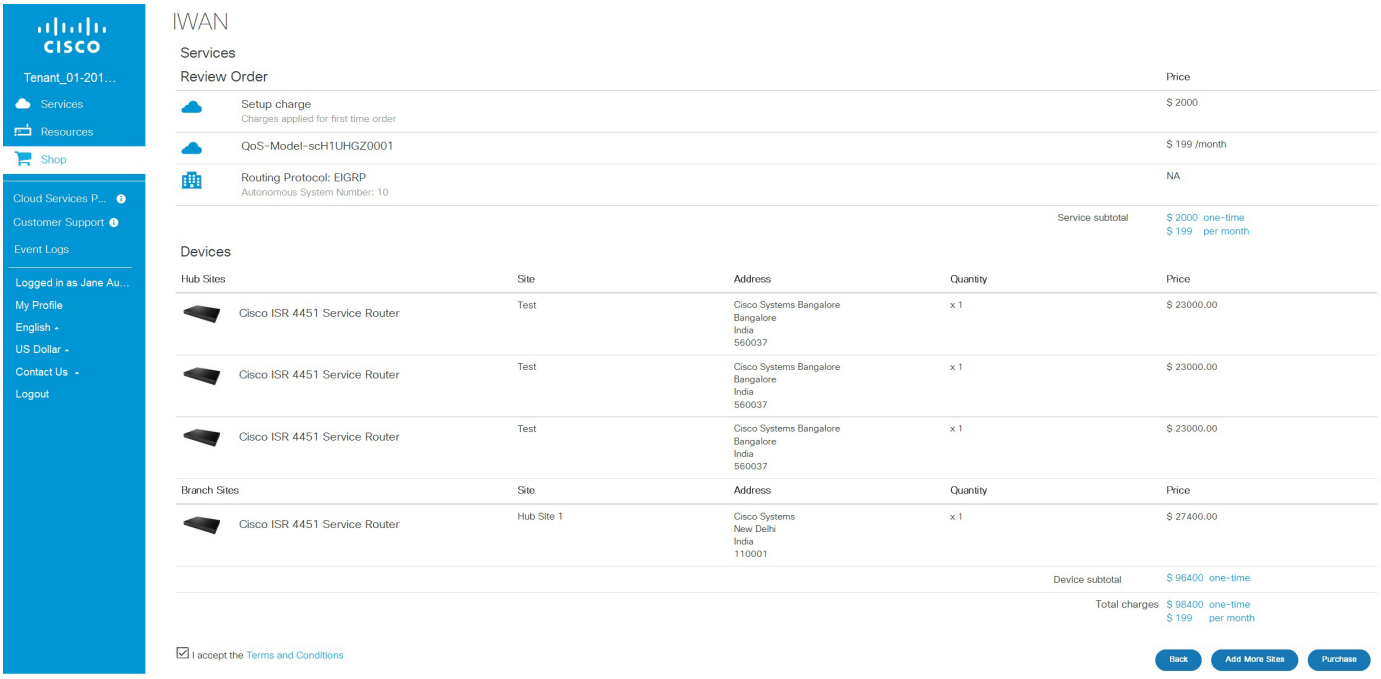
VMS support a Zero Touch Deployment for CPE devices deployment at customer sites. In Figure 22, the user selects the CPE device types for each customer site. For each site defined in the previous stage, the customer will need to add a shipping address to identify the sites to ship the CPE devices. The customer may choose a different device based on locals site requirements.

Figure 22 VMS Service Interface: Select Equipment Screen



Finally, Figure 23 shows the 'Order Summary' for the service the customer just built. The customer will be asked to review the order, followed by an option to purchase.
Note, through this process, the customer has defined the service intent through a series of questions and answers, the next phase will be handled by the Service Interface Back-End, the creation of the service request to the VMS Platform.

Figure 23 VMS Service Interface: Order Summary Screen



Cisco VMS Service Packs

A central theme behind the VMS solution is reducing the operational cost of deploying and maintaining service provider based managed services. The solution shifts the deployment of managed services away from the manual configuration of the latest network devices to the creation of a software abstraction to represent the service definition. This approach allows the service intent of the user, to be realized through the use of service models to automate the creation and customization of SD-WAN services.

