Open Source MANO

An ETSI OSM Community White Paper

OSM RELEASE THREE

A TECHNICAL OVERVIEW

October 2017

Authors:
Adam Israel, OSM VNF Configuration and Abstraction Task Force Lead, Canonical
Adrian Hoban, OSM Technical Steering Committee (TSC) Chair, Intel
Alfonso Tierno Sepúlveda, OSM Resource Orchestration MDG Lead, Telefónica
Francisco Javier Ramon Salguero, OSM Chair, Telefónica
Gerardo García de Blas, OSM TSC, Telefónica
Kiran Kashalkar, OSM User Interface MDG Lead, RIFT.io
Marco Ceppi, OSM Network Service to VNF Communication MDG Lead, Canonical
Mark Shuttleworth, OSM TSC, Canonical
Matt Harper, OSM TSC, RIFT.io
Michael Marchetti, OSM DevOps MDG Lead, Sandvine
Rajesh Velandy, OSM NW Service Orchestration MDG Lead, RIFT.io
Silvia Almagia, OSM, ETSI Centre for Testing and Interoperability
Vanessa Little, OSM TSC, VMware

Editor:
Chris Buerger, OSM Marketing Task Force Chair, Intel

ETSI
06921 Sophia Antipolis CEDEX, France
Tel +33 4 92 94 42 00
info@etsi.org
www.etsi.org

Contents

1 Introduction & Scope

ETSI OSM is an operator-led ETSI community that is delivering a production-quality open source Management and Orchestration (MANO) stack aligned with ETSI NFV Information Models and that meets the requirements of production NFV networks.

The OSM community has set itself the goal of being a world-class production ready solution. OSM Release THREE represents another significant step along this path. It has been engineered, tested and documented to be functionally complete to support Operator RFx processes, and to be a key component for internal/lab and external/field trials as well as interoperability and scalability tests for virtual network functions and services. It allows for rapid installation in VNF vendor, system integrator and operator environments worldwide. OSM Release THREE substantially enhances interoperability with other components (VNFs, VIMs, SDN controllers, monitoring tools) and provides a plug-in framework to make platform maintenance and extensions significantly easier to provide and support.

Building on the capabilities developed for prior releases, Release THREE improves administrator and developer experience, both in terms of usability and installation procedure as well as the modelling of virtualized network functions (VNFs) and network services. In line with the goals of the Open Source MANO project, the output of this modelling work has been contributed to the ETSI NFV Industry Specification Group. Release THREE also provides extremely flexible VNF configuration and advanced networking management as well as improved security capabilities, with advanced access controls.

This White Paper outlines the main architecture of OSM, and the new capabilities developed and open-sourced as part of Release THREE. It also provides insight into a number of the development themes the OSM Technical Steering Committee (TSC) has been pursuing in close collaboration with the OSM End-User Advisory Group (EUAG) as well as the broader OSM community.

More information about ETSI OSM, its community and how to download OSM Release THREE can be found here: https://osm.etsi.org/.

The authors of this White Paper would like to extend a sincere ‘Thank You’ to the entire OSM community that contributed their passionate, collaborative and innovative work to make Release THREE possible.

2 OSM Scope

The OSM community has defined an expansive scope for the project covering both design-time and run-time aspects related to service delivery for telecommunications service provider environments. The express goal is that the OSM code base can be leveraged in these environments as-is in a Roll-Your-Own context, or in whole and/or part of a commercial product offering.

Figure 1 shows the approximate mapping of scope between the OSM components and the ETSI NFV MANO logical view (the background image was extracted from Figure 4 in the NFV Reference Architecture Framework, ETSI GS NFV 002 V1.2.1 (2014-12)).

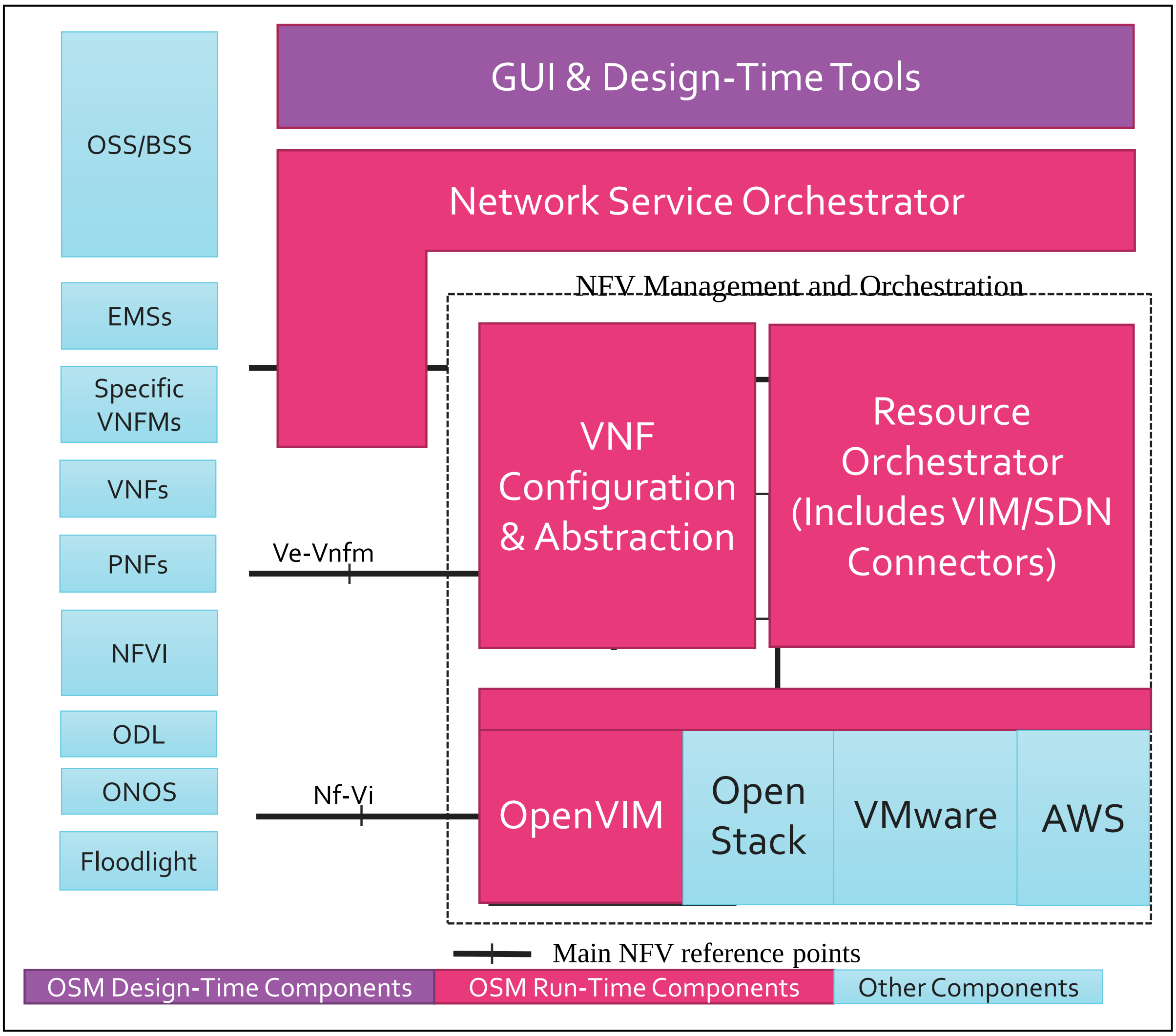
OSS/BSS
GUI & Design-Time Tools
Network Service Orchestrator
NFV Management and Orchestration
EMSs
Specific VNFMs
VNFs
PNFs
NFVI
ODL
ONOS
Floodlight
VNF Configuration & Abstraction
Resource Orchestrator (Includes VIM/SDN Connectors)
Ve-Vnfm
Nf-Vi
OpenVIM
Open Stack
VMware
AWS
Main NFV reference points
OSM Design-Time Components
OSM Run-Time Components
Other Components

