



## CHAPTER 7

# VMS Service Instantiation

---

The key differentiator of 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 Interface 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.

## Service Blueprints

Cloud VPN 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 function pack.

A software function pack lives in the Cisco NSO software modules and has several subcomponents as shown in [Figure 7-1](#) 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 function 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’.

## Function Pack Fundamentals

VMS function 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 exposed API created as part of the software function 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.

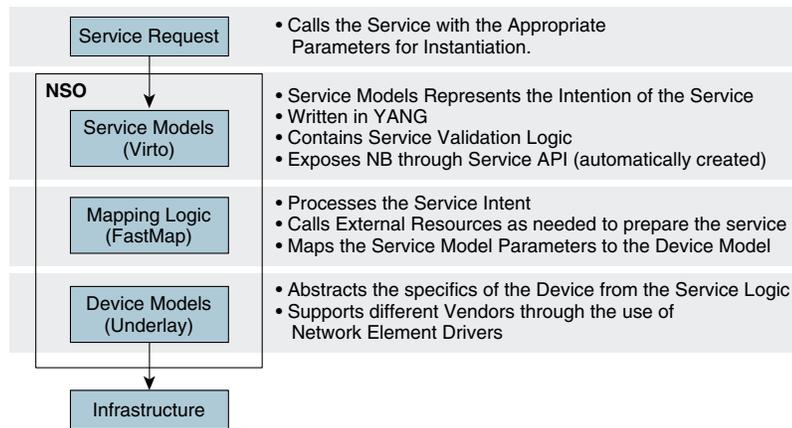
VMS 2.0 includes three Service Packages (function packs):

- Foundation, includes:
  - vRouter VNF type
  - Foundation Service Model with associated Re-Active/Fast-Map Code
- Advanced, includes:
  - vRouter, vFW, & vWeb SecurityVNF types

- Foundation Service Model with associated Re-Active/Fast-Map Code
- Advanced Service Model with associated Re-Active/Fast-Map Code
- Advanced w/Web Security with associated Re-Active/Fast-Map Code

Figure 7-1 shows a detailed view of the Function 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 ‘Virtio’. The Yang service model software has validation logic that validates the service requests to ensure that incorrect service requests are not completed.

**Figure 7-1 Function 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.

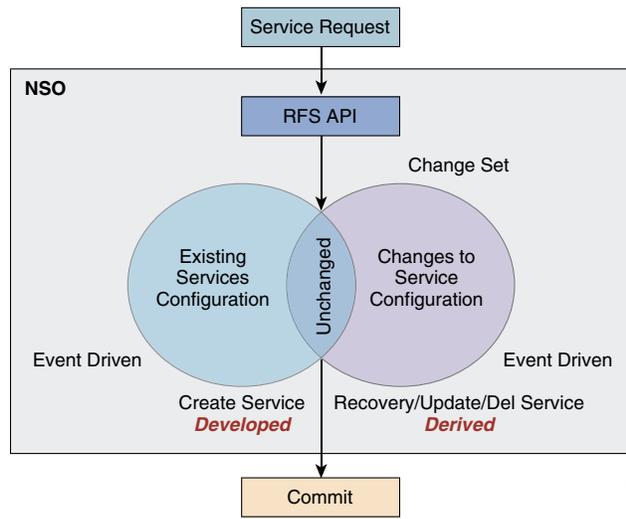
Device models also use Yang modeling constructs but are intended to model the infrastructure rather than the service. This set of device models is what is referred to as the underlay and is an abstraction of only the physical and virtual infrastructure that must be configured to enable the requested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

## 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 7-2, 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 7-2 Cisco NSO Service Request Functionality