VMS is a next generation managed service solution for service providers who are interested in hosting enterprise connectivity and security features in their cloud infrastructure. The keys to the VMS solution are virtualization, plug-n-play CPE devices, and a flexible orchestration engine capable of centralizing the configurations of all the devices involved in the delivery of a service. With flexibility inherent in the orchestration engine, service providers can offer end customers the ability to order the service that best meets their technical and TCO requirements.

The VMS architecture is based on a service package infrastructure. VMS service packages are bundled Virtual Network Function (VNF) types, the type of services available that are tightly coupled to the VNF types included in a specific service package. The end customer, based on a service provider deployment, has a choice based on services that can be orchestrated given the VNF types, which are included in the service package bundle.

This chapter contains the following sections:

Cisco VMS Cloud VPN Service Pack

VMS Cloud VPN Services evolves traditional Managed Services from a transport offer with excessive operational requirements spread across multiple enterprise sites to a next generation cloud based solution delivering deterministic services based on centralized orchestration of all service components. With the enterprise site domain extended into the Service Provider Cloud, VMS creates a single aggregation point where all off site enterprise traffic is processed by a common set of services. Access to these services is possible from anywhere on the Internet footprint. By using secure tunnels over the Internet, not only does VMS Cloud VPN services offer worldwide access, but removes the need to contract for private VPN service.

At the Customer Facing Service (CFS) level, the VMS Service Interface includes a front-end or OSS level module; a component of the Front-End is a self-service user portal. With this portal, Service Providers have the flexibility to allow their customers or an internal sales group, to order, provision and manage the service. The VMS Service Interface includes a Back-End module to convert the service intent or definition, as determined during Front-End user interaction, into a parameterized set of arguments that represent the service request. The service request is passed to the VMS Solution's Resource Facing Service (RFS) layer, specifically the Network Service Orchestrator (NSO), to initiate service orchestration.

When visualizing VMS Cloud VPN services, the concept of an enterprise boundary defined by a traditional DMZ located at the customer site is replaced with a virtual boundary located in the VMS Cloud VPN Service Cloud. With traditional Managed Services, the site DMZ consists of a series of devices or services located locally to secure the site. The site DMZ must be built out and managed at each site, making management repetitive and costly. The VMS Solution replaces all site DMZ constructs with a single virtualized DMZ implemented within the VMS Cloud VPN Service.

The VMS Cloud VPN Service Pack offer the following Cloud VPN services:

- Cloud VPN Foundation
- Cloud VPN Advanced
- Cloud VPN Advanced w/Web Security

CloudVPN Foundation Service

The Foundation Service is the base level VMS Service, through managed CPE devices; enterprise sites communicate through secure IPSec Tunnels to a vRouter located in the Service Provider's Cloud VPN Service's cloud. The enterprise's CPE devices peer directly to the vRouter using an Over-The-Top model. Both CPE devices and the vRouter have public IP addresses, enabling the use of the Internet as the transport. All site-to-site traffic traverses the vRouter.

Traffic destined to Internet sites, not associated with the SP Cloud VPN service cloud are routed directly to their destination via a "split tunnel" mechanism configured on the CPE device.

CloudVPN Advanced Service

The Advanced Service extends the Foundation Service by providing Internet access in the Cloud VPN Service's cloud. Site-to-site traffic is treated in the same fashion as in the Foundation Service, however traffic destined to general Internet sites utilize a virtual Firewall and NAT located in the Cloud VPN Service's cloud. With this service level, all service access policies are located in a central location, ensuring all users, independent of site location, share a common Firewall service policy and NAT mechanism.

The Advance Service also includes remote user access for mobile or remote workers. Using Cisco AnyConnect, the remote user can access enterprise resources globally through the public Internet.

The end-customer can add an optional Intrusion Prevention Service based on the Service Packages offered by the Service Provider

CloudVPN Advanced w/Web Security Service

The Advanced w/Web Security Service extends the Advanced Service by additional filter granularity through the insertion of Web Security functionality, allowing enterprise greater control over HTTP based traffic destined or received from the Internet.

The end-customer can add an option Intrusion Prevention Service based on the Service Packages offered by the Service Provider.