Figure 1: OSM Mapping to ETSI NFV MANO

2.1 Run-Time Scope

The run-time scope of OSM includes:

- An automated Service Orchestration environment that enables and simplifies the operational considerations of the various lifecycle phases involved in running a complex service based on NFV.
- A superset of ETSI NFV MANO where the salient additional area of scope includes Service Orchestration but also explicitly includes provision for SDN control.
- Delivery of a plugin model for integrating multiple SDN controllers.
- Delivery of a plugin model for integrating multiple VIMs, including public cloud based VIMs.
- Delivery of a plugin model for integrating multiple monitoring tools into the environment.
- One reference VIM that has been optimized for Enhanced Platform Awareness (EPA) to enable high performance VNF deployments.
- An integrated “Generic” VNFM with support for integrating “Specific” VNFMs.
- Support to integrate Physical Network Functions into an automated Network Service deployment.
- Being suitable for both Greenfield and Brownfield deployment scenarios.
- GUI, CLI, Python based client library and REST interfaces to enable access to all features.

2.2 Design-Time Scope

The design-time scope of OSM includes:

- Support for a model-driven environment with Data Models aligned with ETSI NFV MANO.
- The capability for Create/Read/Update/Delete (CRUD) operations on the Network Service Definition.
- Simplifying VNF Package Generation.
- Supplying a Graphical User Interface (GUI) to accelerate the network service design time phase, VNF on-boarding and deployment.

3 OSM Release THREE Architecture

The OSM community has continued to make advancements in the OSM architecture. The logical blocks that represent the functionality delivered by OSM are shown in Figure 2. These are color coded by run-time and design time components.

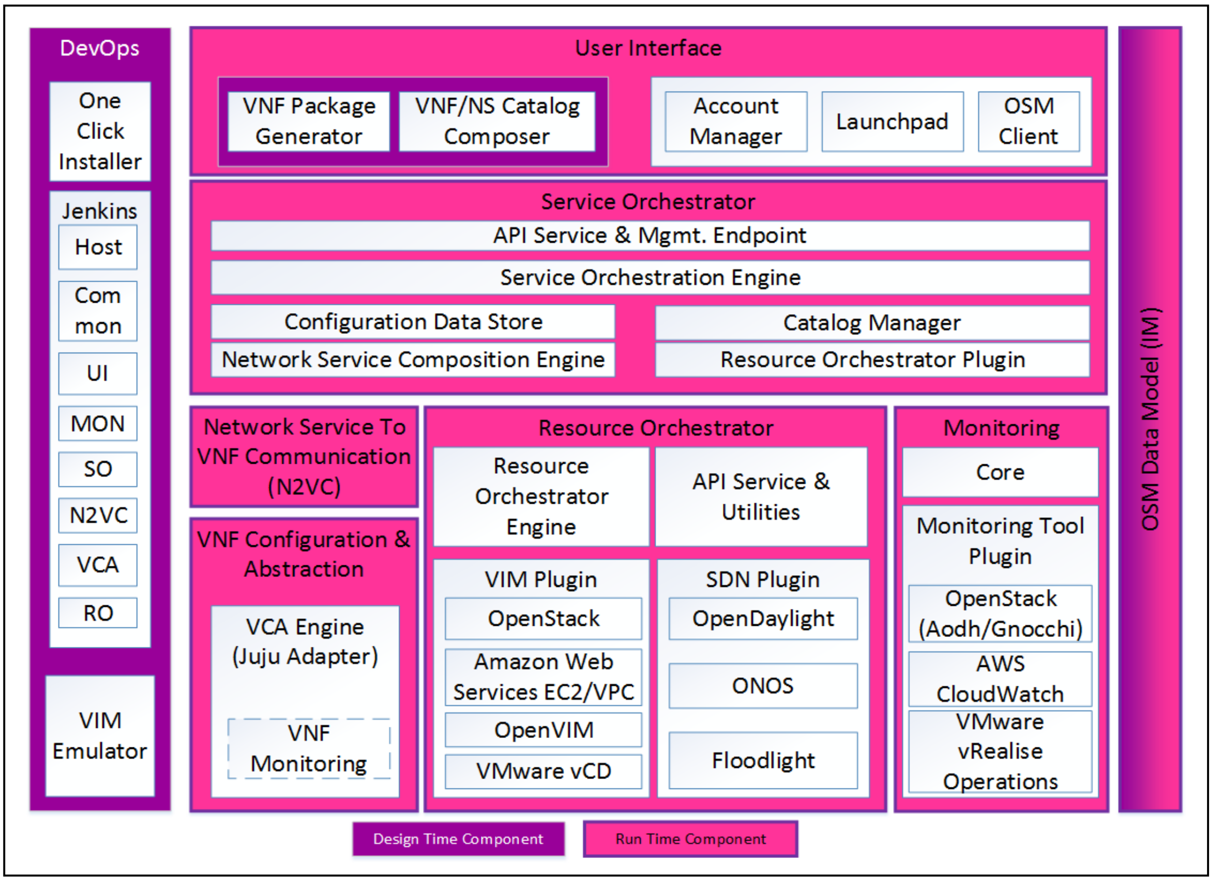

Figure 2: OSM Release THREE Architecture

3.1 OSM Release Design-Time Components

3.1.1 DevOps

DevOps has taken a significant step forward in OSM Release THREE with the creation of a new module responsible for the Continuous Integration (CI) and Continuous Development (CD) workflow to deliver a world class experience for OSM developers.

The CI pipeline has been built with four stages (Figure 3).

- Stage 1 focuses on Gerrit which supports community collaboration such as code reviews. At this time, Gerrit does not have a mechanism to invoke multiple parallel pipelines from a single trigger. This first stage in the OSM CI/CD pipeline allows for a single Gerrit trigger (e.g. based on a code commit) to initiate a multi-branch pipeline defined in stage 2.

