Open Source MANO

An ETSI OSM Community White Paper

OSM RELEASE TWO

A TECHNICAL OVERVIEW

April 2017

Authors:

Adrian Hoban, Chair OSM TSC, Intel

Alfonso Tierno Sepúlveda, OSM Resource Orchestration MDG, Telefónica

Francisco Javier Ramon Salguero, OSM Chair, Telefónica

Gerardo García de Blas, OSM TSC, Telefónica

Kiran Kashalkar, OSM User Interface MDG, RIFT.io

Marco Ceppi, OSM VNF Configuration and Abstraction MDG, Canonical

Mark Shuttleworth, OSM TSC, Canonical

Matt Harper, OSM TSC, RIFT.io

Rajesh Velandy, OSM NW Service Orchestration MDG, RIFT.io

Silvia Almagia, OSM, ETSI

Vanessa Little, OSM TSC, VMware

Editor:

Chris Buerger, OSM Marketing Task Force Chair, Intel

Contents

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 in 2017. OSM Release TWO represents a 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 TWO substantially enhances interoperability with other components (VNFs, VIMs, SDN controllers) 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 TWO 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 OSM open source project, the output of this modelling work has been contributed to ETSI NFV. Release TWO also provides extremely flexible VNF configuration and advanced networking management as well as improved troubleshooting capabilities, with advanced logging.

This White Paper outlines the main architecture of OSM and the new capabilities developed and open-sourced as part of Release TWO. 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 TWO 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 TWO possible.

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)).
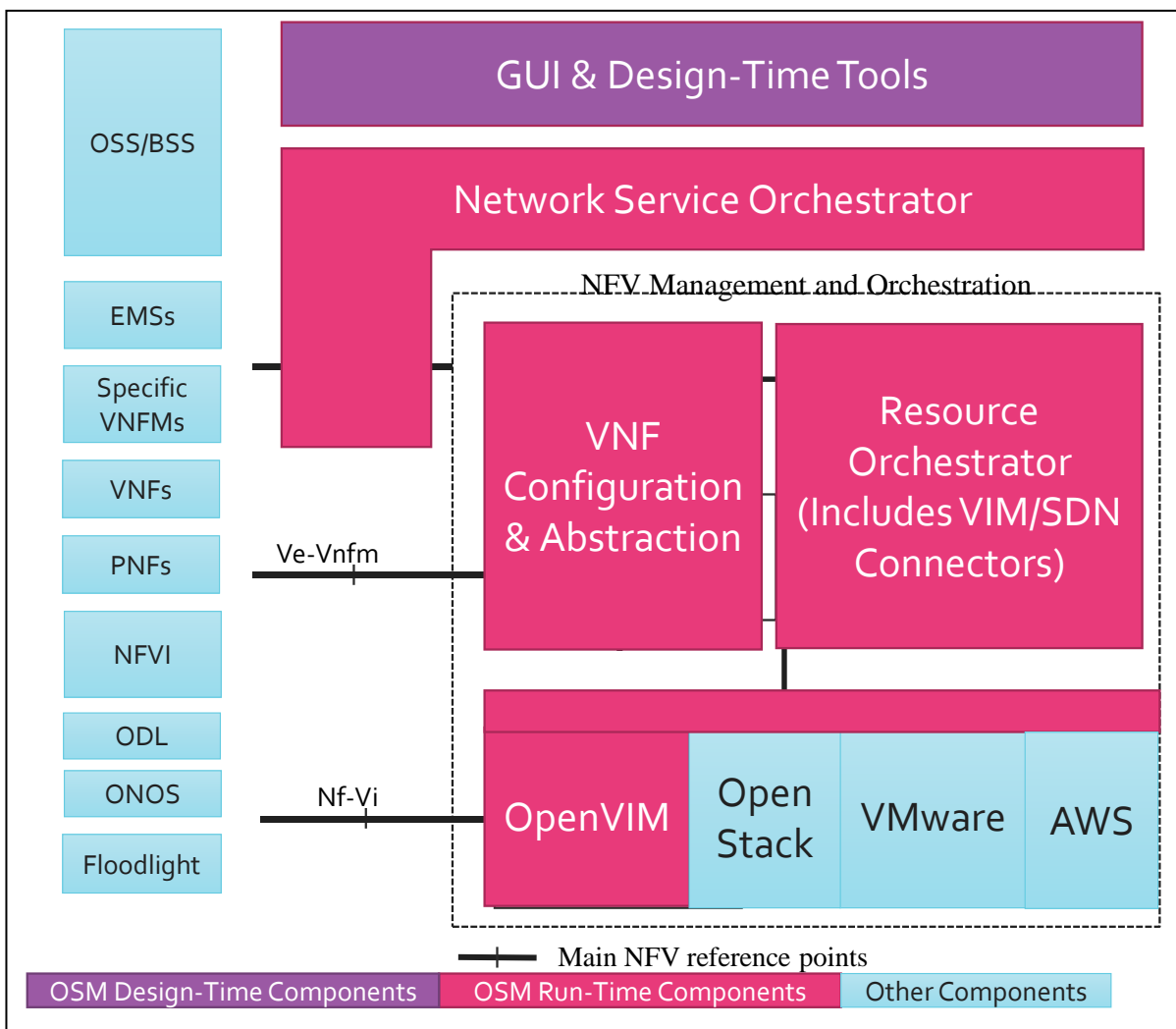