Cisco VMS Virtual Converged Edge (vCE) Service Pack

As a service provider, you may have customers with remote branches that utilize different WAN access methods (such as Cloud VPN and MPLS). The Virtual Converged Edge (VCE) attachment circuit interface, which is available as an add-on transport option when ordering a Cloud VPN service-level package, enables those branches to communicate and share a common service chain. The VCE service targets the Cisco Cloud Services Router (CSR) 1000V associated with a service chain.

The following list summarizes the benefits that the VCE attachment circuit provides:

- Integration with Cloud VPN service-level packages
- Extension of secure Internet and other NFV services to customers on MPLS networks
- Integration of remote branches and datacenters across Cloud VPN and MPLS networks
- eBGP routing support
- VLAN manual handoff

- VCE data collection
- Graphical user interface for VCE ordering, peering configuration, and metrics

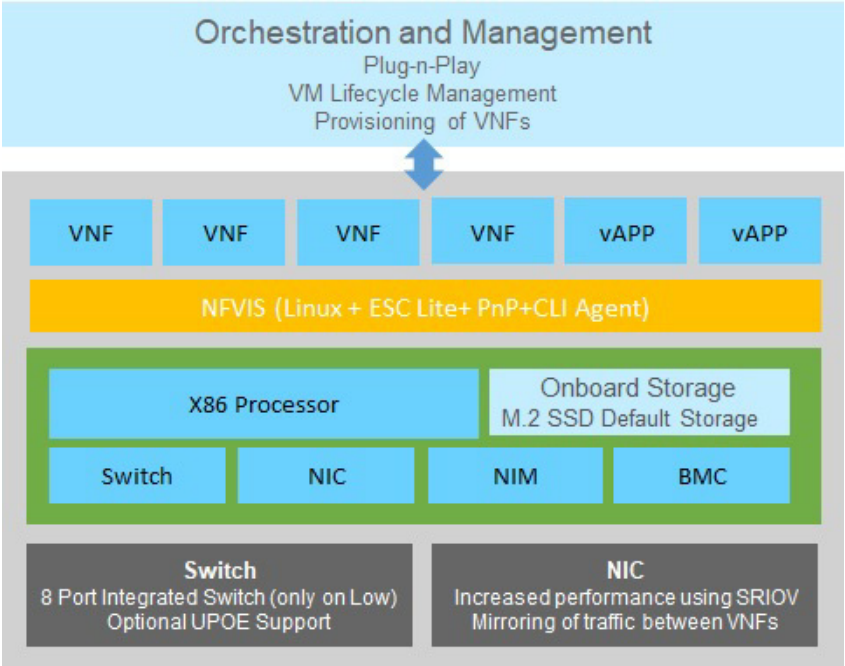
Cisco VMS vBranch Service Pack

Cisco VMS vBranch service pack enables unified routing, switching, storage, processing, and a host of other computing and networking activities into a into a single box. The vBranch service pack provides a way to collapse the services that a branch requires into a single box, which results in easier management of services, and smaller device footprint on a branch site.

The VMS vBranch service pack includes the following:

- An orchestration environment to allow automation of virtualized network service deployment, consisting of multiple Virtualized Network Functions (VNF).
- VNFs, which provide the desired network functionality, or even non-networking software applications, required at a deployment location.
- The NFV Infrastructure Software platform to facilitate the deployment and operation of VNFs and hardware components.

The figure below illustrates the functional architecture of a vBranch site.



Some of the advantages of the VMS vBranch service pack are:

- Zero touch provisioning for initial device connectivity through PnP server processes.
- Service provisioning of on-premise CPEs through orchestration.
- User interface portal for ordering service, network visualization, and performance or fault monitoring.
- Lifecycle Management.

In the VMS vBranch 3.1, VMS vBranch supports the branch site on Cisco 5000 Enterprise Network Compute System (ENCS) platform.