## Service Device Mapping

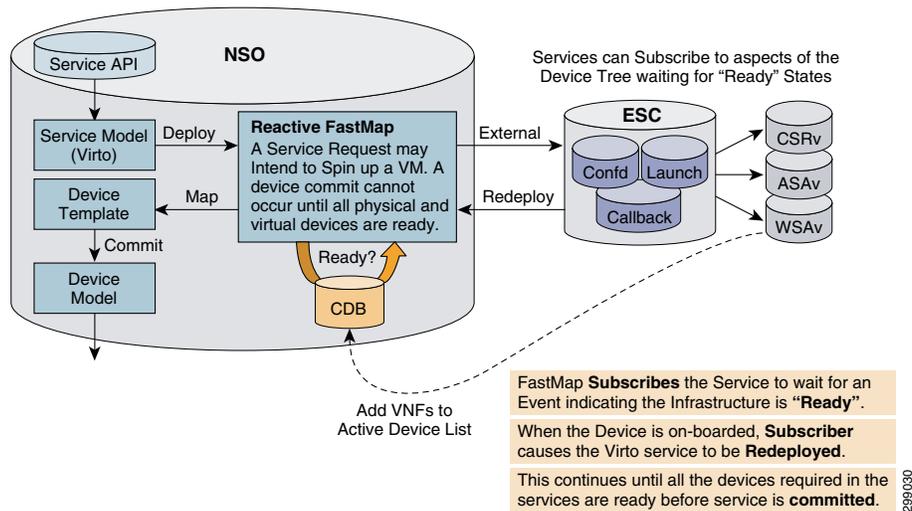
The purpose of the Fastmap software in the function pack of a service blueprint is to map service intent to the device infrastructure. The Fastmap software process uses Java logic resulting in the creation of a service template that is mapped to a device model. This process works very well for physical 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 7-3](#).

**Figure 7-3 Mapping Services to Non-Existing Infrastructure, Innovative Fastmap Functionality**



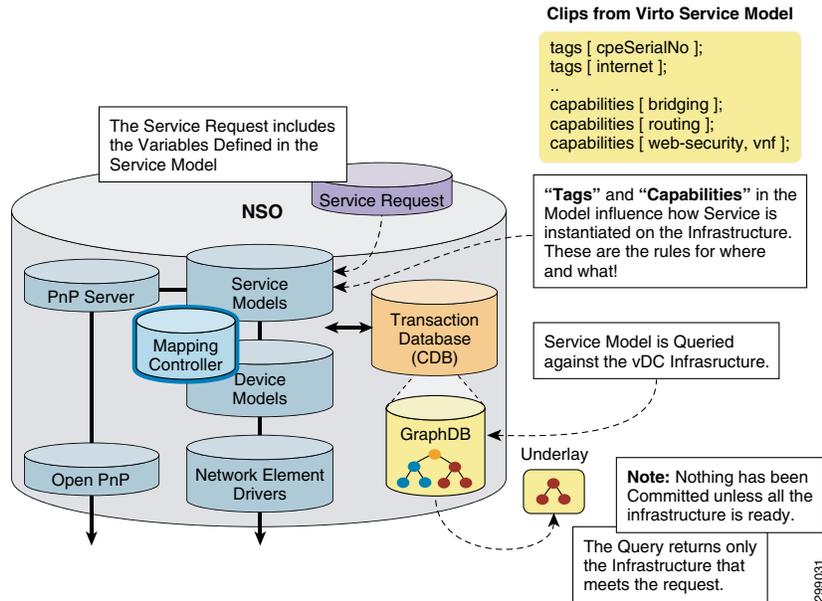
## Configuring Infrastructure Elements

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 infrastructure 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 7-4](#).