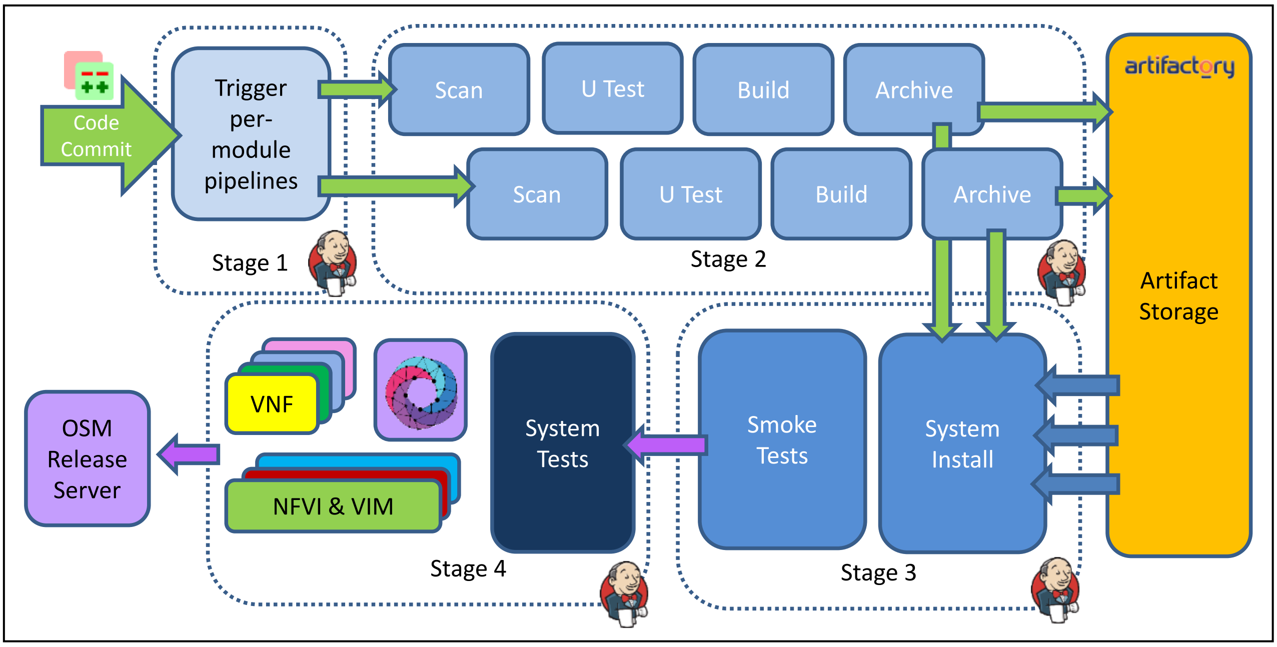

Figure 3: OSM CI/CD Pipeline

- Stage 2 includes a per-module pipeline where the testing work related to a specific module resides. All of the OSM modules now support packaging in Docker containers, and this stage operates tests within Docker containers and allows for parallel execution of Gerrit pipelines.

  Please note that Docker containers are not supported for OSM Release THREE installation.

  Module Development Groups (MDGs) only need to implement their respective call-backs for test, build and archive functionality. Stage 2 drives the automated code license scan, module specific unit tests, builds packages and archives the created artifacts. JFrog Artifactory was leveraged to manage artifacts outside of the master Jenkins node [Ref 1].

- Stage 3 focuses on system installation and smoke testing. The installation is based on the binary build artifacts created during stage 2 and from the artifact storage repository. Fresh OSM installs are created inside an LXC container for executing smoke tests. “Smoke” testing is performed as a simple and rapid set of end-to-end system oriented tests such as API checks and VNFD uploads that verify basic OSM functionality continues to work. The pytest tool introduces fixtures that provide a fixed baseline for reliably and repeatedly executing tests. Smoke testing does not depend on NFVI, VIMs or SDN Controllers.

- Stage 4 incorporates end-to-end system tests leveraging real NFV infrastructure, VIMs, SDN controllers and VNFs. It works in conjunction with ETSI’s Hub for Interoperability and Validation (HIVE) testing environment to enable testing on geo-distributed, multi-VIM environments.

One of the challenges with running extensive orchestration tests is the potential impact of hysteresis effects on the VIM and NFVI environment. To support the development of the CI/CD pipeline to leverage an ephemeral NFVI for repeatable MANO test runs, an emulator providing multiple Point-of-Presence (PoP) NFVI environments has been added to the DevOps repo.

3.1.2 User Interface Module (Design-Time Focus)

The two key parts of the User Interface (UI) Module that relate to a design-time focus are the VNF Package Generator and the VNF/NS Catalog Composer.

The VNF Package Generator is a tool that assists VNF providers to create a properly formed package for on-boarding in OSM.

The VNF/NS Catalog Composer is a model-driven Graphical User Interface that supports both VNF providers and an Operator's Network Service designers to rapidly develop descriptors that accurately represent the essence of the entity being modelled for deployments.

3.2 OSM Release Run-Time Components

3.2.1 User Interface Module (Run-Time Focus)

The Account Manager manages access credentials for the VIM environments.

Launchpad is the interactive GUI into the Run-Time system. It can be used to conveniently manage Lifecycle operations on VNFs and Network Services. Launchpad also provides real-time statistics for VNFs, network services, and a detailed view of compute and network topologies. Launchpad interfaces with the OSM system via the Northbound REST API.

The OSM Client provides a CLI client to remotely interact with OSM's Northbound REST API. It provides a python functional library to programmatically interact with OSM remotely.

3.2.2 Service Orchestrator Module

The API Service & Management Endpoint component is providing the primary API endpoint into OSM.

The Service Orchestration Engine is responsible for all aspects of service orchestration including lifecycle management and service primitive execution. It is effectively the "master" orchestration component in the system that governs the workflow throughout OSM. The Service Orchestration Engine is responsible for supporting the concepts of multi-tenancy, projects, users, and enforcing role-based access controls.

The Configuration Data Store is responsible for persistently storing the SO state, particularly in the context of VNF and NSD deployment records.

The Network Service Composition Engine is responsible for supporting network service and VNF descriptor composition. It validates that the composed Network Services and Virtual Network Function descriptors conform to the defined YANG schema.

The Catalog Manager is responsible for supporting the Create/Read/Update/Delete lifecycle operations on the defined VNF and NS descriptors and packages.

The Resource Orchestrator Plugin is responsible for providing an interface to integrate the Resource Orchestrator.

3.2.3 Network Service to VNF Communication Module

The Network Service to VNF Communication (N2VC) Module is responsible for the plugin framework between the SO and the VNF Configuration and Abstraction (VCA) layer.

3.2.4 VNF Configuration and Abstraction

The VNF Configuration and Abstraction (VCA) layer is responsible for enabling configurations, actions and notifications to/from the VNFs and/or Element Managers. When backed by Juju, it provides the facility to create generic or specific indirect-mode VNFMs, via charms that can support the interface the VNF/EM chooses to export.

3.2.5 Resource Orchestrator Module

The API Service & Utilities endpoint is responsible for providing the interface into the RO (for the SO to consume) and provides a number of utilities for internal to RO consumption.

The Resource Orchestration Engine is responsible for managing and coordinating resource allocations across multiple geo-distributed VIMs and multiple SDN controllers.

The VIM and SDN Plugins are responsible for connecting the Resource Orchestration Engine with the specific interface provided by the VIMs and SDN controllers.

3.2.6 Monitoring Module (Experimental)

One of the guiding principles for the OSM Monitoring Module (MON) is that it is required to interface with and leverage existing or new monitoring systems. The Monitoring Module is not intended to replicate or compete with those systems.

The Monitoring Module should mostly be considered as a tool for driving monitoring configuration updates to the external monitoring tool and as a conduit for steering actionable events into the Service Orchestrator. These actionable events may be either directly triggered by running NS/VNFs or deduced by the external monitoring tools.

One of the most powerful things OSM is delivering as a part of the Monitoring Module is the ability to correlate telemetry related to the VMs and VNFs to the relevant Network Services. Automated correlation is expected to provide a considerable user experience improvement to OSM users and drive up efficiency for operators in a Telecommunications environment.

Apache Kafka was used as the Monitoring Module message bus implementation. It is a fault-tolerant message passing system that supports a publish-subscribe model that aligns with the Monitoring Module's architecture. Messages sent to, or received from the Monitoring Module core will be passed via the message bus for both internal and external components of monitoring. Apache Kafka "topics" and "partitions" are used to segregate messages to MON.

The Monitoring Module is architected to support a flexible plugin method to integrate with the monitoring tool of choice. For Release THREE, OpenStack Aodh, OpenStack Gnocchi, Amazon CloudWatch, and VMware vRealize^{TM} Operations Manager tools are supported. The Monitoring Module Models sub-component contains the definition of alarms and metrics that OSM can process. The monitoring tool plugin is responsible for translating (aka normalizing) alarms and metrics from the innate format of the monitoring tool into the format that OSM interprets.

3.2.7 OSM Information Model Module

OSM is based on a model-driven architecture. The architectural direction has always been to use the same model as the basis of both the design-time capabilities and the run-time capabilities. However, in previous releases, some translation of the model was required on the interface between internal components. This internal model translation was not visible to and had no impact on OSM users, but did represent an area of inefficiency that the community was keen to address.

