Ultra Services Platform (USP) Introduction

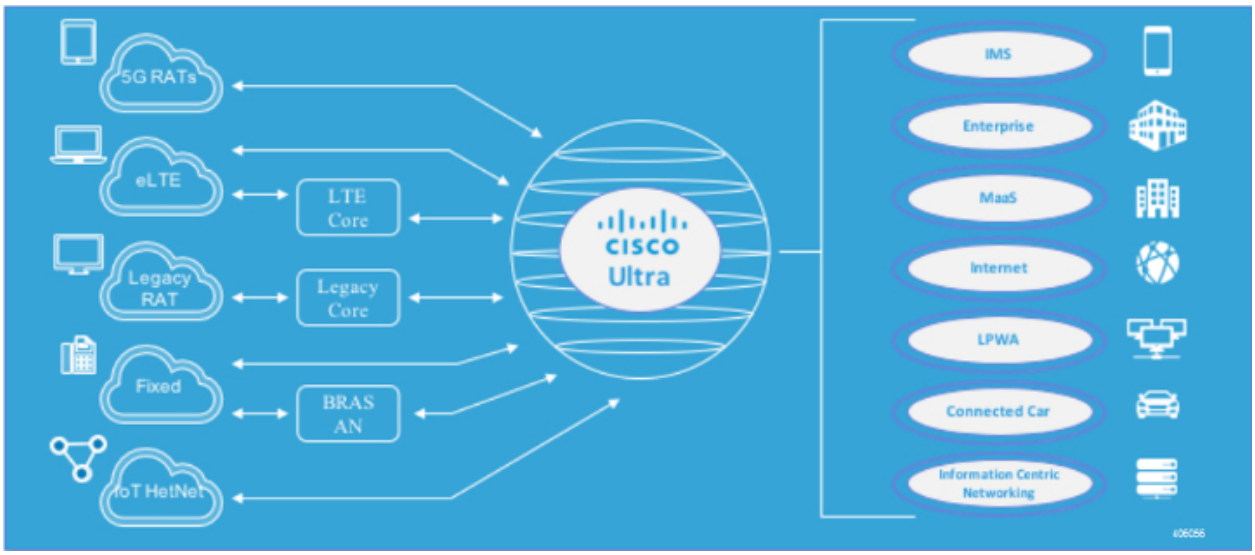
USP Introduction

The Ultra Services Platform (USP) is a 5G-ready virtual mobility network platform that provides a robust and highly scalable architecture that can quickly deploy mobility services across a distributed network in a virtualized environment. 5G will support countless emerging use cases with a variety of applications that drive significant variability in their performance attributes. From delay-sensitive mobile video applications to infrequent connectivity for simple devices, the diversity of use cases will demand substantially increased throughput, lower latency, ultra-high reliability with substantially higher connection densities.

The USP is a complex Virtual Network Function (VNF) conforming to the European Telecommunications Standards Institute (ETSI) Network Function Virtualization (NFV) and NFV Management and Orchestration (MANO) specifications. Unlike simple VNFs constrained to a single Virtual Machine (VM), the USP is a complex VNF comprised of multiple VNF Components (VNFCs) with a variable number of VMs depending on feature optioning and desired performance specifications.

Leveraging these virtualization, automation and orchestration technologies, the USP enables a NFV architecture that allows VNFs to be “sliced” into smaller, customizable end-to-end instances capable of seamless scaling regardless of the use case. The flexibility brings network providers to true Mobility-as-a-Service (MaaS) offering.

Figure 1: USP Network Slicing



USP Architecture

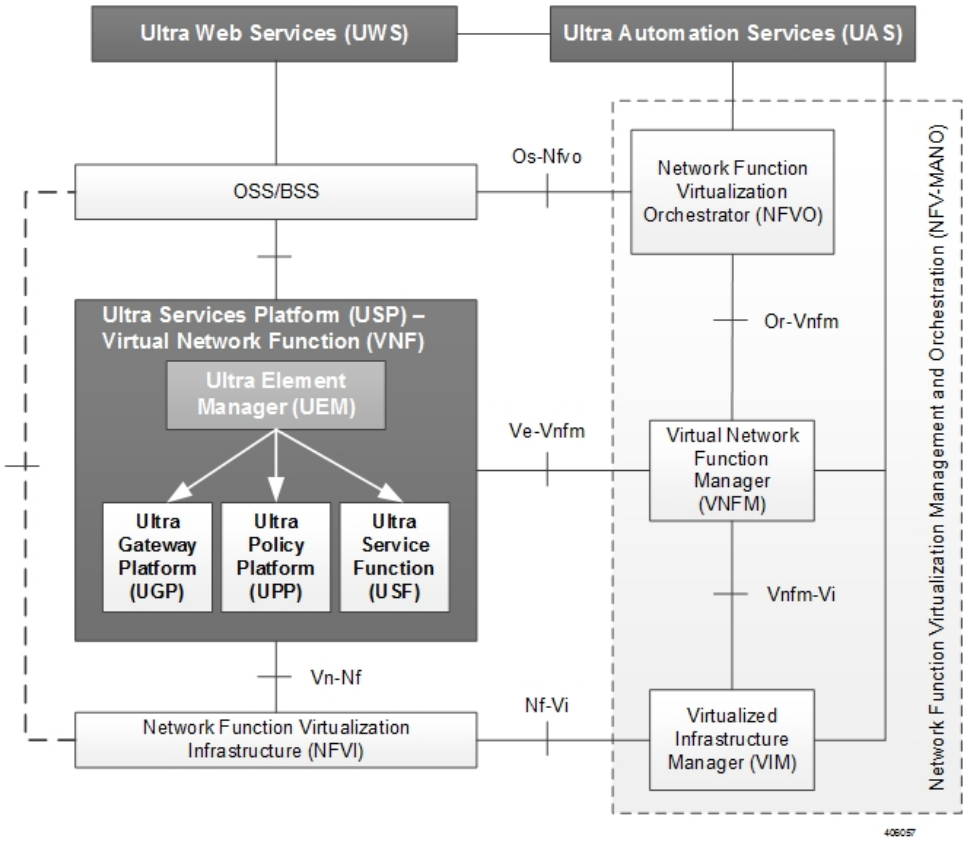
The USP solution comprises the following components:

- **Ultra Service Platform VNF:** The USP couples a Virtual Network Function Element Manager (VNF-EM) and multiple VNF components (VNFCs) into a single complex VNF. This coupling conforms to the European Telecommunications Standards Institute (ETSI) NFV Management and Orchestration (NFV MANO) standard and greatly simplifies MANO operations. A separate web-based customer portal called the Ultra Web Service (UWS) is supported in conjunction with the USP VNF and other network elements to simplify the deployment and management of the VNF.
- **Ultra Web Services (UWS):** The UWS provides an environment to graphically construct the USP VNF by allowing a user to select which VNF components are present and enter the necessary deployment parameters needed to instantiate the solution. Once this composition process is complete, the UWS passes the configuration to Ultra Automation Services which generates an ETSI NFV-compliant VNF Descriptor (VNFD). The VNFD is then on-boarded into an NFV Orchestrator (NFVO).

Important UWS is not supported in 6.x releases.

- **Ultra Automation Services (UAS):** UAS provides a suite of automation tools that simplify the on-boarding process of the USP VNF into any Cisco or third-party NFV infrastructure (NFVI).

Figure 2: USP Solution Components in the ETSI MANO Network



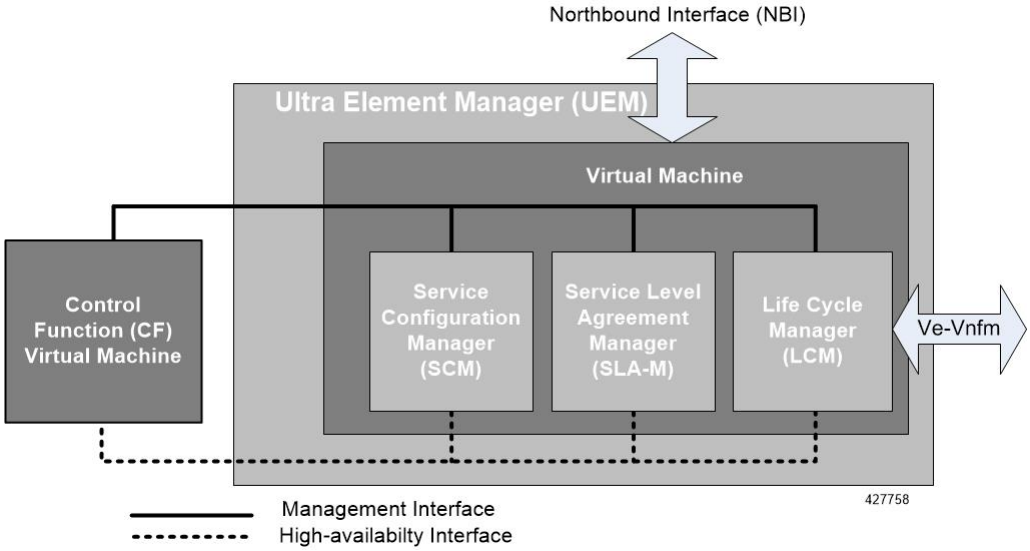
USP VNF Architecture

This section provides information on the VNF components (VNFCs) that comprise the USP architecture.