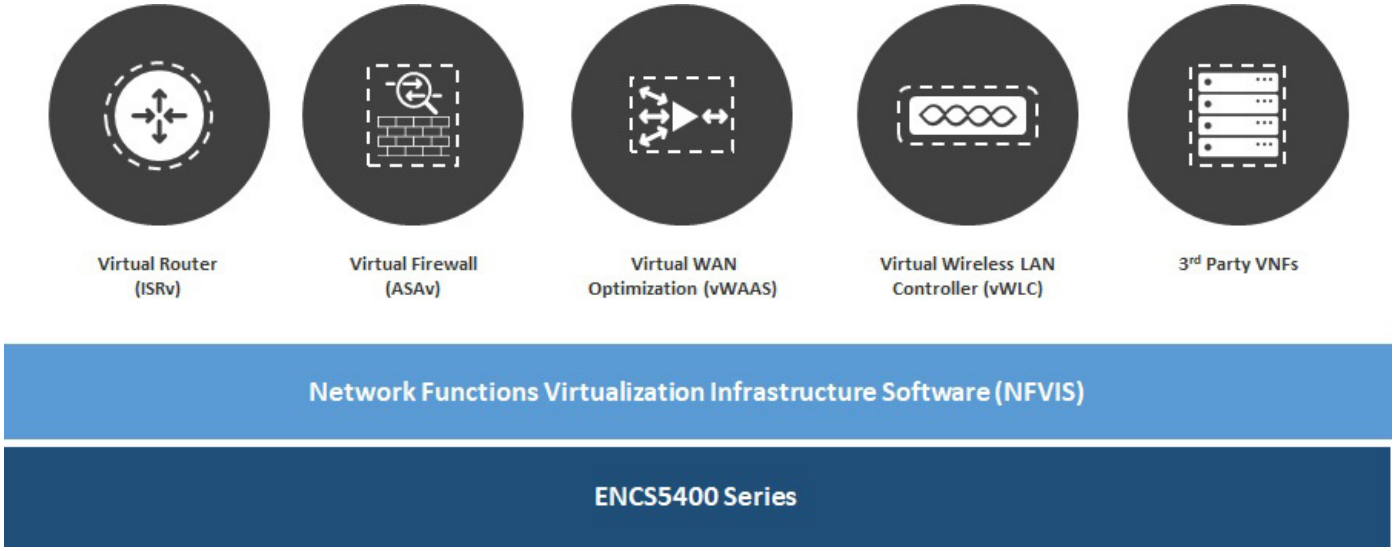
Cisco Enterprise Network Compute System

The Cisco 5000 Enterprise Network Compute System (ENCS) is a line of compute appliances designed for the Cisco Enterprise Network Functions Virtualization (ENFV) solution. It delivers a new standard of software-defined flexibility and performance, and offers a low Total Cost of Ownership (TCO). The 5000 ENCS is a hybrid platform that combines the best attributes of a traditional router and a traditional server, and offers the same functionality with a smaller infrastructure footprint. Offered with the Cisco Integrated Services Virtual Router (ISRv) and NFV Infrastructure Software (NFVIS) as the hosting layer, the platform offers a complete solution for a simplified deployment.

Supported VMs

Currently, the following Cisco supplied VMs and third party VMs are supported on Cisco ENCS:

- **Cisco Integrated Services Virtual Router (ISRv)–**A virtual form-factor of the Cisco IOS XE software router that delivers WAN gateway and network services functions into virtual environments.
- **Cisco Adaptive Security Virtual Appliance (ASAv)–**Enables ASA firewall and VPN capabilities on virtualized environments to safeguard traffic and multitenant architectures. Optimized for data center deployments, it is designed to work in multiple hypervisor environments, reduce administrative overhead, and increase operational efficiency.
- **Cisco Virtual Wide Area Application Services (vWAAS)–**A virtual appliance that accelerates business applications delivered from private and virtual private cloud infrastructure. Cisco vWAAS enables you to rapidly create WAN optimization services with minimal network configuration or disruption.
- **Virtual Wireless LAN Controller (vWLC)–**Virtual form-factor controller for any x86 server with VMware Hypervisor ESXi**.**
- **3rd Party VNFs–**Third party VNFs.



Cisco VMS SDWAN Service Pack

Cisco SDWAN service pack enables service providers to deploy and manage SDWAN service for their customers. The deployment of an SDWAN service in the context of a managed service requires deployment per customer and includes the SDWAN management control plane (vManage, vBond and vSmart), and the corresponding data plane (vEdge).