The OSM Information Model Module was created to be the single point of authority on the OSM data model that is leveraged by the different components. This helps the move towards a methodology where two of the most important data models in the system, the VNF Descriptor (VNFD) and the Network Service Descriptor (NSD), can be shared in their innate forms between components. OSM modules can act authoritatively on the relevant parts of the VNFD/NSD.

4 OSM Development Themes

The OSM community is working towards a target of production readiness and has identified a number of development themes that help to direct community innovation over multiple release cycles. While innovation is certainly not limited to these areas, having this focus has helped the community to deliver Release THREE on schedule.

On-going development themes are:

- On-boarding experience & VNF packaging to lower the barrier of entry for VNF providers.
- Simplified install and upgrade process to accelerate adoption and deployment combined with an improved development environment to facilitate an expansion of the developer community.
- Enhanced Platform Awareness (EPA) based resource allocation to facilitate high performance VNF deployments with lower Total Cost of Ownership for the operator.
- Dynamic configuration (Day 2 operation) of deployed services.
- Service Modelling to simplify, accelerate and standardize the design-time phase.
- Service Assurance to build towards automated assessment of key performance indicators with policy governance.
- Security including key management and Role Based Access Control.
- Resiliency for scalability and recovery.
- Multi-VIM support expanding OSM so that VMware vCloud Director, VMware Integrated OpenStack (VIO), multiple generations of OpenStack, OpenVIM and Amazon Web Services Elastic Compute Cloud are enabled.
- Multi-Site support to enable automated service delivery across multiple sites where a site is represented as a grouping of infrastructure managed by a VIM.

5 OSM Release THREE Overview

OSM Release THREE has made significant steps in advancing on each of the themes noted above. However, there is one theme in particular that is worthy of further mention in the context of a Release THREE overview, i.e. production readiness.

The very explicit goal of the community is to enable production-ready deployments in operator networks. OSM Release THREE represents another significant milestone along this path. The bar the community is setting for itself before applying a “deployment ready” label is high. Nonetheless, the leadership team feel comfortable in advising operators that they should now consider this release to be of sufficient capability as a framework for PoCs and field trials. Operators can also use OSM Release THREE to progress their RFx processes.

The full list of distinct capabilities that haves been progressed can be found on the OSM website and WIKIs. The following section notes the salient categories of innovation that articulate the cohesion for this release.

5.1 Security

One of the challenges of deploying an NFV based network service is to make sure the MANO environment is compatible with the organizational structures that exist in the operator environment. It is required that the visibility into the network and the control of network operations presented to different actors employed by the operators can be suitably configured to match their roles in the organization. This separation of visibility/control of the network is required for secure and robust operation of the network.

To address these requirements, OSM Release THREE has added comprehensive Role-Based Access Control and Multi-Tenancy/Project to the interface model.

5.2 Service Assurance

An assured network service delivery environment requires the ability for running network services that can scale-out their support level, with of course the ability to scale-in if the need for the additional capacity is no longer required. OSM Release THREE includes support for network scaling events to add and remove full VNF instances from a running Network Service.

5.3 Resiliency

A resilient service delivery platform is required for production readiness. OSM Release THREE has made considerable progress on this topic by improving the recovery on single component failure, supporting multiple VCA instances and by offering improved scalability of the OSM platform.

5.4 Usability

OSM continues to focus on being an easy to use MANO platform. The Python based OSM client offers a straightforward method to interact with the most commonly used OSM operations. Most VNF consoles are now accessible via the GUI. And, once installed, the user can upgrade to maintenance releases without re-installation.

5.5 Interoperability

One of the guiding principles of OSM is that each component is both replaceable and pluggable. To that end, Release THREE has taken another substantial step forward to drive interoperability with other components such as VNFs, VIMs, SDN controllers and monitoring tools.

At the VIM level, the Amazon Web Services EC2 plugin was extended, further improvements were added to the VMware vCloud Director VIM plugin compatible with vCloud NFV 1.5 and 2.0, and support for VMware Integrated OpenStack (VIO) was delivered.

At the SDN controller level, the SDN controller support received further incremental refining for ONOS as well as the OpenDaylight (ODL) and Floodlight.

For service assurance, the new monitoring plugin framework enabled Amazon CloudWatch, VMware vRealize Operations Manager, OpenStack Aodh and OpenStack Gnocchi monitoring tools.

5.6 Release THREE Community Highlights

The beating heart of OSM is in the wonderful community. OSM have been delighted to welcome so many new members and participants to the project, with 81 companies now signed up before Release THREE completed.

For Release THREE, OSM will continue to recognize outstanding contributors to the community. The individuals awarded will be announced at the OSM Release Four plenary and released on the OSM website at: https://osm.etsi.org/.

6 OSM Release THREE Details

This section presents a more detailed view on a number of the advancements that have been achieved with OSM Release THREE.

6.1 Security Details

Security is often a topic that is considered to be synonymous with access authorization or data/user plane network security. For OSM, the community has considered security to represent a much broader and more nuanced set of topics. For Release THREE the focus has been on providing extensions related to access control security.

6.1.1 Role-Based Access Control (RBAC)

The NFV Orchestrator requires a significant set of capabilities and privileges to perform all its required tasks such as VNF on-boarding, NS design & on-boarding, NS deployment, day-2 operations, NS shutdown, addition of new datacenters/VIMs, etc. However, not all of these tasks are expected to be performed by the same user or type of user in the organization. Each of the stages in the lifecycle of a VNF or NS may have different implications in terms of service continuity, validation, license consumption, access to credentials, etc.

OSM Release THREE allows the definition of different roles, defined by admin user, with different sets of privileges. All users are mapped to at least one of these roles.

6.1.2 Multi-Project

In OSM, the concept of multi-project access control is closely related to RBAC. A “project” is used to group things such as VNFDs, NSDs, NS instances, and datacenters (VIMs). The RBAC can be applied on the project definition to apply consistent access to the defined grouping of resources that OSM manages.

In OSM, users have roles and can belong to more than one project. Each user is associated with one or more projects and each user has one or more roles on each project. The role is system-wide and is defined based on permissions on API endpoints.

Permissions at some VIMs are restricted by the admin of the infrastructure. This is particularly true of public cloud infrastructure. In general, OSM would not have sufficient rights to create or edit tenants at VIM level. The use of the project definition enables more flexibility in grouping access to resources than if OSM allowed a VIM tenant view to permeate up the MANO stack.

During the creation of OSM projects, the admin should indicate in which datacenters the new OSM project is authorized. Per datacenter, the credentials and the VIM tenant should be provided (of course, OSM projects might be able to share the same VIM tenant).

6.2 Service Assurance Details

6.2.1 Network Service Scaling

Network Service Scaling was released as an experimental feature for Release TWO. Some updates to the codebase combined with additional testing for the new functionality means OSM Release THREE can now support Network Service scaling use cases. A Network Service scaling use case is defined as support for adding and/or removing full VNFs to/from a running Network Service.

The Scaling Groups construct within the Network Service Descriptor can be used to identify VNFs that need to be scaled simultaneously. A Scaling Group contains one or more VNFs. A scaling action is performed on the Scaling Group. The added VNFs instances are attached to the same network as the initial instances of the VNFs.

The scaling action is triggered manually by the network operator using the UI or the OSM Northbound API.

6.2.2 Explicit Port Ordering Support for VDUs

Consider a use case where a VDU is providing firewall functionality and has two interfaces. The VDU needs to determine which of two interfaces it should configure as the external (possibly WAN) connection and which is the internal (possibly LAN) connection. VDUs that have multiple interfaces, such as in this example, often require that interfaces allocated to the VDUs are provided in a consistent order. The VDU uses this ordering convention to select the appropriate interface for WAN/LAN configuration.

In OSM Release THREE, it is now possible to specify ordering of interfaces (both external and internal) so that the VM is orchestrated with that specific ordering. The interface ordering will be preserved according to the combination of the following two criteria:

- Respect the order of port definitions at VNFD, which should be taken as the default order to be passed to the VIM.
- Optionally, explicit assignment of PCI addresses per port.