Ultra Element Manager (UEM)

The UEM manages all the major components of the USP architecture. Conforming to ETSI MANO, the UEM is modeled as the element management system (EMS) for the USP which is a complex VNF comprised of multiple VNFCs. The UEM and the complex VNF are represented to the Management and Orchestration (MANO) infrastructure through their own VNF descriptors (VNFDs).

Figure 3: Ultra Element Manager Composition



Although comprised of multiple modules, the UEM provides a single northbound interface (NBI) to external elements such as the OSS/BSS and Ultra Web Service (UWS).

The UEM provides the following network management functions:

- Configuration
- Fault management
- Usage accounting
- Performance measurement
- Security management
- Operational state of VNF

The northbound interface exposes all the information collected, aggregated and exposed through an API interface.

- All the interactions with entities northbound of the UEM happen via a single programmable API interface (e.g. REST, NETCONF, SNMP, etc.) for the purpose of collecting:
  - Configuration data for platform services and for Day-N configuration of its various components
  - Operational data pertaining to the system such as topology (VDU creation and organization) and different levels of VDU and service liveliness and KPIs based on the topology
  - Event streams (NETCONF notifications) that are used by the UEM to asynchronously notify northbound entities
  - Remote Procedure Calls (RPCs) used to expose some of the functionalities offered by the platform or its components such as packet tracing or mirroring
  - Asynchronous notifications: When an event that is relevant to northbound, is received from southbound, the SCM relays the event via a Netconf notification

These functions are provided via several different modules that comprise the UEM:

- **Lifecycle Manager (LCM):** The LCM exposes a single and common interface to the VNFM (Ve-Vnfm) that is used for performing life-cycle management procedures on a VNF. As a component within the UEM, it supports the various middleware application programming interfaces (APIs) required to interact with VNF and its components. Refer to Life Cycle Manager, on page 6 for more information.
- **Service Configuration Manager (SCM):** Leverages a YANG-based information model for configuration to provide configuration information to the VNFC Control Function (CF) VMs and other third-party components. It performs this functionality via NETCONF interfaces using pre-configured templates/network element drivers (NEDs). Configuration information is stored in the configuration database (CDB) and passed to the CF VM over the configuration interface via ConfD. Refer to Service Configuration Manager, on page 7 for more information.
- **Service Level Agreement Manager (SLA-M):** Provides timely access to information such as key performance indicators (KPIs), serviceability events, and diagnostic and troubleshooting information pertaining to components within the USP VNF instance such as:
  - The Lifecycle Manager
  - The Control Function (CF)
  - VMs that are part of the VNFCs
  - Any 3rd party applications related to USF service chains (depending on the VNFC)

  The SLA-M passes the information it collects over the northbound interface of the UEM. Refer to Service Level Agreement Manager , on page 7 for more information.

  Based on the StarOS, the CF is a central sub-system of the VNF that interacts with other sub-systems like service functions (SFs), network functions (NFs), and Application Functions (AFs) using field-tested software tasks that provide robust operation, scalability, and availability. It is equipped with a corresponding CDB for storing configuration information provided by the SCM via ConfD and/or CLI over the management interface. Refer to Control Function, on page 10 for more information.

Important In 6.3 and later releases, the UEM can be deployed as a separate VNF. In such a deployment, each UEM can deploy and communicate with multiple CFs.

High-availability (HA) is ensured across all of these components by the UEM-HA framework via a light-weight protocol that monitors the CF and SLA-M over the High-availability interface. All components are deployed redundantly. In the event of an issue, functions will be switched-over to the standby host. The SLA-M also uses the NETCONF interface to pull KPIs and event/log information from the CF.

Table 1: TCP/UDP Services and Open Ports for UEM, on page 5 lists the TCP/UDP services and the corresponding open ports for UEM.

Table 1: TCP/UDP Services and Open Ports for UEM

Port	Service
67	DHCP
68	DHCP

Port	Service
22	SSH
162	SNMP
830	NETCONF
2022	NETCONF
2024	NETCONF
2888	ZOOKEEPER
2889	ZOOKEEPER
2890	ZOOKEEPER
3888	ZOOKEEPER
3889	ZOOKEEPER
3890	ZOOKEEPER
4570	NETCONF
8888	NETCONF
2181	ZOOKEEPER

Life Cycle Manager

The Life Cycle Manager (LCM) is the UEM component that adapts an USP VNF to an external VNFM. The UEM provides a generic API to manage software, compute, and networking resources. When a VNFM brings up a new USP VNF, the VNFM starts redundant UEM VDUs. The VNFM also provides an initial set of VDUs as specified in the catalog for other USP virtual resources (for example, USP CF or USP SF). As the system initializes, the VNF components can bring VDUs online or offline using the UEM as a proxy to the external VNFM. The UEM provides a generic API to the other USP components, and a set of UEM adapters that attune the UEM to variety of external VNFMs.

Important The Cisco Elastic Services Controller (ESC) is the only supported VNFM in this USP release.

The LCM performs life-cycle management procedures on a VNF through a single and common interface to the VNFM. It can communicate with any off-the-shelf VNFM for resource allocation, configuration, monitoring, and lifecycle event updates. The LCM provides a common API to handle all VNFM instantiation flow requests for USP VNFs. It also communicates with a StarOS agent to provide all service and application level monitoring and lifecycle management.

The LCM provides the following functions:

- VNF registration through the onboarding of a virtualized network function descriptor (VNFD) by the VNFM

- Day-0 VNF configuration
- Handling key performance indicator (KPI) data in real-time
- Handling life-cycle events from VNFCs
- VNF termination

Communication between the Life Cycle Manager (LCM) and the VNFM is made possible through the integration of adapters that support VNFM products from multiple vendors. As an UEM component, the LCM includes middleware APIs that support the interface with SLA-M. The APIs are used to monitor KPIs pertaining to VNFC health and VM resource usage (for example, CPU, memory, etc.). APIs that support VNFC configuration establish interfaces to the CF via both the Management and High-availability buses to:

- Provision VMs based on information contained in virtualization descriptor units (VDUs) within the VNFD and associate the VMs to the internal network
- Add and initialize VMs as needed
- Request VNF infrastructure characteristics (for example, topology, deployment policies, etc.)
- Request VNF termination, migration, or destruction
- Request Day-N configuration for a specific VNFC
- Create and associate network ports to VDUs
- Provision networking configurations
- Provide life-cycle event notifications such as service status, configuration status, and HA events
- Provide an interface for determining NFVI information associated with the VDUs

Service Configuration Manager

The Service Configuration Manager (SCM) provides configuration information to the VNFC Control Function (CF) VMs and other third-party components. It performs this functionality via NETCONF interfaces using pre-configured templates/network element drivers (NEDs). Configuration information is stored in the configuration database (CDB) and passed to the CF VM over the management bus via ConfD data models.

During the initial VNF instantiation process, the SCM component will perform the initial detailed configuration of each VNF Component (gateway, in-line service function, etc.). This process is known as a Day-1 configuration. Additionally, when a change to any of the detailed configuration parameters of any of the VNF components after the VNF has already been deployed, the SCM will modify the specific parts of a detailed service configuration for any of the VNF Components. This is known as a Day-N configuration.

Service Level Agreement Manager

The Service Level Agreement Manager (SLA-M) provides timely access to information such as key performance indicators (KPIs), serviceability events, and diagnostic and troubleshooting information pertaining to components within the USP VNF instance including:

- The Life Cycle Manager (LCM)
- The Control Function (CF)
- VMs that are part of the VNFCs

- Any 3rd party applications related to USF service chains (depending on the VNFC)

This component is responsible for translating the requests from the Northbound layer into requests to the Southbound layer as well as for receiving and processing events and information from the Southbound layer to offer into aggregated form to the Northbound layer. It also populates a data store to maintain and expose historical data.