**Figure 7-4 Mapping Service Intent to Specific Topology, Querying the Underlay**

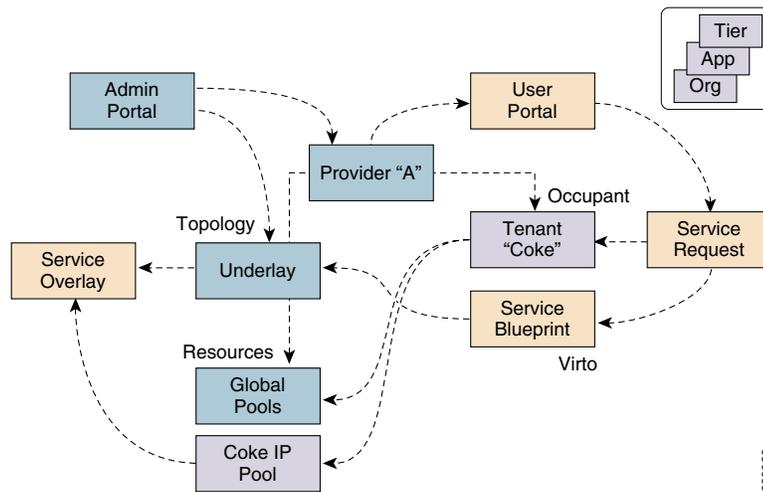


## Allocating Resources

Once an infrastructure underlay has been loaded into the Cisco NSO software, it can be allocated to a specific provider. An underlay configuration can only be allocated to a particular provider at a time. Resource Pools are Cisco NSO elements that are used to create the are part of that underlay architecture configurations. Resource pools are required by Cisco NSO orchestration software components such as the Plug-n-Play server, CPEs, Compute, and the Bridging functions. Global pools from the Underlay are used by the service pProvider to assign IP addressing to these components. These pools can then be further refined to on-boarded Tenants.

Associating Provider underlay resources and Tenants with service definition models for each tenant, Services is the role of the Occupancy model software process. As already discussed, Resources in the underlay are configured into NSO and allocated to a Provider instance. The Occupancy model is the manner by which multiple customers can then be allocated across those pools of service provider resources. The resource allocation service model is shown in [Figure 7-5](#).

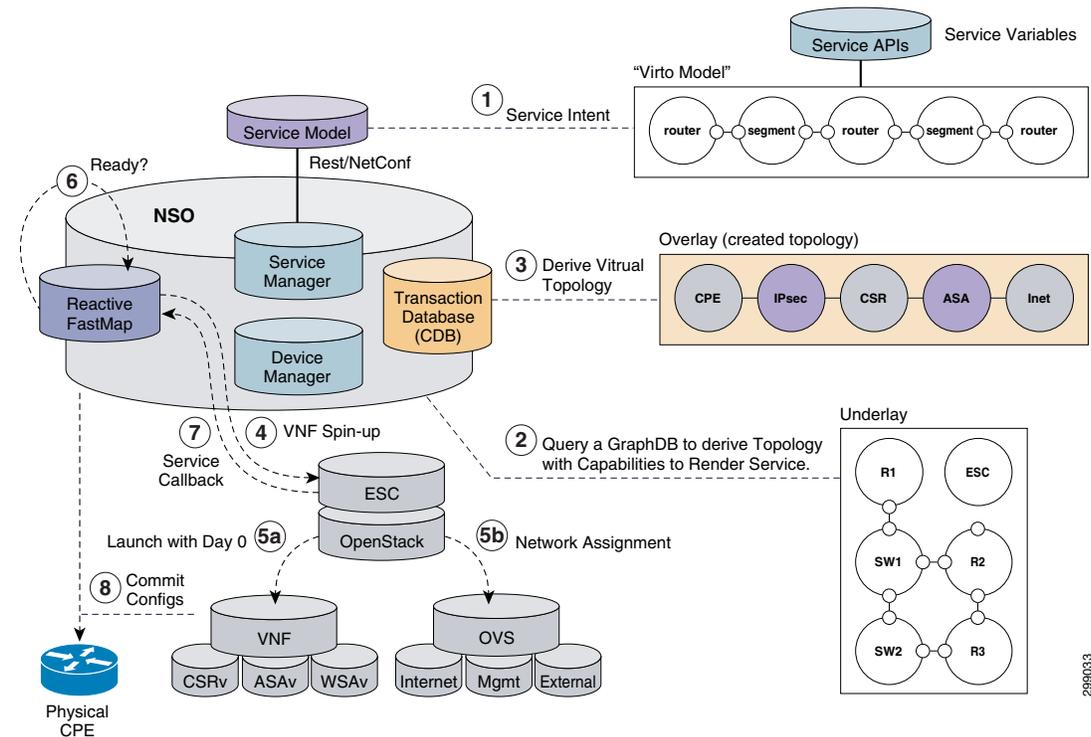
Figure 7-5 Cisco NCO, Occupancy Model View



## Putting the Pieces Together

Figure 7-6 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 Function 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 7-6 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).