6.2.3 Affinity and Anti-Affinity Rules For VNF Deployment

Different VDUs of the same VNF often require the ability to be deployed in different parts of the same datacenter. This behavior is often required by VNFs that implement active-standby resiliency strategies that are more compelling when the VDUs are separated based on some characteristic. VIM environments allow for resources to be logically partitioned, such as with OpenStack “Availability Zones”.

OSM Release THREE has added the ability to deploy VNFs in different regions of the datacenter based on the logical separation of the availability zones articulated with the data model.

6.2.4 Allow Concurrent Access to VIMs

OSM Release THREE has been extended to support concurrent actions over more than one datacenter connection. This extension allows OSM to overcome problematic situations due to VIM response time, e.g. long response times in geo-distributed deployments, or missing VIM responses due to communication problems.

OSM THREE includes asynchronous VIM operations support in OSM components to prevent potential side-effects on Service Management in multi-site deployments.

6.2.5 Monitoring (Experimental)

The Monitoring Module is new for OSM Release THREE. Its architecture has been introduced at a high level in section 3.2.6 above. This section will describe the alarms and metrics that are supported in Release THREE and made available up to the Kafka consumer.

The MON Models sub-component contains the definition of alarms and metrics that OSM can process. The monitoring tool plugin is responsible for translating (aka normalizing) alarms and metrics from the innate format of the monitoring tool into the format that OSM can interpret.

Table 1 shows the normalized metrics that OSM uses and their corresponding monitoring tool metrics.

Normalized Metric Name	Unit	VMware vROPs Metric	Amazon CloudWatch Metric	OpenStack Metric
AVERAGE_MEMORY_UTILIZATION	%	mem|usage_average	Not Supported	Memory.total (%)
READ_LATENCY_<DISK_NO>	msec	virtualDisk|totalReadLatency_average	Not Supported	Not Supported
WRITE_LATENCY_<DISK_NO>	msec	virtualDisk|totalWriteLatency_average	Not Supported	Not Supported
DISK_READ_OPS	Nos	Not Supported	DiskReadOps	Disk_ops.DISK
DISK_WRITE_OPS	Nos	Not Supported	DiskWriteOps	Disk_ops.DISK
DISK_READ_BYTES	Bytes or bytes/sec	Not Supported	DiskReadBytes	Disk_octets.DISK
DISK_WRITE_BYTES	Bytes or Bytes/sec	Not Supported	DiskWriteBytes	Disk_octets.DISK
PACKETS_DROPPED_<NIC_NO>	Nos	net|dropped	Not Supported	if_dropped.INTERFACE
PACKETS_RECEIVED	Nos	net:Aggregate of all instances|packetsRxPerSec	NetworkPacketsIn	if_packets.INTERFACE
PACKETS_SENT	Nos	net:Aggregate of all instances|packetsTxPerSec	NetworkPacketsOut	If_packets.INTERFACE
CPU_UTILIZATION	%	cpu|usage_average	CPUUtilization	Percent.virt_cpu_total

Table 1: OSM Normalized Metrics

Table 2 shows the normalized alarms that OSM uses and their corresponding monitoring tool metrics that are used to support triggering these alarms.

Normalized Alarm Name	VMware vROPs Metric	Amazon CloudWatch Metric	OpenStack Metric
Average_Memory_Usage_Above_Threshold	Average_Memory_Usage_Above_Threshold	Not Supported	Average_Memory_Usage_Above_Threshold
Read_Latency_Above_Threshold	Read_Latency_Above_Threshold	Not Supported	Not Supported
Write_Latency_Above_Threshold	Write_Latency_Above_Threshold	Not Supported	Not Supported
Disk_read_ops_above_threshold	Not Supported	Disk_read_ops_above_threshold	Disk_ops.DISK
Disk_write_ops_above_threshold	Not Supported	Disk_write_ops_above_threshold	Disk_ops.DISK
Disk_read_bytes_above_threshold	Not Supported	Disk_read_bytes_above_threshold	Disk_octets.DISK
Disk_write_bytes_above_threshold	Not Supported	Disk_write_bytes_above_threshold	Disk_octets.DISK
Net_Packets_Dropped	Net_Packets_Dropped	Not Supported	Net_Packets_Dropped
Packets_in_Above_Threshold	Packets_in_Above_Threshold	Packets_in_Above_Threshold	if_packets.INTERFACE
Packets_out_Above_Threshold	Packets_out_Above_Threshold	Packets_out_Above_Threshold	if_packets.INTERFACE
CPU_Utilization_Above_Threshold	CPU_Utilization_Above_Threshold	CPU_Utilization_Above_Threshold	CPU_Utilization_Above_Threshold

Table 2: OSM Normalized Alarms

6.2.6 VNF SW Upgrade (Experimental)

Once a Network Service is running in a production environment it is likely that some nodes (VNFs) may require a SW upgrade. OSM supports a method to run these SW upgrades by enabling this lifecycle operation to be described via the VNFD.

For OSM Release THREE, this is an experimental capability that will be enhanced to meet the production ready requirement that OSM has applied to other features.

6.3 Resiliency

6.3.1 OSM Platform Resiliency to Single Component Failure

The NFV Orchestrator is a critical component for the operator in a production environment. As such, it should be capable of recovering from unexpected failures of its components.

After recovering from single component failures the system should not impact existing running services.

For Release THREE, OSM is leveraging the container deployment model of OSM to take advance of container restart capabilities to improve resiliency. Where OSM components require state to be maintained the architecture has been updated to promote the use of external (to the component) state management.

This approach has allowed OSM to significantly improve resiliency in Release THREE.

6.3.2 Support of Remote or Multiple VCA Instances

For Release TWO, OSM assumed that the management networks of all the VNF instances were remotely accessible by OSM from outside the datacenter, either directly or via floating IPs, so that VCA could drive successfully its operations with the VNFs. While this architecture has many advantages, in telco clouds it was considered to be too rigid, especially in the context of hybrid deployments that include public clouds.

In order to provide more flexibility for those use cases, Release THREE has been enhanced to allow OSM to work with more than one VCA instances simultaneously. The VCA instances can be in charge of specific datacenters and, can even be deployed inside them and attached to the management network.

6.3.3 Improved Scalability of OSM Platform

In order to support growing demands of dynamic network deployments, OSM needs to be capable of increasing its own resources via scaling-out OSM components.

Each OSM component is making progress on improving its scalability. For Release THREE, there have been significant improvements in the SO internal mechanisms to allow running multiple instances. Each instances of the SO management agent has its own endpoint. State is available across all SO instances as messages are routed via the Distributed Transaction System so it does not matter which SO instance the user connects to. If a single endpoint is required, a load balancer can be placed in front of the system. The default SO deployment configuration supports an active-passive-passive redundancy schema. In addition, the interaction with VCA, via the N2VC module now supports asynchronous operations.

6.4 Usability

6.4.1 Clean-up of OSM's Northbound API

The OSM architecture has a northbound REST API that is intended to allow the invocation of the main actions by external systems such as an OSS. This API is intended to support all the operations that OSM provides, and should be the unique entry point for all the interactions with the system (OSS, UI and CLI). The SO is in charge of triggering all the subsequent actions and requests to the rest of OSM components.

However, previous releases had some instances of non-compliance with this architecture where some Northbound API operations required direct access to VCA or RO APIs. This internal behavior resulted in some impacts for the user such as expanding the number of ports that needed to be opened on firewalls.