This component implements the following functionalities according to the way data are exposed northbound:

- **Immediate Access:** Requests coming from northbound (for example, access to the operational state of a particular VDU) are translated into a southbound request (for example, accessing the VDU operational state in a data source).
- **Historical Access:** The history of data or events in a store are maintained for later retrieval. SLA-M uses NCS's CDB for this purpose. The MAAPI session is initiated with NCS and the SLA-M proactively fills the operational data corresponding to historical data whenever it is collected (via periodic polling or notifications). In this scenario, access from northbound takes place by retrieving data directly from CDB instead of invoking a callback registered previously since no callback would have been registered for such data.
- **Aggregated Access:** In this case SLA-M retrieves the "non-aggregated" data from the data sources and then applies aggregation logic using the topology information exposed in the northbound model. When the callback corresponding to the aggregated access is invoked, the SLA-M accesses the northbound operational data describing the topology via MAAPI, and performs the needed aggregation of the retrieved data.

KPIs

Each unit of the system is monitored through a set of KPIs. KPIs are quantities that evolve over time. The SLA-M provides northbound entities with mechanism for accessing a current snapshot of such quantities (instantaneous KPIs) in aggregated or non-aggregated form. In addition, it keeps a history of a user-set number of the most recent KPI samples.

Refer to USP KPI Descriptions for a listing and description of KPIs supported in this release.

Two kinds of KPIs are collected:

- Basic (non-aggregated) KPIs
- Aggregated KPIs

Basic (non-aggregated) KPIs:

These are performance indicators at the VDU level which are provided to the SLA-M by either the CF or the VFNM Proxy Function.

The LCM provides all basic KPIs coming from the NFVI/VIM (for example, host/guest CPU load, memory, etc.), while the CF provides all other basic KPIs such as application specific metrics and process level information.

The following non-aggregate KPIs are provided by the CF to the SLA-M:

- Performance KPIs for each constituent VDR (*/vnfrs/vnfr/deployment-flavor-record/element-group-records /element-group-record/constituent-vdrs/constituent-vdr/performance-stats*).
- The contribution of the Performance KPIs for each constituent VDR to a specific Network Path (*/vnfrs/vnfr/deployment-flavor-record/element-group-records/element-group-record/service-function-chain*

-
records/service-function-chain-record/network-fwd-path-records/network-fwd-path-record/vdr-stats/vdr-stat).

- Flow Cache KPIs for each constituent VDR (*/vnfrs/vnfr/deployment-flavor-record/element-group-records/element- group-record/constituent-vdrs/constituent-vdr/flow-cache-stats*).

The following non-aggregate KPIs are provided by the VNFM-proxy to the SLA-M:

- NFVI KPIs for each constituent VDR (*/vnfrs/vnfr/deployment-flavor-record/element-group-records/element-group-record/constituent-vdrs/constituent-vdr/nfvi-stats*). These are exposed by the LCM to the UEM and the UEM mirrors them northbound.

Aggregated KPIs:

These are indicators derived by SLA-M from the basic KPIs and that reflect the performance of a group of VDUs.

The SLA-M builds aggregated KPIs at different levels of the grouping hierarchy by leveraging topology information. A typical example is building network throughput at the service chain level or slice level or system level. Note that while the SLA-M has the responsibility to build the aggregated KPI, it relies on other components to get the topology that drive such aggregation.

Starting from the non-aggregate KPIs described above, the SLA-M builds the following aggregated KPIs:

- Performance KPIs aggregated at:
- Network Path (*/vnfrs/vnfr/deployment-flavor-record/element-group-records/element-group-record /service-function-chain-records/service-function-chain-record/network-fwd-path-records/network-fwd-path -record/performance-stats*)
- Service Function Chain (*/vnfrs/vnfr/deployment-flavor-record/element-group-records/element-group -record/service-function-chain-records/service-function-chain-record/performance-stats*)
- Element Group (*/vnfrs/vnfr/deployment-flavor-record/element-group-records/element-group-record/performance-stats*)
- Vnf (*/vnfrs/vnfr/performance-stats*)
- Vnf for specific Service Function Chain (i.e. Performance-stats for a given service-function-chain across all the element-groups) (*/vnfrs/vnfr/service-function-chain-records/service-function-chain-record/performance-stats*)
- Flow Cache KPIs aggregated at:
    - VNF (*/vnfrs/vnfr/flow-cache-stats*)
- NFVI KPIs aggregated at:
    - Element group (*/vnfrs/vnfr/deployment-flavor-record/element-group-records/element-group-record/nfvi-stats*)
    - VNF (*/vnfrs/vnfr/nfvi-stats*)

Control Function

The Control Function (CF) is a StarOS based central sub-system of the VNF. It interacts with other sub-systems such as service functions (SFs), network functions (NFs), and Application Functions (AFs), and uses field-tested software tasks that provide robust operation, scalability, and availability. The VNFD and VNFR are equipped with a corresponding configuration database (CDB) for storing configuration information provided by the SCM via ConfD and/or CLI NEDs over the management interface.

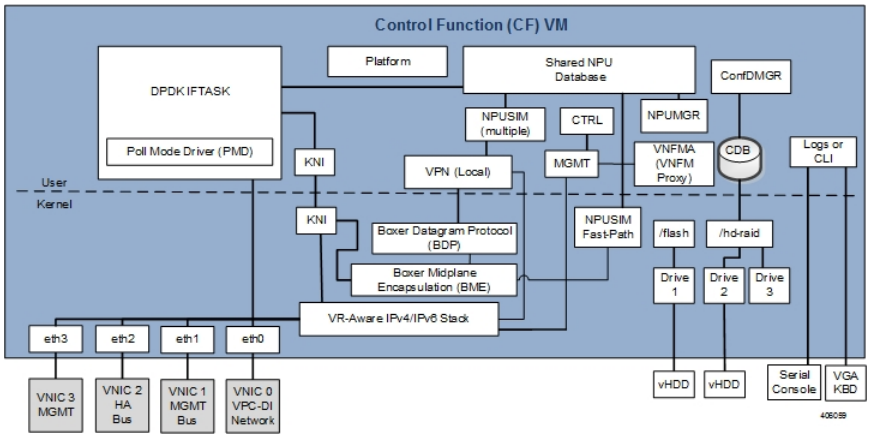
The CF also communicates over the High-availability (HA) interface for communicating with the LCM and to provide KPIs and event logs to the SLA-M.

Two CF VMs act as an active:standby (1:1) redundant pair. Within the StarOS, each CF VM is viewed as a virtual card and is responsible for the following functions:

- Hosting Controller tasks
- Hosting the Local context VPNMGR
- Hosting Local context (MGMT) and DI-Network vNICs
- Managing System boot image and configuration storage on vHDD
- Facilitating record storage on vHDD
- Providing Out-of-Band (OOB) management (vSerial and vKVM) for CLI and logging
- Working with the LCM to:
  - Bring VDUs online or offline during system initialization, request more VDUs for scale-out, return VDUs for scale-in lifecycle operations using VPD
  - Facilitate VDU internal management and configuration using predefined artifacts
- Providing KPI, event, and log information to the SLA-M as requested/needed

Note Refer to the Life Cycle Manager, on page 6 section for more information.

Figure 4: CF VM



Important The Intel Data Plane Development Kit (DPDK) Internal Forwarder task (IFTASK) is used to enhance USP system performance. It is required for system operation. Upon CF instantiation, DPDK allocates a certain proportion of the CPU cores to IFTASK depending on the total number of CPU cores.

Service Function

Service Function (SF) VMs provide service context (user I/O ports) and handle protocol signaling and session processing tasks. A UGP instance can have a maximum of 14 SF VMs, of which a maximum of 12 SF VMs can be active. See the *Cisco UGP System Administration Guide*.