## VMS 2.0 Service Details

VMS 2.0 is based on the services introduced in VMS 1.0, this document covers the options added to these VMS 1.0 Services:

- Foundation
- Advanced
- Advanced w/Web Security

Refer to [VMS 1.0 Design Overview and Introduction](#) documentation for service details on the following services and their instantiation:

### Cloud VPN Service Instantiation

Instantiation of a Cloud VPN service commences at with a customer ordering a service. Since service orchestration, based on pre-defined service models, is fundamental to the VMS approach to service instantiation, provisioning, of a service commences immediately once a customer service-request is received by the VMS platform. The following sections segment review the various segments that are involved in the activation of a service based which is generate at the top most layer of this modular and loosely coupled service instantiation process.

## Cloud VPN Service Segmentation

Provisioning of a VMS service involves multiple activities across multiple locations of both physical and virtual resources. [Figure 7-7](#) segments the service acquisition into a series of activities and components in an effort to illustrate the scope of the orchestration challenge.

### 1. Segments

#### a. CFS Layer

The Customer Facing Service segments is where the OSS interacts with the VMS Platform, VMS 2.0 includes a Service Interface or user portal, which based on the VMS Service Package, is designed to allow enterprise users to self-order services and associated options inclusive in the service package.

The Service Interface consists of two parts: a User Interface and a User Backend. The User Interface is graphical interface, designed to allow the end-user the ability to self-order a service, with configurable options, define number of enterprise sites, and have the CPE shipped to each designated site.

The Service Interface User Backend processes the user's options, capturing their service-intent in a service request that is sent to the RFS Layer API. Alternatively, the Service Provider may utilize their OSS to determine service-intent and create a service request to transmit to the RFS Layer API.

#### b. RFS Layer

The Resource Facing Service segment receives the service request from the CFS Layer and is responsible for provisioning the resources to provision the service. Management of resources and provisioning of the service request is performed by the VMS Platform, which utilizes a service-model based orchestration model to identify and provision the resources required to deploy the service defined by the service request. The VMS Platform consists of three major components:

- **NSO**—Network Service Orchestrator
- **ESC**—Elastic Service Controller
- **VIM**—Virtual Infrastructure Manager

#### c. Cloud VPN Services

The Cloud VPN Services segment contains the virtualized service components, hosted in the Service Provider controlled virtualization-infrastructure cloud, that when linked form the base functionality of the service-request initiated in the CFS Layer. Each link segment or service-chain is provisioned through the processing of the service-request originating at the CFS Layer and orchestrated by the RFS Layer. Each service-chain is dedicated to enterprise requesting the service, traffic traversing the Cloud VPN service chain isolated by the dedicated resources and address space.

VMS Service Packages are constructed based on the type of VNFs required to deploy specific service types. The types of Cloud VPN services available will be dependent on the Service Package.

#### d. Service Access

The Service Access segment defines the service access transport and is used provide a direct path from the enterprise CPE to the Cloud VPN service access point or vRouter. VMS 2.0 creates a secure between the CPE and vRouter using a Flex-VPN based IPsec tunnel. For remote users, the service access is defined by an IPsec Tunnel created using an AnyConnect client to the Remote VPN Service located on the vFW,