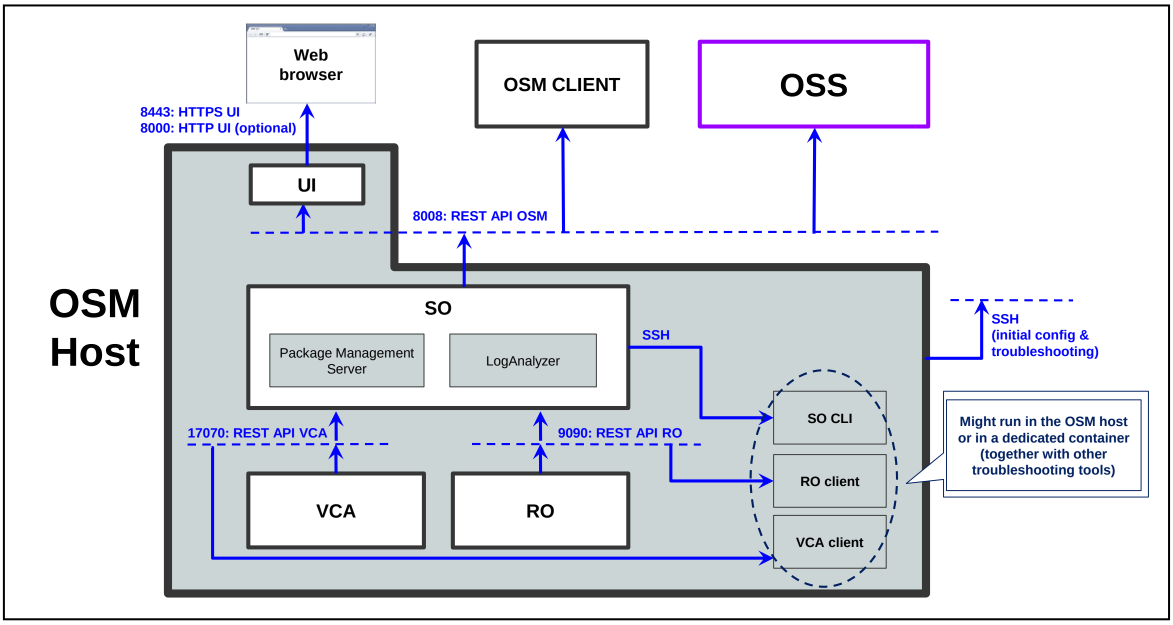

Figure 4: OSM Northbound API Clean-up

For Release THREE OSM consolidated significantly more of the functionality under a single Northbound API entry point as depicted in Figure 4 which results in a considerable simplification of the deployment model.

6.4.2 Detailed Feedback When Deploying and Configuring

To promote faster deployment and fault detection when deploying network services, OSM Release THREE has increased the level of detail and correlation in logs and feedback. This allows more effective root cause analysis of errors/failures in deployment and configuration, and helps troubleshooting them.

When deploying a NS, configuring it or invoking a service primitive, the SO logs and UI presentation now offer more detailed feedback in case of error. In addition, incremental status updates on the progress of the deployment or configuration tasks are now available in the UI.

6.4.3 Unified CLI Client

The OSM Client provides a Command Line Interface (CLI) client to remotely interact with OSM's Northbound REST API. A Python functional library also allows to programmatically interact with OSM remotely via REST. For Release THREE, the client has been extended to support many of the updates to the data model, and the Northbound API clean-up noted in section 6.4.1.

6.4.4 UI Separated From SO

For OSM Release TWO, the installer created a single container for the User Interface (UI) and Service Orchestrator (SO) components. However, in some deployments it could be convenient to separate the UI and SO modules, which would also allow to scale each component independently.

This separation has been implemented for Release THREE, and now the one-click installation tool supports a deployment model where the UI and SO are deployed in separate containers.

6.5 Continuous Integration/Continuous Testing

6.5.1 SW Upgrade of the OSM Platform

The NFV Orchestrator, as any other complex system in production, is expected to require regular software upgrades, including bug fixes, security patches or new functionality. A mechanism is needed to guarantee that SW upgrades in the platform do not disrupt service continuity, and provide a safe rollback mechanism in case of failure allowing the system to restore from its last known internal state.

Release THREE represents a big step forward on the path to enabling seamless upgrades of OSM. The RO component has been updated to support functionality such as database rollback upon upgrade failure and the VCA has been updated to support Debian packages upgrades.

6.6 Interoperability

6.6.1 OpenStack v3 API

The OSM OpenStack VIM connector has been updated to include support for the Keystone v3 API, specifically the v3.3 version. For Release TWO, this functionality was considered experimental. For Release THREE the additional testing that has been carried out has given the community sufficient confidence to remove the experimental tag.

Keystone is the OpenStack project that is responsible for managing identity and authorization between OpenStack projects. As of the Mitaka OpenStack release, the Keystone project marked the v2 API as deprecated with support for it expiring with the Queens release (expected H1/2018).

The Keystone v3 API, as with other v3 APIs in OpenStack, has introduced a micro-versioning strategy that facilitates further innovation in the OpenStack project and enables related API updates. The v3 API also brings important changes in terminology. The “tenant” concept has been replaced by the term “project”. Collections of users are defined using “Groups”. Projects, users and groups can be described in the context of “Domains” where the names only have to be unique within their owning domain.

6.6.2 OSM Remote Labs Network

The OSM Remote Labs Network has been enabled by the secure ETSI remote lab testing infrastructure framework with the Hub for Interoperability and Validation and ETSI (HIVE). This remote lab testing framework allows for instances of OSM running in the ETSI hosted lab to connect securely over VPN tunnels to remote OSM labs. These remote labs run different types and instances of NFVI + VIM environments contributed by the community.

HIVE is a fundamental component of the ETSI NFV Plugtests infrastructure that allows to run interoperability testing among remote implementations. Now, at the core of the OSM Remote Labs Network, HIVE is also a key enabler of the OSM CI/CD process where remote labs running NFVI+VIM environments are permanently and securely connected (see Figure 5). HIVE helps to ensure that OSM inter-operates successfully with multiple VIMs, SDN Controllers and NFV Infrastructure while helping to reduce barriers for community engagement.