Each SF VM dynamically takes on one of three roles as directed by the CF:

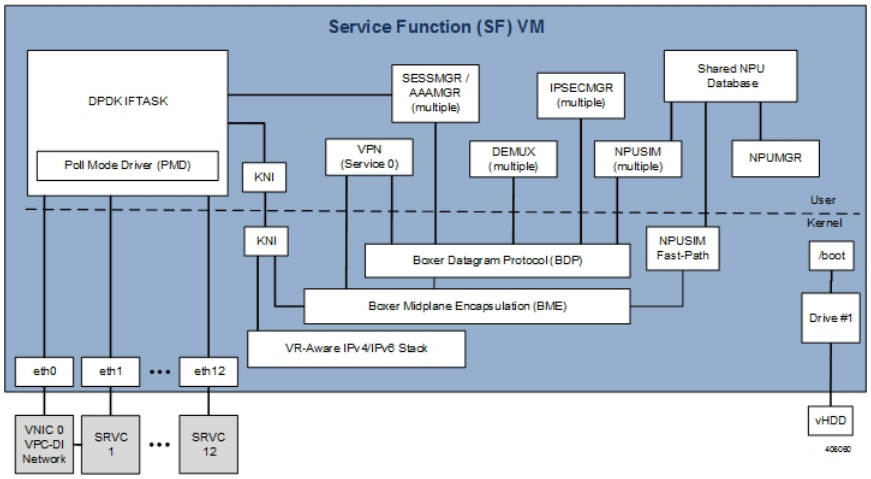
- Demux VM (flow assignments)
- Session VM (traffic handling)
- Standby VM (n+1 redundancy)

An SF provides the following functions:

Function Location	Runs on
NPUSIM fastpath/slow path (NPU emulation and routing to CPU)	Demux VM, Session VM, Standby VM
IFTASK based on the Intel® Data Plane Development Kit (DPDK)	Demux VM, Session VM, Standby VM
Non-local context (SRVC) vNIC ports	Demux VM, Session VM, Standby VM
VPNMGR and Demux for service contexts (first VM)	Demux VM
SESSMGR and AAAMGR for session processing (additional VMs)	Session VM
Egress forwarding decisions	
Crypto processing	

The minimum configuration for an Ultra Gateway Platform instance requires four SFs: two active, one demux, and one standby.

Figure 5: SF VM



Note The Intel Data Plane Development Kit (DPDK) Internal Forwarder task (IFTASK) is used to enhance USP system performance. It is required for system operation. Upon CF instantiation, DPDK allocates a certain proportion of the CPU cores to IFTASK depending on the total number of CPU cores.

When deployed in support of the Ultra Services Framework (USF), the SF facilitates the StarOS software tasks pertaining to the IP Services Gateway (IPSG) traffic detection function (TDF). The IPSG receives subscriber policy information from the Policy and Charging Rules Function (PCRF) over the Gx/Gx+ interface. It uses this policy information to steer subscriber session traffic received over the Gi/SGi interface through the SFC as required.

Network Function

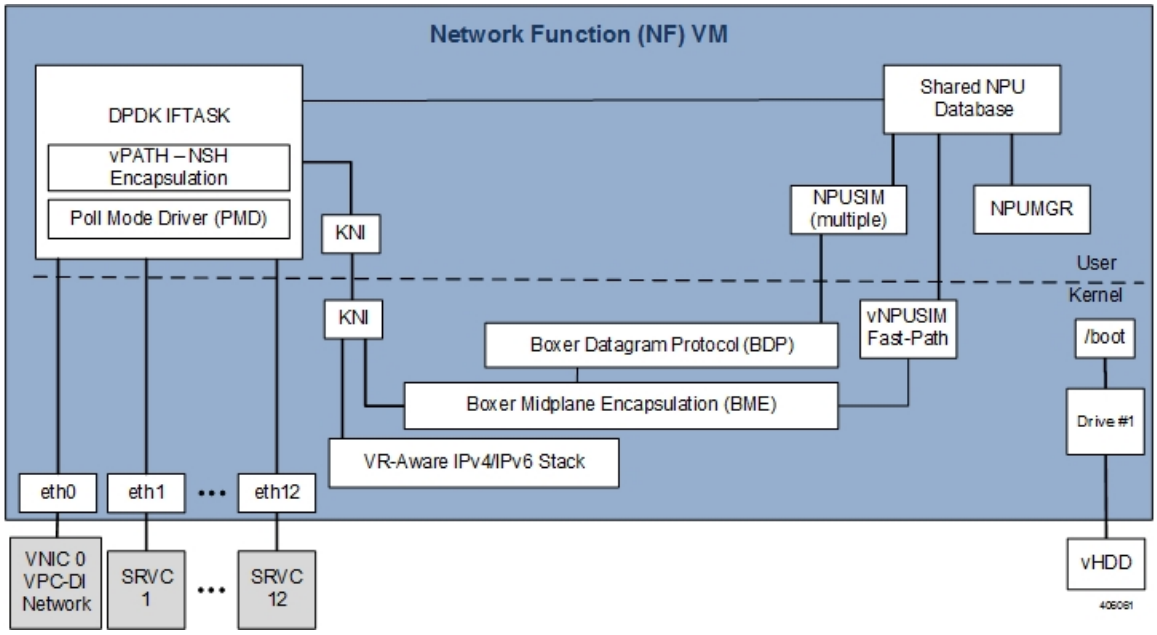
The Network Function (NF) is a virtual machine that is dedicated as a networking adapter between a DI system and external routers. The NF can be used to aggregate the VNF external connection points to a consolidated set of external interfaces. NF virtual machines are typically used for larger DI systems to limit the number of external interfaces to those present on a smaller set of virtual machines. The NF facilitates the building of large scale, high performance systems by providing the virtual equivalent of specialized Network Processing Unit (NPU) hardware.

The NF provides the following functions:

- Serves as a dedicated system for performing high speed traffic classification and flow/counter aggregation based on policies (n-tuple; each NF has access to complete set of policies)
- Limits the number of external interfaces required by aggregating external connection points to a consolidated set of high speed interfaces
- Operates as networking adapter between USP VNFs and external routers
- Subscriber awareness and stickiness as part of flow classification.
- Traffic classification and load balancing

The NF deploys a FAST-PATH architecture leveraging the NPU Manager and NPU SIM software tasks to ensure performance and scalability.

Figure 6: NF VM



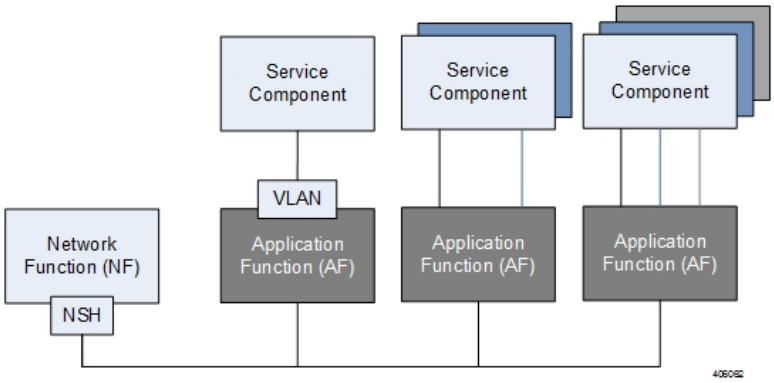
The mobility/DPDK internal forwarder (IF) is the core functional block for the USP architecture. It runs NPUSIM with DPDK into NF. The main functions of the mobility forwarder are:

- Performing the flow classification for each incoming packet, based on pre-configured rules.
- Deriving the service chain that needs to be associated with a flow
- Maintaining the subscriber stickiness - Meaning all the flows of a subscriber should land on the same service path (service path maps to AF).
- Performing the NSH encapsulation/ decapsulation. It uses NSH for communicating the service chain information across the nodes.