The SDWAN service pack management control plane and data plane consists of:

- **vManage**—The vManage is a centralized dashboard that enables automatic configuration, management, and monitoring of the overlay network. Users login to vManage to centrally manage all aspects of the network life cycle—from initial deployment, on-going monitoring and troubleshooting, to change control and software upgrades.
- **vBond**—The vBond facilitates the initial bring-up by performing initial authentication and authorization of all elements into the network. vBond provides the information on how each of the components connects to other components. It plays an important role in enabling devices that sit behind the NAT to communicate with the network.
- **vSmart Controller**—The vSmart controllers establish the secure SSL connections to all other components in the network, and run an Overlay Management Protocol (OMP) to exchange routing, security, and policy information. The centralized policy engine in vSmart provides policy constructs to manipulate routing information, access control, segmentation, extranets, and service chaining.
- **vEdge Router**—The vEdge router establishes secure connectivity to all of the control components and also establishes IPSec sessions with other vEdge routers in the WAN network. In the VMS SDWAN 3.1.1, you can deploy a customer site on Cisco 5000 Enterprise Network Compute System (ENCS) platform.
  The Cisco 5000 Enterprise Network Compute System (ENCS) is a line of compute appliances designed for the Cisco Enterprise Network Functions Virtualization (ENFV) solution. Cisco 5000 ENCS is a hybrid platform that combines the best attributes of a traditional router and a traditional server, and offers the same functionality with a smaller infrastructure footprint.

Some of the advantages of the VMS SDWAN service pack are:

- Provides the interface to associate the tenant (customer) with the Control Plane and Data Plane.
- User interface portal for ordering service (Control Plane and Data Plane Connectivity) and network visualization.
- Lifecycle management of services.

Cisco VMS Managed Device Service Pack

Cisco VMS Managed Device service pack enables service providers to provide their customers manage devices services through a self-service portal. With Managed Device service pack, IT organizations can bring into its network (on-board) devices located at the customer premise (CPEs) and apply or manage configuration settings remotely from its Network Operations Center (NOC). The service provider can configure parameterized configuration template that need to be deployed on these CPEs.

VMS Managed Device service pack makes device deployment fast and easy. Using this service pack user interface you can configure and deploy a VMS CPEs.

- Some of the advantages are as follows:
- Zero touch provisioning for initial device connectivity through PnP server processes.
- Service provisioning of on-premise routers through NSO orchestration.
- User Interface portal for configuration templates, ordering service, and performance or fault monitoring.

Cisco 4000 Series Integrated Services Routers

Cisco VMS Managed Device, supports ISR 4k series. The Cisco 4000 Series Integrated Services Routers (ISR) revolutionize WAN communications in the enterprise branch. With new levels of built-in intelligent network capabilities and convergence, the routers specifically address the growing need for application-aware networking in distributed enterprise sites. These locations tend to have lean IT resources. But they often also have a growing need for direct communication with both private data centers and public clouds across diverse links, including Multiprotocol Label Switching (MPLS) VPNs and the Internet.

Cisco VMS Service Extensions

This chapter provides an overview of VMS service extensions and contains the following topics:

Understanding How Cisco VMS Service Extensions Work

VMS service extensions simplifies how configuration snippets can be applied to a service or a device. VMS leverages the underlying capability of Cisco Network Services Orchestrator (NSO) custom templates, which get pushed along with the derived configuration templates. Service extensions can be used, in most cases, to map services to device configurations, without the need for any additional programming.

VMS service extension templates use variables to map service attributes to the corresponding device configurations and are applied. The service extensions allow a declarative way to describe such manipulations. The VMS operator can apply a service extension template to an existing service chain in VMS or to a device, without having to manually go into the NSO CLI. This service extension template is used by NSO to add, modify, or delete service configuration snippets before NSO pushes the configuration to the devices.