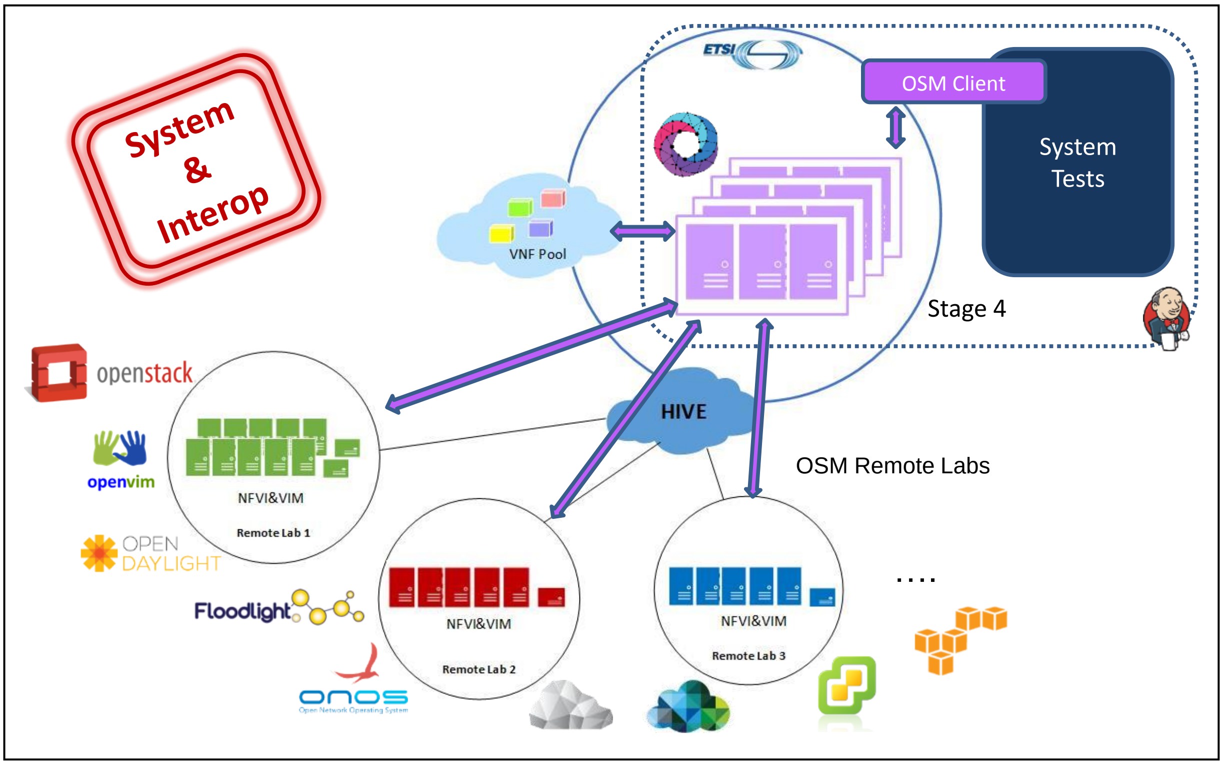

Figure 5: OSM Remote Labs Network

During the Release THREE development cycle, the OSM community welcomed the addition of a new VMware Integrated OpenStack (VIO) based remote lab environment. This augments the comprehensive remote testing infrastructure that already includes a VMware vCloud Director based VIM environment, an OpenStack Newton based environment and a Wind River® Titanium Server™ based environment. These labs are available for the OSM community to leverage as a part of the OSM CI/CD testing pipeline.

6.6.3 Multi-PoP NFVI/VIM Emulation Platform

The landscape for deploying a NFV service is typically complex. The infrastructure to support the service deployment are geo-distributed with different types and sizes of datacenters available at different locations in the network. For a VNF or a MANO provider, the lack of multiple VIM and NFVI environments for their development teams to use may be a considerable inhibitor to testing complex deployment scenarios as required in production networks. In fact, for many, the VIM/NFVI environment may be a contested resource within the company which leads to reduced time available to test deployment scenarios.

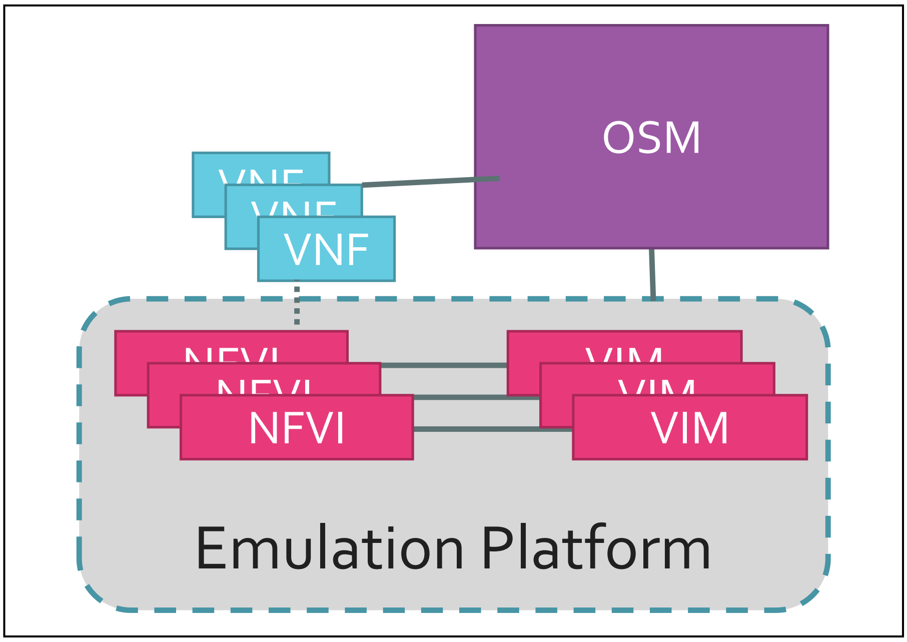

Figure 6: Multi-PoP NFVI+VIM Emulation Platform

A multi-PoP infrastructure emulation platform is included in Release THREE (Figure 6). This emulation platform provides a sandbox environment for NFV prototyping, on-boarding activities with OSM, and testing of OSM components as a part of the DevOps CI/CD testing models.

6.7 Data Modelling Details

One of the greatest challenges facing the entire community of technologists developing specifications, standards and implementations for NFV relates to the topics of information and data modelling of the NFV solution space.

The OSM community is committed to supporting Network Service Descriptor (NSD) and Virtual Network Function Descriptor (VNFD) model alignment in the industry. We believe that having an industry align around a Common Information Model will benefit everyone involved in this space. An industry agreed Common Information Model should help:

- Reduce the effort for VNF vendors to be on-boarded in different MANO offerings.
- Minimize on-boarding time.
- Promote a more open ecosystem for VNF vendors to participate in.
- Facilitate a switch of development effort from custom on-boarding effort to other value-add efforts such as improved data plane performance, security and service assurance.
- Progress interoperability and portability.
- Lower the barrier for entry of innovative offerings.
- Lower operator costs.

A Common Data Model is considered to be a desired end-state for modelling alignment, further progressing the Information Model alignment activities in the industry. Data Model alignment will involve attribute alignment, encoding alignment and packaging alignment. While this very much represents an agreeable goal, the industry is likely to have to progress through a number of steps to get there.

OSM is following a model-driven design methodology. This enables OSM to upgrade to newer models with minimal impact on the system. For Release THREE, the OSM NSD and VNFD Data Models continue with broad alignment with the ETSI NFV phase 1 MANO NSD and VNFD Information Models [Ref 1].

During the development of this release, the community assessed the ETSI NFV IFA011 VNFD [Ref 3] and ETSI NFV IFA014 NSD [Ref 4] Information Models. To support industry initiatives to work towards a Common Information Model, the OSM community has submitted feedback to the ETSI NFV ISG Interfaces and Architecture (IFA) Working Group. This feedback is being actively developed in conjunction with IFA with a view to adding positive advancements to the phase 2 Information Models via maintenance releases and will also be considered as part of the ETSI NFV phase 3 development activities.

7 Official Proofs of Concept

During the OSM Release THREE development cycle the OSM community created an official OSM Proof of Concept Framework [Ref 5]. OSM members and users are encouraged to run Proofs of Concept (PoCs) relevant to showcase the value of OSM and to identify opportunities for further development of OSM.

7.1 OSM PoC#1: DevOps in Service Chains & 5G Network Slices

7.1.1 PoC#1 Abstract

In order to successfully compete in the market, communications service providers (CSPs) must offer hyper scale, dynamic services that can be quickly configured and deployed while continuing to meet agreed-upon levels of service and security. Secondly, service-oriented network slicing is one of the key technologies to meet the 5G vision. To this end, Arctos Labs, Netrounds and RIFT.io have developed ETSI OSM PoC#1 with Telenor on Intel® Architecture to showcase a solution that embodies management, orchestration, and testing of 5G network slices in a virtualized environment [Figure 7]. The PoC, which uses – amongst other software components – Open Source MANO (OSM) was exhibited at Mobile World Congress 2017. The main message of the PoC is that end-to-end service validation and assurance orchestrated with automated and software-based (open source) tools is essential to ensure customer experience in the new agile world.