Application Function

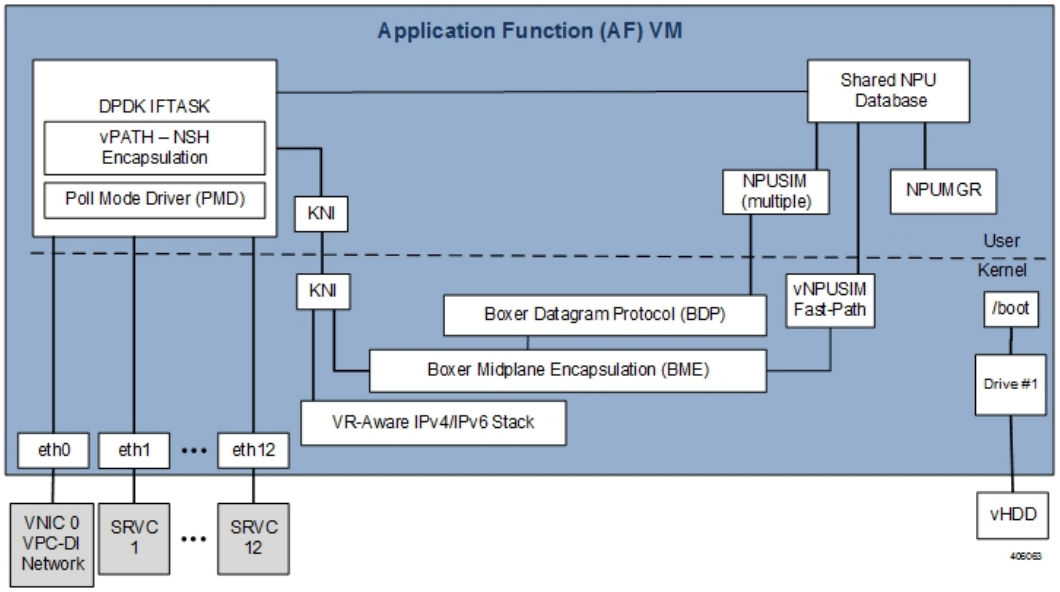
The Application Function (AF) is a virtual machine that is dedicated for Ultra Service Framework within a Gi-LAN Service Function Chain. The CF manages the system initialization, resource management, and high availability of the AF virtual machines. Packets that will be routed through a service function are encapsulated by the NF using NSH chain and routed to the AF. The AF learns of the specific service chain from the NSH header and routes the un-encapsulated packets through the Ultra Service Components (USCs) that comprise the chain. Once the packets are serviced, they are re-encapsulated and routed back to the NF.

Figure 7: AF Network



The AF VM maps the service chain identifier to a local tag representing the link/path between the NF and service component. The service path consists of a single service function, chain of different service functions, or service path spawned over multiple hosts. Like the NF, the AF deploys a FAST-PATH architecture leveraging the network processing unit (NPU) Manager and NPU SIM software tasks to ensure performance and scalability.

Figure 8: AF VM



USP VNF Types

The USP supports different types of VNFs that provide a variety of mobility services. Each VNF consists of components (VNFCs) which run on different virtual machines (VMs). The following VNF types are supported in this release:

- **Ultra Gateway Platform (UGP):** The UGP currently provides virtualized instances of the various 3G and 4G mobile packet core (MPC) gateways that enable mobile operators to offer enhanced mobile data services to their subscribers. The UGP addresses the scaling and redundancy limitations of VPC-SI (Single Instance) by extending the StarOS boundaries beyond a single VM. UGP allows multiple VMs

to act as a single StarOS instance with shared interfaces, shared service addresses, load balancing, redundancy, and a single point of management.

- **Ultra Policy Platform (UPP):** Delivers next generation policy and subscriber management functionality by leveraging the Cisco Policy Suite (CPS). CPS is carrier-grade policy, charging, and subscriber data management solution. It helps service providers rapidly create and bring services to market, deliver a positive user experience, and optimize network resources.

Note The UPP is not supported in this release.

- **Ultra Service Framework (USF):** The USF enables enhanced processing through traffic steering capabilities for subscriber inline services. USF Gi-LAN Service Function Chains (SFC) classify and steer traffic enabling mobile operators to quickly deploy new services and applications to their subscribers.

Ultra Gateway Platform

The UGP currently provides virtualized instances of the various 3G and 4G mobile packet core (MPC) gateways that enable mobile operators to offer enhanced mobile data services to their subscribers. The UGP addresses the scaling and redundancy limitations of VPC-SI (Single Instance) by extending the StarOS boundaries beyond a single VM. UGP allows multiple VMs to act as a single StarOS instance with shared interfaces, shared service addresses, load balancing, redundancy, and a single point of management.

The UGP includes the following features:

- Software defined, fully featured packet core functionality
- Multi-generational
- Separated management, control and user-planes
- Remotely deployable user plane for ultimate elasticity and scalability

Ultra Service Framework

The Ultra Service Framework (USF) is a Cisco 4G/5G pluggable framework that enables enhanced session processing through traffic steering capabilities for packets received over the Gi/SGi interface. It provides a pluggable framework for in-line, subscriber-aware, enhanced services.

It is integrated as separately upgradeable software packages. These applications are generically referred to enablers or services. However, in the context of USF, they are known as Ultra Service Components (USCs). Mobile operators not only deploy USCs to improve and add value to subscriber experience, but also to optimize and increase performance and efficiency within their network infrastructure.

The USF provides native life-cycle management and configuration automated by the converged platform framework. Leveraging 3GPP Flexible Mobile Service Steering (FMSS) and IETF(S) Gi-LAN Service Function Chaining (SFC) concepts, the USF classifies and steers session traffic (per-session or per-flow) to applications based on defined policies.

Ultra Automation Services

Ultra Automation Services (UAS) is an automation framework consisting of a set of software roles used to automate the VIM and USP-based VNF deployment as well as related components such as the VNFM. Beyond deployment automation, UAS manages software bundle components within an inventory manager. In addition, it can also be used to automate the deployment of third party components such as NFVI/VIM, test tools, and USFs that are not part of the distributed USP software bundle. The UAS consists of:

- AutoIT, on page 17
- AutoDeploy, on page 19
- AutoVNF, on page 21

Figure 9: VIM Installation Automation Workflow, on page 16 displays a high-level view of the VIM installation automation process workflow using UAS.

Figure 9: VIM Installation Automation Workflow

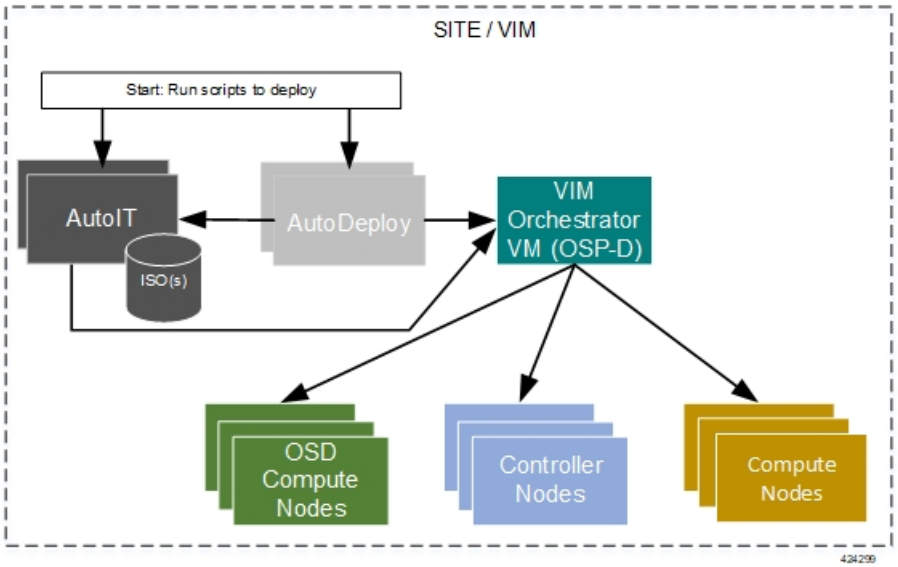
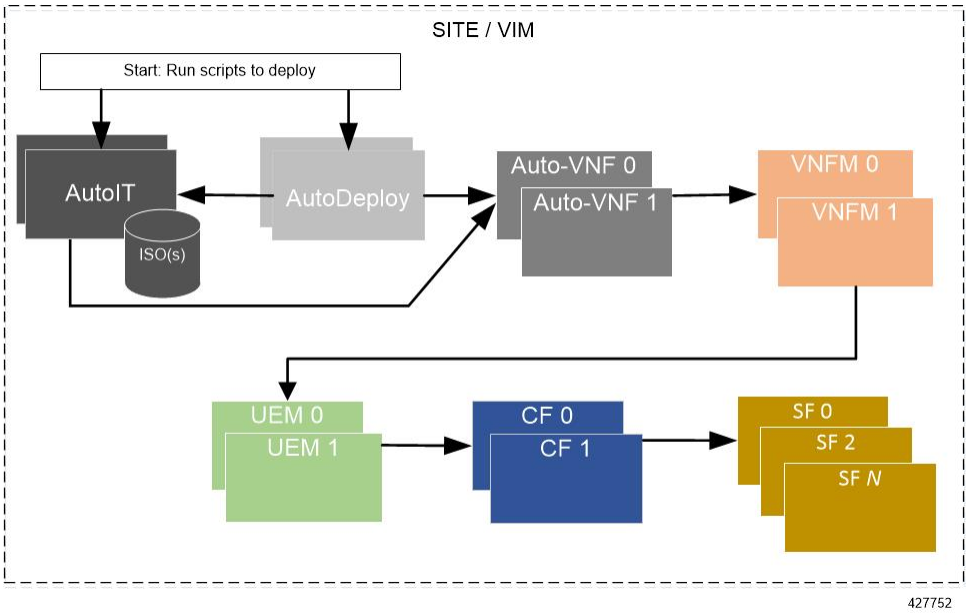


Figure 10: High-level Single VNF Deployment Automation Workflow, on page 17 displays a high-level view of the deployment automation workflow for a single VNF. In a multi-VNF environment, AutoDeploy can deploy up to four VNFs concurrently. Additional details pertaining to the deployment automation process are provided in the deployment automation documentation.