Figure 1 OSM Mapping to ETSI NFV MANO

Run-Time Scope

The run-time scope of OSM includes:

- Anautomated 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.

- 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, Client and REST interfaces to enable access to all features.

Design-Time Scope

The design-time scope of OSM includes:

- Delivery of a capability for Create/Read/Update/Delete (CRUD) operations on the Network Service Definition.

- Supporting a Model-Driven environment with Data Models aligned with ETSI NFV MANO.

- Simplifying VNF Package Generation.

- Supplying a Graphical User Interface (GUI) to accelerate the network service design time phase.

OSM Development Themes

The OSM community has identified a number of development themes that are used 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 TWO on schedule.

On-going development themes are:

- On-boarding experience & VNF packaging to lower the barrier of entry for VNF vendors.

- Simplified install and upgrade processes to accelerate adoption and deployment combined with an improved development environment to facilitate an expansion of the developer community.

- EPA-based resource allocation to facilitate high performance VNF deployments with lower Total Cost of Ownership (TCO) for the operator.

- Dynamic configuration (Day 2 operation) of deployed services.

- Service Modelling to simplify, accelerate and standardize the design-time phase.

- Multi-VIM support expanding OSM so that VMware, multiple versions 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.

The OSM community are working towards a target of production readiness in 2017.

OSM Release TWO Overview

OSM Release TWO 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 TWO overview, i.e. the production readiness in 2017.

The very explicit goal of the community is to enable production-ready deployments in operator networks. OSM Release TWO represents a 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 feels comfortable in advising operators that they should now consider this release to be of sufficient functionality as a framework for PoCs and trials. Operators can also use OSM Release TWO to progress their RFx processes.

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

Interoperability

One of the guiding principles of OSM is that each component is both replaceable and pluggable. To that end, Release TWO has taken another substantial step forward to drive **interoperability** with other components (VNFs, VIMs, SDN controllers). At the VNF level, there was a significant increase in the number of on-boarded VNFs. At the VIM level, an **Amazon Web Services EC2 plugin** was created and significant improvements in the **VMware vCloud Director VIM plugin** compatible with vCloud NFV 1.5 and 2.0 have been delivered. In addition, the **SDN controller** support now includes ONOS as well as the previously supported ODL and Floodlight and they can now be managed directly from the Resource Orchestrator (RO).

Packaging, Distribution & Installation

A significant portion of the development effort within the OSM technical community has focused on responding to input from developers and operators to greatly enhance the **user experience**, both in terms of usability and installation procedure. As a result, there have been reductions in the OSM resource footprint, the addition of a Docker image distribution method, as well as package management and related UI updates.

Usability

OSM continues to focus on being an easy to use MANO platform. A new 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 update to maintenance releases without re-installation.

Data Modelling

One of the greatest challenges facing the entire community of technologists developing specifications, standards and implementations for NFV relates to the topic of information and data modelling of the NFV solution space. For Release TWO, the OSM community has developed further **enhancements to modelling** of VNFs and Network Services. Based on the implementation experience gained in developing OSM Release TWO, the community is collaborating with the ETSI NFV Industry Specification Group (ISG). Formalized feedback is

being provided to help harmonize and progress the VNF Descriptor and Network Service Descriptor (NSD) definitions.