e. Enterprise Access

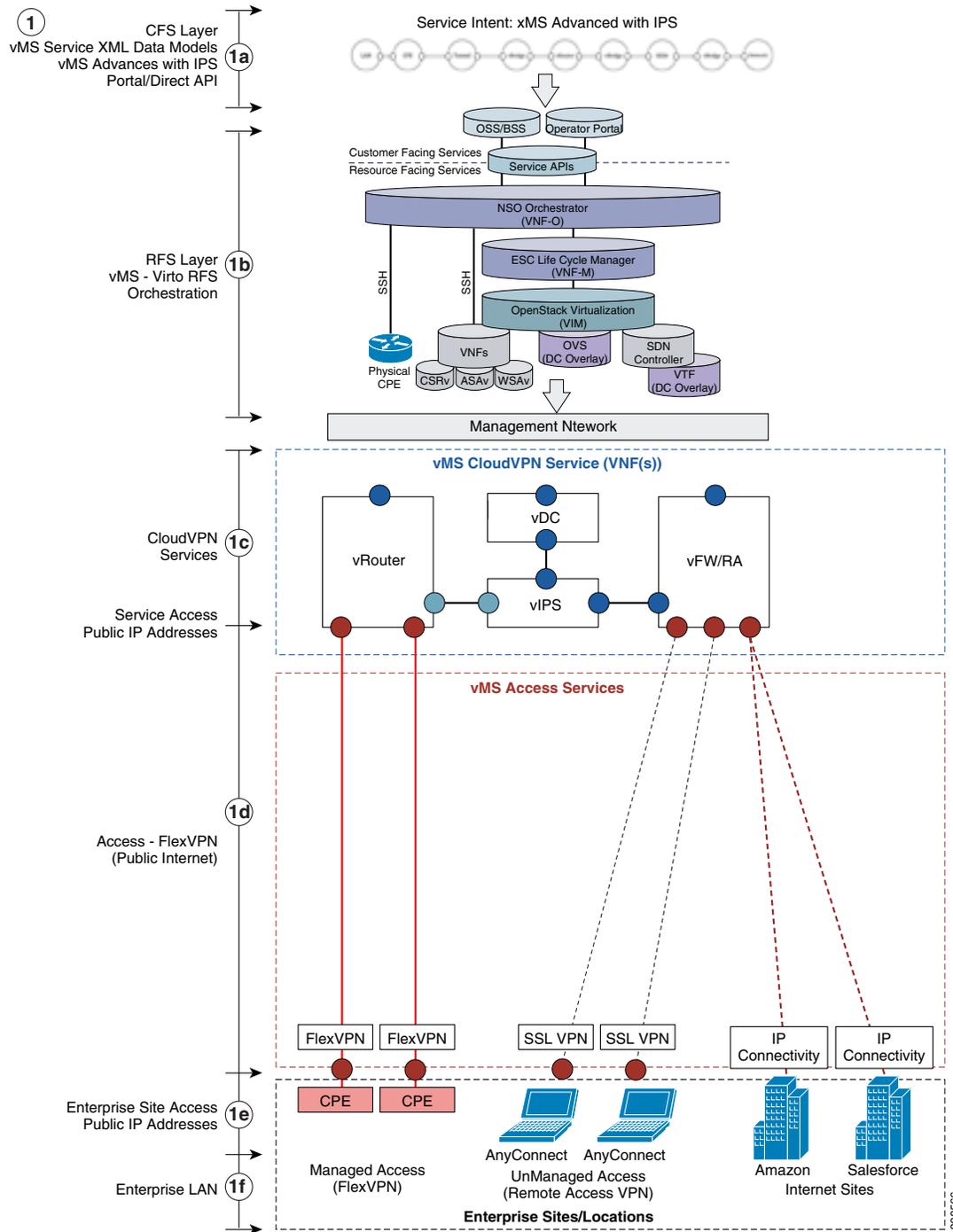
The Enterprise Access segment consists of CPE devices located at the enterprise sites and remote user clients that are part of the Internet cloud. As part of the VMS Service, the CPE device is shipped to the enterprise site and when plugged-in, at this point a Zero-Touch-Deployment mechanism configures the device; the only requirement is IP connectivity on the Internet.

Once the CPE device is connected at the enterprise site, the device, using a ZSTD model registers with VMS through a Plug-n-Play mechanism that is initiated through a Public IP address. Once registered, through the use of serial number ID of the CPE device, both system and service configuration occurs through the NSO module.

f. Enterprise LAN

The Enterprise LAN segment is not part of VMS domain, however VMS services will route traffic to/from the address space.

Figure 7-7 Cloud VPN Service Segmentation



299566