Important In this release, multi-VNF deployments are supported only in the context of the Ultra M solution. Refer to the *Ultra M Solutions Guide* for details.

Figure 10: High-level Single VNF Deployment Automation Workflow



AutoIT

AutoIT is the UAS software role used to automate the process of:

- Deploying the VIM Orchestrator (synonymous with the OpenStack Undercloud).
- Installing the virtual infrastructure manager (VIM, synonymous with the OpenStack Overcloud) which manages the network function virtualization infrastructure (NFVI).
- Onboarding/upgrading the USP ISO software package onto the Ultra M Manager Node.

AutoIT performs the deployments based on manifests it receives from AutoDeploy. Additionally, also hosts a webserver to facilitate VM deployment and delivery of software packages using REST and ConfD APIs for provisioning Overcloud nodes.

AutoIT can be deployed in the following scenarios:

- As a single VM on the Ultra M Manager Node (the same physical server as AutoDeploy and OSP-D VM) during a bare metal installation.
- In high-availability (HA) mode which provides 1:1 redundancy. When deployed in HA mode, two AutoIT VMs are deployed: one active, one standby.
- As a single VM within an existing OpenStack deployment.
- In HA mode within an existing OpenStack deployment.

When supporting VIM installation automation processes, AutoIT:

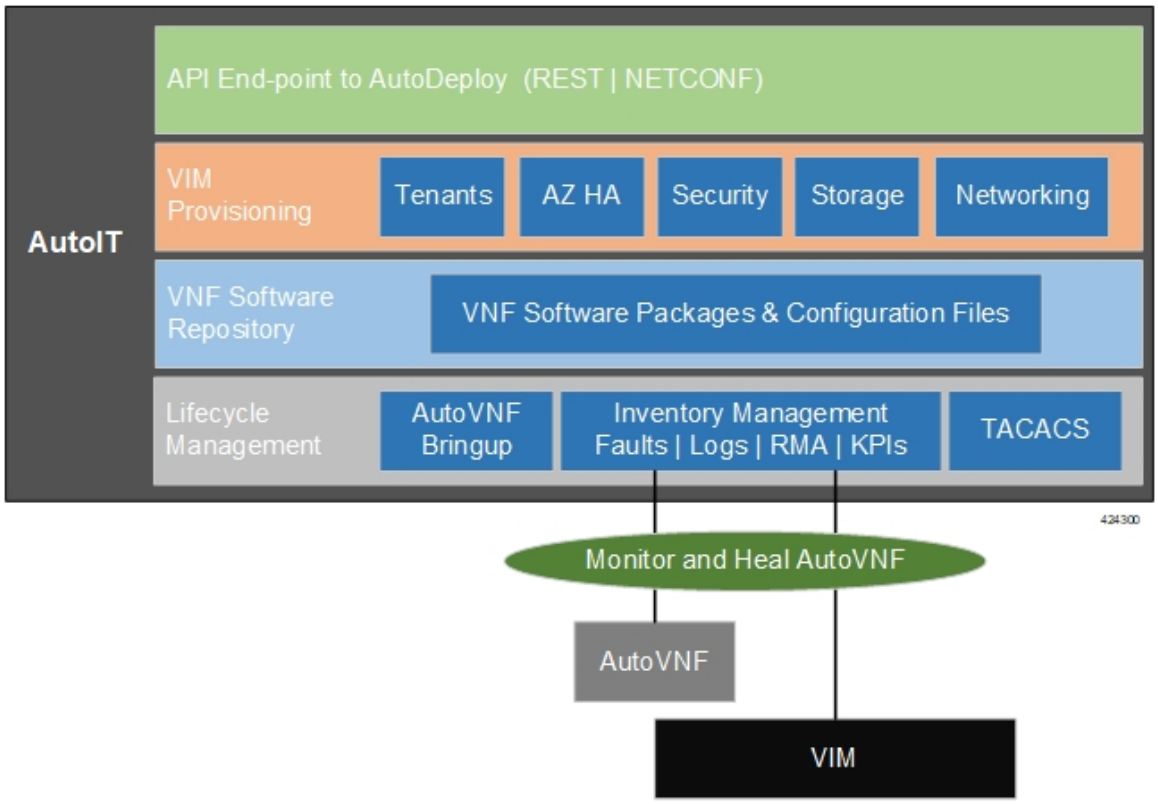
- Sets up AutoIT nodes
- API endpoint based on ConfD to Auto-Deploy and NSO

- Deploys the VIM Orchestrator
- Works through the VIM Orchestrator to deploy the VIM
- Brings up OSP-D as a VM

When supporting VNF deployment automation processes, AutoIT:

- Onboarding Ultra Automation Services (UAS) VMs.
- VIM provisioning to onboard VNFs.
- Manages different version of software packages by hosting into YUM repo.
- APIs to onboard VNF packages.
- Brings up AutoVNF VMs and monitors for failures.
- Stores release public key information in the ISO database for RPM signature verification by YUM through the installation process.

Figure 11: AutoIT Functions



Important In this release, AutoIT is only supported for use with Ultra M solutions based on the Hyper-Converged architecture.

In addition to supporting deployment workflows, AutoIT provides a centralized monitor and management function within the Ultra M solution. This function provides a central aggregation point for events (faults and alarms) and a proxy point for syslogs generated by the different components within the solution.

Table 2: TCP/UDP Services and Open Ports for AutoIT, on page 19 lists the TCP/UDP services and the corresponding open ports for AutoIT.

Table 2: TCP/UDP Services and Open Ports for AutoIT

Port	Service
22	SSH
8888	CONFD
4569	CONFD
514	SYSLOG
2022	CONFD
5000	HTTP
8008	CONFD
2024	CONFD
161	SNMP

AutoDeploy

AutoDeploy is the UAS software role that provides single- and multi-Site AutoVNF orchestration. In this context, a “Site” is a single VIM instance. As such, a single AutoDeploy instance is capable of deploying the AutoVNF UAS software roles within multiple deployment scenarios:

- Single VIM/Single VNF
- Single VIM/Multi-VNF

Important In this release, multi-VNF deployments are supported only in the context of the Ultra M solution. Refer to the *Ultra M Solutions Guide* for details.

In a multi-VNF environment, AutoDeploy can deploy up to four VNFs concurrently. Additional details pertaining to the deployment automation process are provided in the deployment automation documentation.

AutoDeploy can be deployed in the following scenarios:

- As part of VIM installation automation process:
  - On bare-metal with high availability (HA) support. HA support provides 1:1 VM redundancy. When deployed in HA mode, two AutoDeploy VMs are deployed on the same physical server: one active, one standby.

- On bare-metal without HA support. In this scenario, a single AutoDeploy VM is deployed.

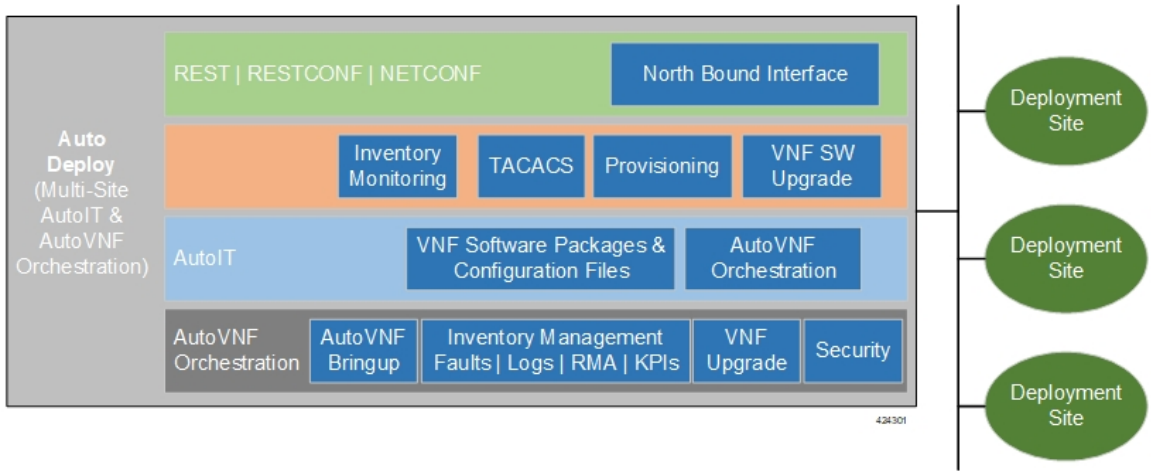
- As part of an existing deployment:
  - In HA mode within an existing OpenStack deployment. When deployed in HA mode, two AutoDeploy VMs are deployed on the same physical server: one active, one standby.
  - As a single VM within an existing OpenStack deployment.