Data Plane Configuration

OSM Release TWO has introduced powerful underlay network management functionality that can be used to configure E-LINE and E-LAN configurations. For greater flexibility and functionality, OpenVIM now also includes support for Open vSwitch.

Service Assurance

An assured network service delivery environment requires the ability for running network services to be able to scale-out their support level as well as the ability to scale-in if the need for the additional capacity is no longer required. OSM Release TWO includes experimental support for network scaling events to add and remove full VNFs from a running Network Service.

Release TWO 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 65 companies now signed up before Release TWO completed.

For Release TWO, OSM also held the inaugural awards. Three members of the community were acknowledged for their fantastic contributions to the community during the release TWO cycle:

Outstanding Community Development:

- Kaela Loffler (Netrounds)

Outstanding Technical Contributors:

- Mike Marchetti (Sandvine)
- Jokin Garay (Keynetic)

OSM Release TWO Details

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

Interoperability Details

ETSI NFV Plugtests

The ETSI Centre for Testing and Interoperability organized the 1st NFV Plugtest in January 2017 [Ref 1]. This event was focused on interoperability testing between the grouping of Network Function Virtualization Infrastructure (NFVI) and Virtual Infrastructure Manager, the Management and Orchestration (MANO) layers (logically composed of the Virtual Network Function Manager (VNFM) and NFV Orchestration (NFVO)) and the Virtual Network Functions (VNF). Over 31 organizations and 160 engineers were involved in running 160 interoperability test sessions combining 15 VNFs, 9 MANO solutions and 11 NFVI+VIM platforms.

The OSM community participated in this event as one of the MANO solutions. The OSM VNF Descriptor (VNFD) and Network Service Descriptor (NSD) Data Models were leveraged to successfully on-board all of the VNFs participating in the Plugtest with all of the participant NFVI+VIM platforms.

In keeping with OSM's mantra of delivering production worthy code, the bugs in the Release ONE codebase that were identified during the Plugtest have now been resolved for in Release TWO.

The opportunity to perform interoperability tests with so many different NFVI+VIM platforms and VNFs helped the OSM community to prioritize a number of features for Release TWO.

Amazon Web Services EC2

To enable public cloud support, an OSM VIM plugin was developed for Release TWO to support Amazon Web Services (AWS) Elastic Compute Cloud (EC2) and Amazon Virtual Private Cloud (VPC) products. Supporting the EC2 APIs enables OSM users to leverage compute capacity in the AWS public cloud and the use of the VPC APIs enables OSM users to leverage logically isolated sections of the AWS public cloud.

This new capability allows for automated NFV deployment to public cloud. For Release TWO, the OSM community believe that this functionality will be particularly useful for developers and testers of OSM that wish to run basic NFV tests and use cases without the need to set up their own private cloud environment.

As noted above, the OSM Resource Orchestrator (RO) has the capability to support multiple instances and multiple types of VIMs simultaneously. By adding AWS support to the repertoire of supported VIMs, this also opens the door to hybrid private/public cloud NFV deployment scenarios with a future potential for cloud-bursting use cases.

With the creation of yet another VIM plugin, the OSM community has also confirmed the flexibility of the OSM VIM plugin model while helping to ensure that the OSM Data Models remain agnostic to the underlying VIM platforms.

Multi-Disk Support

Some VNFs require the ability to support multiple disks per Virtual Deployment Units (VDUs) where a disk is some form of mass storage device.

Two new capabilities have been added to Release TWO.

1. Support for creating ephemeral empty disks via the VIM.

2. Support for new volumes based on an existing cloud image.

Based on the experience from the ETSI NFV Plugtest, the OSM community believes that these three configuration options for multiple disk support should be sufficient to support this aspect for the majority of VNFs.

OpenStack v3 API (Experimental)

The OSM OpenStack VIM connector has been updated to include experimental support for the Keystone v3 API, specifically the v3.3 version.

Keystone is the OpenStack project that is responsible for managing identity and authorization between OpenStack projects. As of the Mitaka OpenStack release (April 2016), 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.

VMware vCloud Director VIM Connector Enhancements

Release TWO adds a suite of incremental improvements to the VIM connector (aka the VIM plugin) over the version that was available with Release ONE.

