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:

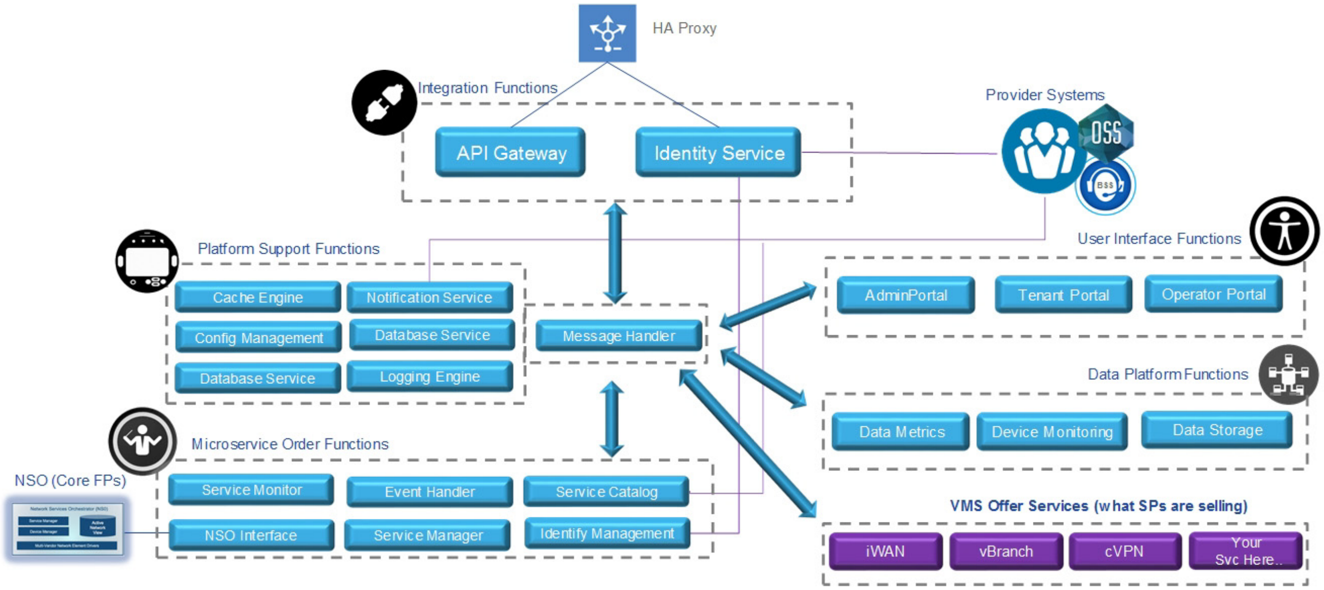
- VMS Platform
- VMS Service Packs

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



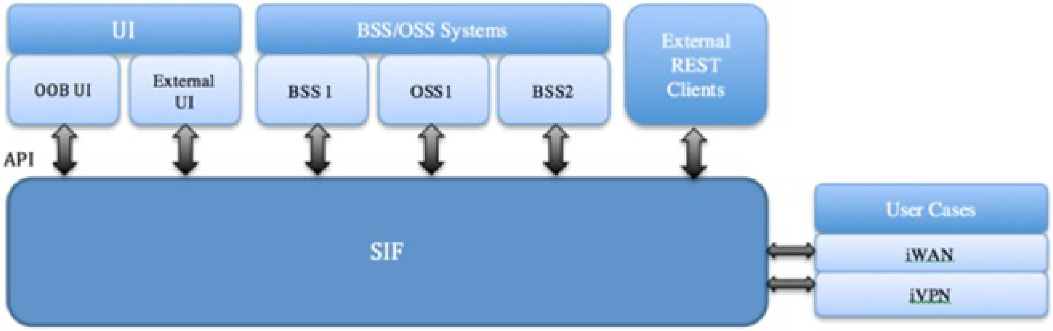
The key capabilities of VMS solution are:

- 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

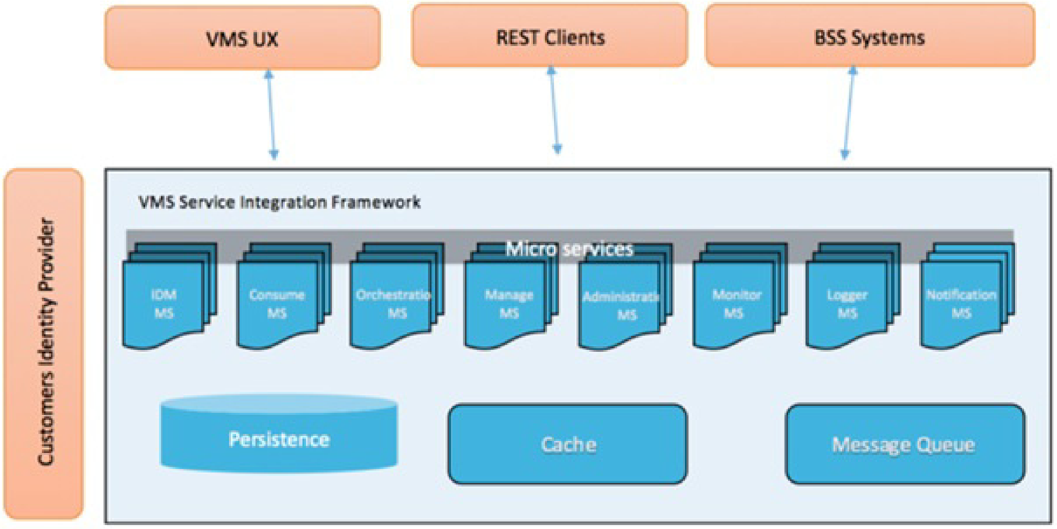
All of SIF functions and integrations are enabled as microservices that contain the logic to consume the functions exposed by the platform and use cases. There are a standard set of microservices that consume the base platform functionality required to operate the platform.

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



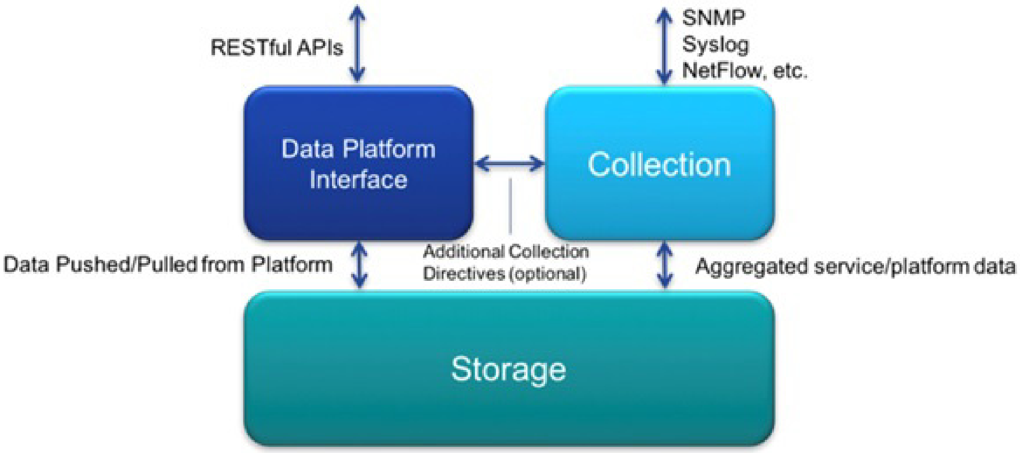
Data Platform

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


RESTful APIs
SNMP
Syslog
NetFlow, etc.
Data Platform Interface
Collection
Data Pushed/Pulled from Platform
Additional Collection Directives (optional)
Aggregated service/platform data
Storage