In this release, one AutoDeploy VM is deployed per VIM. The AutoDeploy VM must have network access to the VIM in order to provide orchestration.

Once instantiated, AutoDeploy provides the following functionality:

- AutoVNFs bootstrapping and provisioning for deployments (Day-0/Day-1/Day-N).
- AutoVNF Deployments Life-Cycle including start, stop and Inventory management (consolidated).
- Performs release image signing validation by verifying the certificate and public key provided in the release ISO.

Figure 12: AutoDeploy Functions



AutoDeploy operations are performed using any of the following methods:

- ConfD CLI and API based transactions
- WebUI based transactions

Table 3: TCP/UDP Services and Open Ports for AutoDeploy, on page 20 lists the TCP/UDP services and the corresponding open ports for AutoDeploy.

Table 3: TCP/UDP Services and Open Ports for AutoDeploy

Port	Service
5000	HTTP
22	SSH

Port	Service
2022	CONFD
2024	CONFD
8443	HTTPS
8888	CONFD
514	SYSLOG

AutoVNF

AutoVNF is the software role within UAS that provides deployment orchestration for USP-based VNFs. It does this by emulating an NFVO and VNFM for deployments.

When used in Ultra M solutuon deployments, AutoVNF is instantiated by the AutoDeploy software role based on configuration data you provide. It is deployed with a 1:1 HA redundancy model. Processes across the VMs are monitored and restarted if necessary. ConfD synchronizes the CDB between the active and standby VMs. Each of the VMs are deployed on separate Compute nodes within your VIM.

For VNF deployments brought up using only AutoVNF (e.g. Stand-alone AutoVNF-based deployments), only a single VM is deployed.

Once operational, AutoVNF provides the following functionality:

- Deploys the Elastic Services Controller (ESC), which serves as the VNFM, per configurable YANG-based definitions.

  Note The Cisco Elastic Services Controller (ESC) is the only supported VNFM in this USP release.

- Onboards all required UEM VMs via the VNFM.
- Leverages configurable YANG-based definitions to generate the VNF descriptor (VNFD) required to onboard the VNF using UEM workflows.
- Determines all required resources for the VNF including images, flavors, networks, subnets and invokes NETCONF-based APIs to provision all of these resources into OpenStack through the VNFM.
- Ensures all references, network, images, and flavors exist on the VIM, if supplied.
- Monitors for NETCONF-based notifications, submits the transaction, and waits until the given transaction succeeds.
- Monitors inventory in terms of operational state and KPIs and auto-heals the VNFM and UEM.
- Orchestrates USP-based VNF upgrades regardless of whether or not Inter-Chassis Session Recovery (ICSR) is enabled on the VNF.
- Implements a ConfD-based architecture to provide life cycle management (LCM) through VNF-EM, VNFM, and VIM plugins as shown in Figure 14: AutoVNF ConfD-based Architecture for Deployment Automation, on page 23.

- Supports standard, ConfD-based REST/RESTCONF/NETCONF north-bound interfaces (NBIs).
- Provides VNF security, credentials, and SSH keys through the use of secure-tokens.
- Hosts an HTTP server to serve GET URLs supplied into the VNFD that include such things as configuration files, VDU images, etc.
- Supplies the VNFD to the UEM upon instantiation as Day-0 configuration using an appropriate VNFM-supported mechanism (e.g. in the case of ESC as the VNFM, the VNFD is passed as a Day-0 configuration using the ESC's deployment APIs).
- Onboards all Day-0 configuration files onto the UEM to be passed on to VDUs.
- Allocates the management IP for the CF and UEM VMs along with Virtual IP (VIP) addresses.

Figure 13: AutoVNF Functions

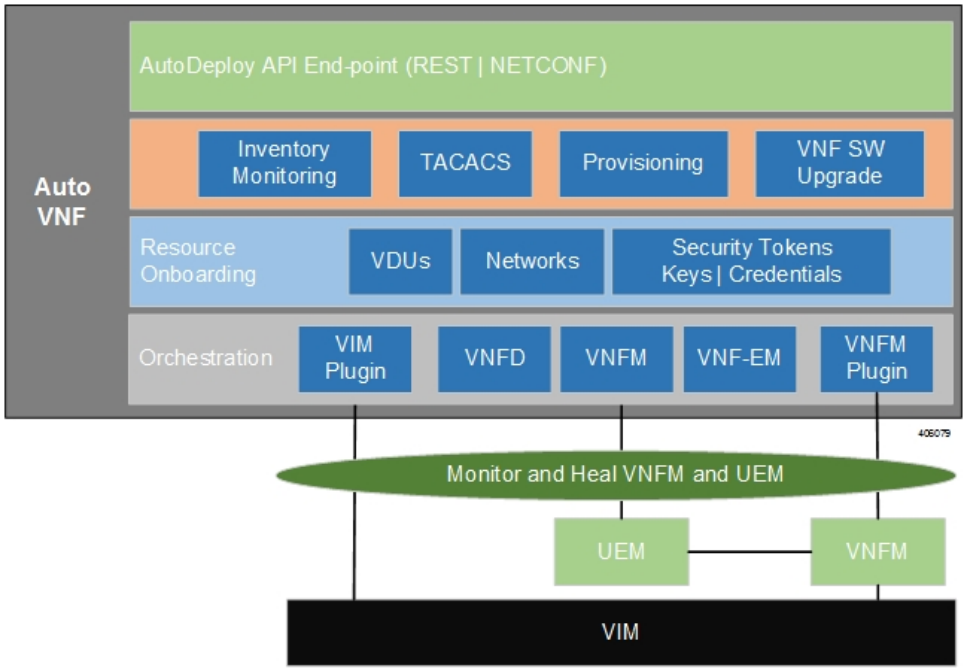
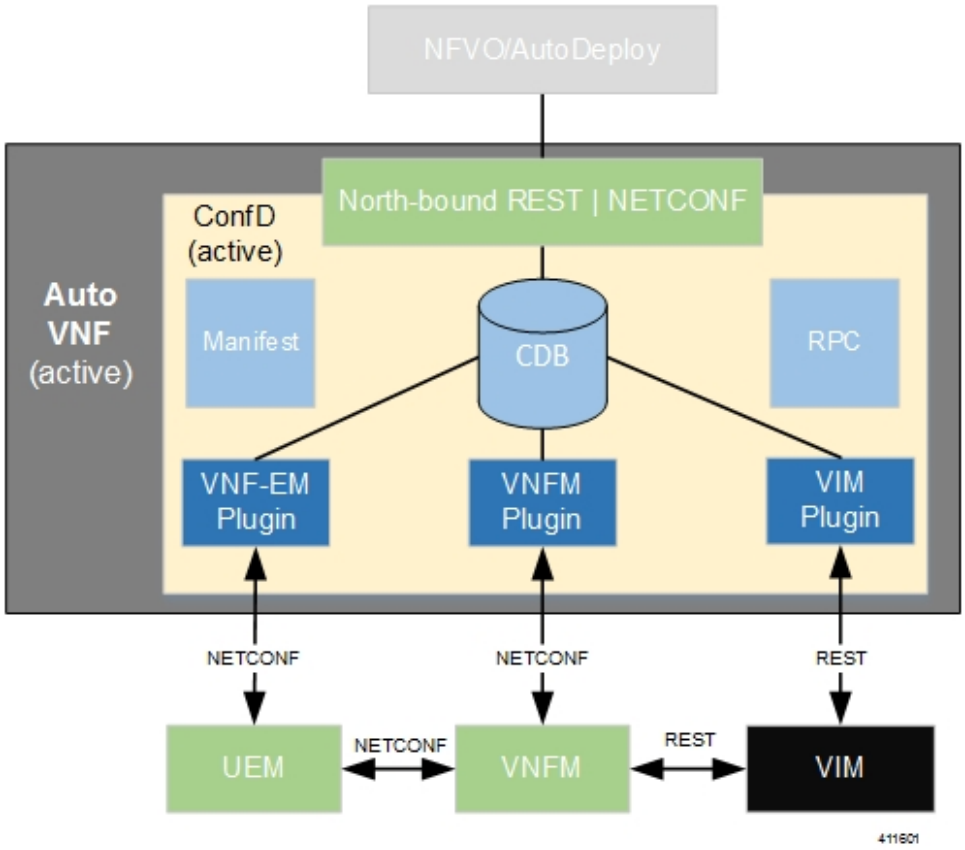


Figure 14: AutoVNF ConfD-based Architecture for Deployment Automation



AutoVNF operations can be performed using any of the following methods:

- ConfD CLI based transactions
- WebUI based transactions
- Netconf based transactions

Table 4: TCP/UDP Services and Open Ports for AutoVNF, on page 23 lists the TCP/UDP services and the corresponding open ports for AutoVNF.