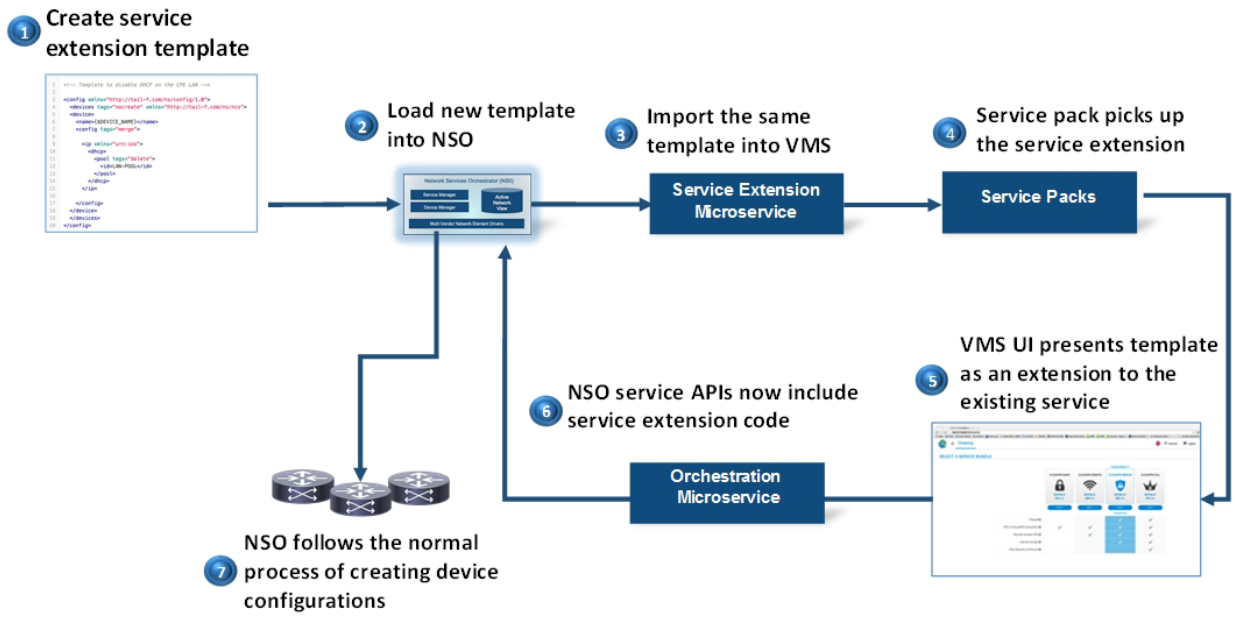
You can apply VMS service extension templates to a service ordering workflow or a single device. When you import a service extension template into VMS, you can specify if the template is to be applied to a service workflow or a device.

When a service extension template is applied to service ordering workflow, VMS service workflow gathers the parameter values the tenant users enter during service ordering process. These values are passed to NSO, which further uses these values in the device configurations.

When a service extension template is applied to a device after the service is ordered, the service extension template is applied to the device outside the workflow, and is available for use for individual devices.

The illustration below depicts the end-to-end workflow that needs to be followed to work with service extension templates in VMS.

Figure 24 VMS Service Extensions Workflow



Creating a VMS Service Extension Template XML File

The VMS service extension template is an XML file. The structure of that file is defined by the YANG model.

The basic principles of defining a template are as following:

1. A template is an XML file (for example *mytemplate.xml*) that corresponds to a node in the device tree.
2. Each value in a template is stored as a string. This string value is converted to the actual value type of the YANG model when the template is applied.
3. The value part of the XML tag that needs to be configured must be represented with a variable name prefixed with '$' literal.
4. The templates allow for defining different behavior while applying the template. This is accomplished by setting tags such as `merge, replace, delete, create or nocreate` on the relevant nodes in the template.

For example, to create a template to set the NTP server on a device, the sample template XML file should be:

```
<config
    xmlns="http://tail-f.com/ns/config/1.0">
    <device
        xmlns="http://tail-f.com/ns/ncs">
        <name>ntp</name>
        <config tags="merge">
            <system xmlns="http://pica8.org/yang">
                <ntp-server-ip>{$NTP}</ntp-server-ip>
            </system>
```

```
            <ntp xmlns="urn:ios">
                <server>
                    <server-list>
                        <ip-address>{$NTP}</ip-address>
                    </server-list>
                </server>
            </ntp>
        </config>
    </devices>
</config>
```

After you create a service extension template, you need to do the following:

- Import the template XML file into NSO. For details see, the service pack guides.
- Import the template XML file into VMS. For details see, the service pack guides.