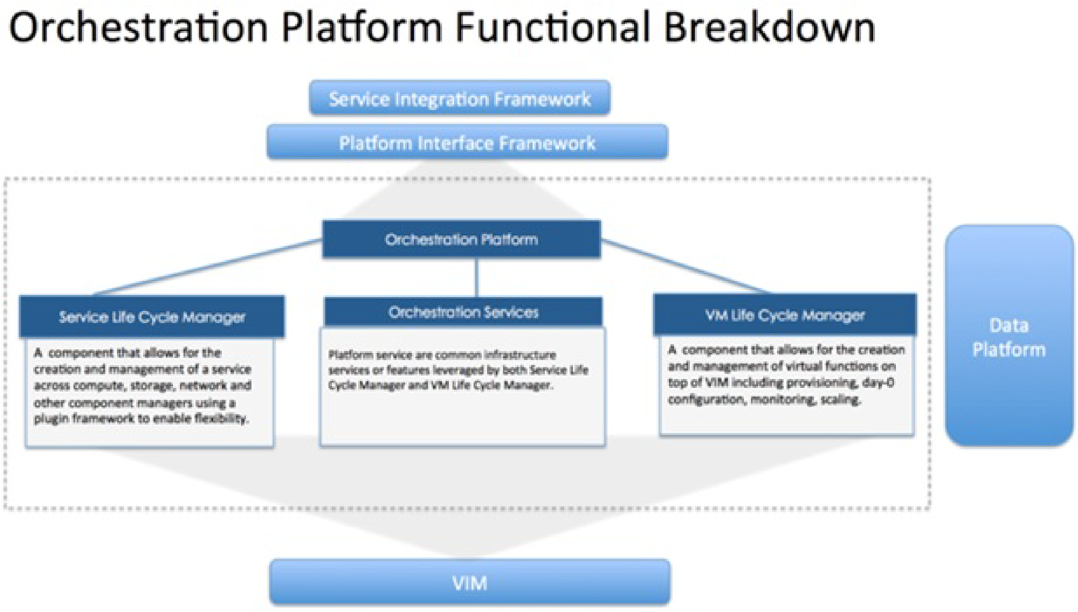
The major function performed by the data platform component is the collection of data. The collection interface allows a platform component or application the ability to instruct the data platform which data to collect, the frequency with this to collect, and how to store and correlate the data. This 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


Orchestration Platform Functional Breakdown
Service Integration Framework
Platform Interface Framework
Orchestration Platform
Service Life Cycle Manager
Orchestration Services
VM Life Cycle Manager
Data Platform
VIM

A component that allows for the creation and management of a service across compute, storage, network and other component managers using a plugin framework to enable flexibility.

Platform service are common infrastructure services or features leveraged by both Service Life Cycle Manager and VM Life Cycle Manager.

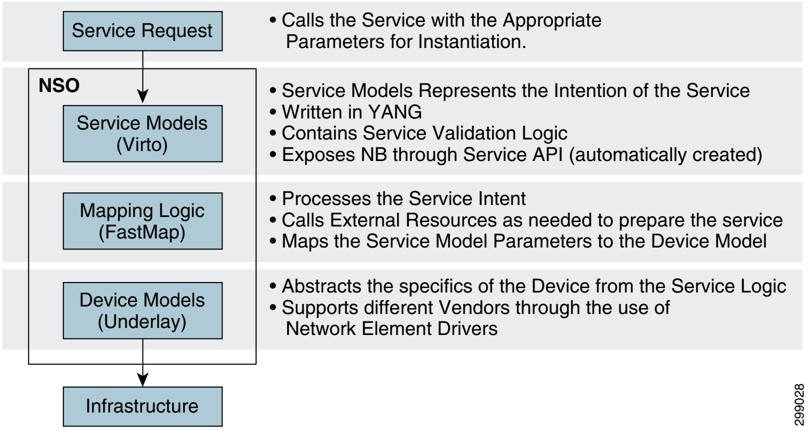
A component that allows for the creation and management of virtual functions on top of VIM including provisioning, day-0 configuration, monitoring, scaling.

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


Service Request
• Calls the Service with the Appropriate Parameters for Instantiation.
NSO
Service Models (Virto)
• Service Models Represents the Intention of the Service
• Written in YANG
• Contains Service Validation Logic
• Exposes NB through Service API (automatically created)
Mapping Logic (FastMap)
• Processes the Service Intent
• Calls External Resources as needed to prepare the service
• Maps the Service Model Parameters to the Device Model
Device Models (Underlay)
• Abstracts the specifics of the Device from the Service Logic
• Supports different Vendors through the use of Network Element Drivers
Infrastructure
299028

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.