Table 4: TCP/UDP Services and Open Ports for AutoVNF

Port	Service
5000	HTTP
22	SSH
2022	CONFD
2024	CONFD
4569	CONFD

Port	Service
8443	HTTPS
8888	CONFD
8008	CONFD
514	SYSLOG
2181	ZOOKEEPER. AutoVNF connects to Zookeeper running on UEM for status

Ultra Web Services

The Ultra Web Service (UWS) provides a web-based graphical user interface (GUI) and a set of functional modules that enable users to manage and interact with the USP VNF. It provides a single framework and a top-level dashboard for users to interact with the USP VNF. It includes the following features:

- Service Assurance
- Validation
- VNF-EM Virtualization
- VNF Components
- NFVI/VIM

Leveraging backend-APIs, the VNF visualization module of UWS is used to create, deploy and monitor a variety of USP VNFs based on specific use cases and applications. The VNFs can include definitions for the gateway type, policy options, service function chaining features, and more. After the VNFs are created, users can deploy each VNF to a target infrastructure choice. The USP tracks deploy operations. Users can display the tracked information on the dashboard, and can monitor the status of a selected deployment operation. The dashboard also displays aggregate KPIs from all deployed VNFs, allowing users to easily view and monitor aggregate metrics for a given environment.

UWS software is part of the UAS software package and is installed automatically with the AutoDeploy software role.

The following browser versions are supported for use with the UWS GUI:

- Firefox: 55.0.3 (64-bit)
- Safari: Version 10.1.1 (10603.2.5)
- Chrome: 58.0.3029.110 (64-bit)
- Edge: 38.14393.1066.0

Important UWS is not supported in 6.x releases.

USP VNF Component Redundancy and Availability

Platform Requirements

The USP VNF relies on the underlying hardware and hypervisor for overall system redundancy and availability.

The hardware and hypervisor should provide:

- Redundant hardware components where practical (such as power supplies and storage drives)
- Redundant network paths (dual fabric/NICs, with automatic failover)
- Redundant network uplinks (switches, routers, etc.)

High availability can be achieved only if the underlying infrastructure (hosts, hypervisor, and network) can provide availability and reliability that exceeds expected values. The USP VNF is only as reliable as the environment on which it runs.

Inter-Chassis Session Recovery (ICSR) is also recommended to improve availability and recovery time in the case of a non-redundant hardware failure (such as CPU, memory, motherboard, hypervisor software). ICSR provides redundancy at the session level for gateways only. See ICSR Support, on page 27 for more information.

UEM Redundancy

A minimum of three UEM VMs is required to support redundancy in USP releases prior to 6.3. In 6.3 and later releases, changes are made to the UEM redundancy model in order to optimize the VM requirements. In this release, a minimum of two UEM VMs is sufficient to support redundancy. The UEM supports active-standby 1:1 instances for redundancy reasons.

When three UEM VMs are used, they are deployed as part of an HA cluster which are 1:n redundant for overall management and inter-VNFM communications. The three VMs are deployed as follows: 1 leader or master (active), 1 follower or slave (standby), and 1 follower (standby).

When two VMs are used, the master UEM has two zookeeper instances running, both instances have their own IP, ID, PID, log/data directory and configuration files. The slave UEM also has one zookeeper running, thus meeting three zookeeper instance requirements.

The UEM services will no longer run on the slave UEM to simplify troubleshooting, maintenance, and synchronization related issues.

The number of instances for UEM can be defined as 2 or 3 through the VNFC configuration within NSD. You can configure the instances based on the resource availability and deployment requirements.

To configure the number of instances for UEM, use the following parameter for VNFC EM:

`number-of-instances` *`<instance_num>`*

Note that the **number-of-instances** parameter is mandatory. This parameter allows the user to configure either 2 or 3 UEM instances.

In releases prior to 6.3, the default value was 3 and this parameter was not user configurable. In release 6.3 and beyond, the default value is 2.

Example Configuration for AutoDeploy:

```
nsd nsd-autovnf
    vnfd vpc
      vnfc em
          number-of-instances 2
...
```

For more information, refer to the *Cisco Ultra Services Platform NETCONF API Guide*.

CF Redundancy

By default, the UEM deploys two CF VMs which are 1:1 redundant for control of the USP VNF and the local context/management port. This is the recommended configuration.

The management port vNIC on both CFs are 1:1 redundant for each other and must be placed in the same VLAN in the infrastructure. Only one management port is active at a time.

Note The two CF VMs must not run on the same physical host (server or blade) to achieve redundancy in case of the failure of the host or hypervisor.

SF Redundancy

SFs are deployed using 1:N redundancy. It is recommended that you have at least 2 active and 1 standby SF, however, the number of SF instances will change according to your deployment requirements.

Each SF VM provides network connectivity for service ports. Each SF provides one or more ports and associated interfaces, but the SFs do not provide 1:1 port redundancy as they are not paired together. Redundancy of SF ports should be achieved using ECMP or another supported L3 protocol.

The total throughput required of the USP VNF Instance should not exceed N-2 SFs with session recovery enabled so that any single SF can fail while the others take over its load. Use of loopback interfaces for service IP addresses is highly recommended.

Cisco recommends that you use Bidirectional Forwarding Detection (BFD) and Link Aggregation Group (LAG) for detection of path failures between an SF and the peer router so ECMP paths are excluded in the event of a failure.

1:1 session redundancy within a VNF and Inter-Chassis Session Recovery (ICSR) between VNFs is supported. Note that the session state is check-pointed at various call points within a call flow. Although session state is check-pointed in the UGP, the IP flow state and connection tracking tables are not mirrored. Therefore, any state associated with an IP flow will be lost.

When session recovery is enabled, one VM becomes the VPN/Demux and the remainder are session processing VMs. A standby SF can provide redundancy for any other SF.

Note Each SF VM must run on a different physical host to achieve redundancy in case of the failure of the host or hypervisor.

NF Redundancy

NFs are deployed using 1:N redundancy. You may adjust the number of NF instances according to your deployment requirements.

Note Each NF VM must run on a different physical host to achieve redundancy in case of the failure of the host or hypervisor.

AF Redundancy

AFs are deployed using 1:N redundancy. You may adjust the number of AF instances according to your deployment requirements.

Note Each AF VM must run on a different physical host to achieve redundancy in case of the failure of the host or hypervisor.

Ultra Service Component (USC) Redundancy

The Ultra Services Components (USCs) used in the USF are deployed along with the AF into a MANO construct called an Element Group (EG). An EG is set of VDUs arranged for a unit of redundancy. As such, redundancy is available at the EGs-level and not for the individual USCs. An N:1 redundancy model is supported for Element groups.

ICSR Support

USP VNFs support Inter-Chassis Session Recovery (ICSR) between two VNF instances for services that support Layer 3 ICSR in the StarOS software release. When more than one service type is in use, only those services that support ICSR will be able to use ICSR.

ICSR supports redundancy for Site/row/rack/host outages, and major software faults. To do so, the two USP VNF instances should be run on non-overlapping hosts and network interconnects. ICSR is supported only between like-configured UGP instances.

Note ICSR between an USP VNF instance and another type of platform (such as an ASR 5500) is not supported.

For additional information, refer to the *Inter-Chassis Session Recovery* chapter in the *System Administration Guide* for your platform.