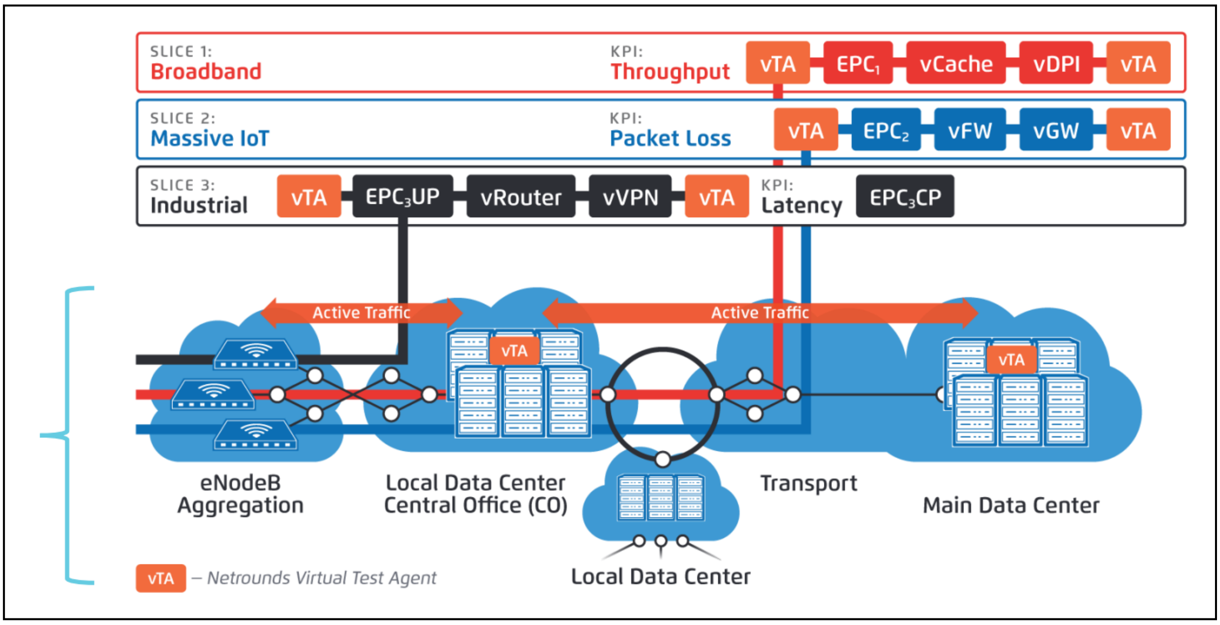

Figure 7: Assuring Expected QoS Levels with Active Measurement*

** Figure extracted from OSM PoC#1 Overview [Ref 6]*

7.1.2 PoC#1: Key Takeaways

Here are the key takeaways from this PoC activity:

- Active service testing and monitoring of individual slices that is orchestrated and fully automated is needed not only during 5G network slice deployment but also during the life cycle of the slices.

- Virtual Test Agents should be included in the network service design (NSD) to enable automated and active service assurance.
- DevOps approaches need to be introduced to enable automated service assurance.
- 5G network slice deployments require different yet critical SLAs that must be defined in the network service model to be monitored and actively measured during life cycle of the network slice.
- Active monitoring of individual slices allows for proactive and automated troubleshooting and self-healing, thus making the self-organizing network (SON) concept end to end.

7.2 OSM PoC#2: OSM with VIO in MEC Architectures

7.2.1 PoC#2 Abstract

Multi-Access Edge (MEC) computing is rapidly becoming a requirement in NFV and IoT architectures for service providers. Only by pushing workloads to the edge of the network is it possible to offer the next gen 5G and IoT services with low latency requirements in an infrastructure that is flexible and robust enough to support them.

POC#2 showcases leveraging OSM Release THREE as an orchestrator, and VMware Integrated Openstack (VIO) as the VIM layer to deploy and maintain edge services in several different use cases that make use of the same hardware footprint, as well as discuss the overall infrastructure topology. These use cases include video transcode at the network edge with ffmpeg, using Fortinet as a Serving Gateway (sGW) in virtual Radio Access Network (vRAN) edge scenarios, SD-WAN with Vyatta vRouter and operational intelligence for distributed architectures. Further details are available in the PoC#2 Overview [Ref7].

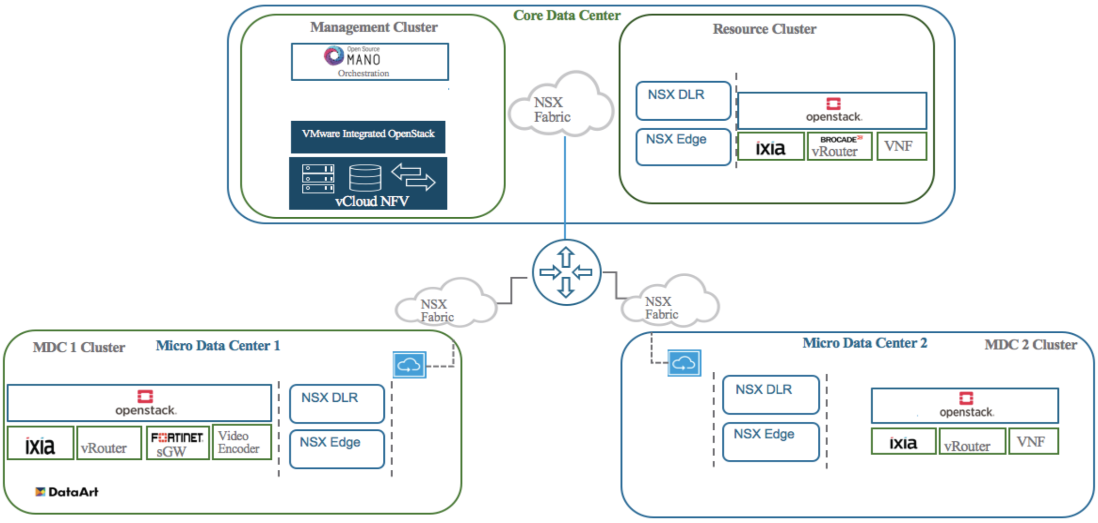

Figure 8: Multi-Access Edge Computing Logical Topology (*)

** Figure extracted from OSM PoC#2 Overview [Ref 7]*

7.2.2 PoC#2: Key Takeaways

The key takeaways from this PoC activity are:

- OSM can be leveraged for distributed multi-datacenter service scenarios.
- OSM is compatible with VMware Integrated OpenStack version 4.0 (Ocata) with NSX as the SDN controller.
- Multi-Access Edge Computing scenarios are possible with OSM and Ocata based OpenStack.
- End to end service monitoring is possible using existing tools.

8 References

1. JFrog Artifactory - https://www.jfrog.com/artifactory/
2. ETSI GS NFV-MAN 001 V1.1.1 (2014-12)
3. ETSI GS NFV-IFA 011, V2.3.1 (2017-08), http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/011/02.03.01_60/gs_NFV-IFA011v020301p.pdf
4. ETSI GS NFV-IFA014 V2.3.1 (2017-08), http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/014/02.03.01_60/gs_NFV-IFA014v020301p.pdf
5. OSM PoC Framework and accepted PoCs: https://osm.etsi.org/wikipub/index.php/OSM_PoCs
6. OSM PoC#1 Overview: https://osm.etsi.org/wikipub/images/5/5e/ETSI_OSM_PoC_1.pdf
7. OSM PoC#2 Overview: https://osm.etsi.org/wikipub/images/c/c3/OSM_POC_2_Overview.pdf

ETSI (European Telecommunications Standards Institute)
06921 Sophia Antipolis CEDEX, France
Tel +33 4 92 94 42 00
info@etsi.org
www.etsi.org

This White Paper is issued for information only. It does not constitute an official or agreed position of ETSI, nor of its Members. The views expressed are entirely those of the author(s).

ETSI declines all responsibility for any errors and any loss or damage resulting from use of the contents of this White Paper.

ETSI also declines responsibility for any infringement of any third party's Intellectual Property Rights (IPR), but will be pleased to acknowledge any IPR and correct any infringement of which it is advised.