Terminology

The table below provides an alphabetical listing of VMS acronyms.

Table 1 Terminology

Acronym	Definition
AAA	Authentication, Authorization, and Accounting
ACL	Access List
API	Application Programmable Interface
ARPU	Average Revenue Per User
ASAv	Adaptive Security Appliance - vFW VNF
BGP	Border Gateway Protocol
BSS	Business Support System
CDB	Configuration Database
CFS	Consumer Facing Service
CLI	Configuration Line Interface
CMSP	Cisco Cloud and Managed Service Program
CPE	Customer Premise Equipment (ISR-G2)
CRUD	Create, Read, Update and Delete orchestration operations
CSR	Cloud Service Router - vRouter VNF
cURL	Client URL
DC	Data Center
DCI	Data Center Interconnect
DEST	Destination IP Address
DHCP	Dynamic Host Configuration Protocol
DMVPN	Dynamic Multi-Point Virtual Private Network
DMZ	Demilitarized Zone (Networking private to public)
DNS	Domain Name System
DST	Destination IP Address
EIGRP	Enhanced Interior Gateway Routing Protocol
ESC	Elastic Services Controller
ETS	European Telecommunications Standards
ETSI	European Telecommunications Standards Institute
EzVPN	Easy Virtual Private Network
FW	Firewall
GraphDB	Graph Data Base
GuestOS	Guest Operating System

Table 1 Terminology

Acronym	Definition
HA	High Availability is
HDR	Header
HTTP	Hypertext Transfer Protocol
I2RS	Interface to Routing System (diagram format)
IKEv2	Internet Key Exchange version 2
IO	Input/Output
Intrusion Detection	Generally refers to the process of passively analyzing network traffic for potential intrusions and storing attack data for security analysis.
Intrusion Prevention	Includes the concept of intrusion detection, but adds the ability to block or alter malicious traffic as it travels across your network
iOS	Cisco Operating System
IP	Internet Protocol
IPSec	Internet Protocol Secure
ISR	Integrated Service Router
IT	Information Technology
KVM	Kernel-based Virtual Machine
LAN	Local Area Network
ML2	Modular Layer 2
MPLS	Multiprotocol Label Switching
NAT	Network Address Translation
NAT44	Network Address Translation IPv4-to-IPv4
NED	Network Element Driver
NEDs	Network Element Drivers
NETCONF	Network Configuration protocol
NFV	Network Functions Virtualization
NH	Next Hope
NIC	Network Interface Card
NICs	Network Interface Cards
NSO	Network Service Orchestrator
ODL	Open Daylight
OS	Operating System
OSPF	Open Shortest Path First
OSS	Operations Support System
OVS	Open Virtual Switch
PE	Provider Edge
PnP	Plug-N-Play
QEMU	Quick Emulator
QoS	Quality of Service
RA	Remote Access
REST	Representation State Transfer

Table 1 Terminology

Acronym	Definition
RFC	Request For Comments
RFS	Resource Facing Service
ROI	Return on Investment
SA	Source Address
SDN	Solution uses software-defined networking
SLA	Service Level Agreement
SMB	Small Medium Business
SNMP	Simple Network Management Protocol
SP	Service Provider
SRC	Source IP Address
SSH	Secure Shell
SSL	Secure Sockets layer
SUDI	Secure Unique Device Identifier
TCO	Total Cost of Ownership
TP	Termination Point
UCS	Unified Computing System
vFW	Virtual Firewall
VIM	Virtual Interface Manager
VirtIO	Virtual Input/Output
VLAN	Virtual Local Area Network
VM	Virtual Machine
VMs	Virtual Machines
VMS	virtual Manage Services
VNF	Virtual Network Function
VNFs	.Virtual Network Functions
VNIC	Virtual Network Interface
VPN	Virtual Private Network
VPP	Vector Packet Parsing
VR	Virtual Router
VRF	Virtual Route Forwarding
VRRP	Virtual Registry Registrar Protocol
WAN	Wide Area Network
WCCPv2	Web Cache Communications Protocol
WSA	Web Security Appliance
WSAv	Web Security Appliance virtualized
XML	eXtensible Marking Language
YANG	Yet Another Next Generation (data modeling)
ZTD	Zero Touch Deployment