- Error handling has been improved with additions such as detecting and reporting errors for malformed IP profiles within the VNFD to quickly identify instantiation failure reasons.

- Some VNF vendors choose to leverage cloud-init for configuring the VNF. The VMware vCloud Director VIM connector can now parse SSH Key Injection specified via cloud-init and delivers this configuration to the booted VM via the VMware vCloud Director post-config features where VMware tools are available to the VM.

- The OVF package generation tool has been committed to the OSM repositories and has been extended to include an option to specify the Operating System type within the OVF package (through the CLI). This OVF package tool is a key component in being able to leverage the VMware vCloud Director VIM with VNFs that have been created in alternative image package formats.

- Multi-Disk support per VDU has been added to this connector.

Packaging, Distribution, & Installation Details

Docker

The OSM SW distribution model for Release ONE was as in source code form. The one-click installation tool supported this model of distribution by downloading the code, creating several LXD (machine) containers for each component and building the SW within the containers.

Docker containers provide an alternative framework to separate the different components of the system into individual process containers. The use of the term "process" instead of "component" is intentional. Processes running within each OSM component have been identified and Docker containers are created per process and not per component.

The one-click OSM installer has been updated to support leveraging the Docker container images. This helps to reduce the installation time.

Using process containers enables an additional type of flexibility in the deployment model for OSM. It opens the possibility of leveraging Container Orchestration Engine technologies as a part of the OSM high availability, resiliency and scale-out deployment strategy. With the Docker process container approach, users will have the flexibility to deploy each OSM process in different locations, scale them independently, and design their own approach for High Availability.

OSM Installation Size Reduction

OSM release ONE requires a minimum of 8 CPUs and 16GB RAM to operate. Although this footprint is not particularly onerous for deployments, it can be a burden for new users of OSM as they may require additional time to setup an environment for first use.

For Release TWO the minimum memory requirement to operate OSM has been reduced by ~62% to 6 GB. This considerable reduction opens up possibilities for testing OSM more easily in private clouds and more cost effectively in public cloud environments.

Package Management

The OSM community has continued to develop the GUI to provide users with an intuitive graphical interface for designing and running network services. For OSM Release TWO the package management functionality in the UI Composer tool has been extended to provide a more complete set of management functionalities. Specifically the UI composer now supports the following capabilities from within the UI:

- Creation of VNF/NS Packages.
- Create, read, update, and delete (CRUD) operations on the various files within the package.
- Export of VNF/NS packages.
- The ability to maintain the visual display layout that the VNF or Network Service designer had created for subsequent import.

UI Composer Reference Lists

The NSD and VNFD have many cross-references. For example, the NSD has potentially multiple constituent VNFD ID references and the Virtual Link Descriptor has potentially multiple VNFD connection point references.

The UI Composer tool has been updated to present dropdown lists of references as valid inputs to various fields. These reference lists are dynamically updated in the UI Composer as the data models are modified. For example, if a user adds another connection point to a VNFD the new identifier for that connection point will be available in the appropriate drop down list.

This change facilitates a reduction in user error via cut-and-paste style errors. Potential challenges related to debugging instantiations created with invalid cross-references has been eradicated as only valid entries in cross-reference fields may be selected. This feature also helps to reduce the time necessary to create VNFDs and NSDs.

Usability Details

Maintenance Release Updates Without Reinstallation

OSM users that have a running Release ONE environment can upgrade to the latest maintenance release without uninstalling. The process to complete the upgrade has been documented on the public OSM wiki on the troubleshooting page [Ref 2].

VNF Console

Access to Physical Network Functions (PNFs) User Interface (UI) for items such as configuration are commonly done through telnet or ssh sessions. It is also common that when accessing the PNF in this manner the user accesses a custom console that is different to a typical Linux root Shell. This custom console exposes only specific configuration parameters applicable to the PNF.

With VNFs, two different approaches are common:

1. The access to the VNF UI is through telnet or ssh. This is similar to the typical PNFs access method described above.

2. An alternative is that the VNF UI is accessed through a terminal via VNC [Ref 3] or SPICE [Ref 4]. This is a similar model to how users of guest VMs in the cloud access the environment.

This feature enhancement focuses on the alternative approach by providing access to the VNF terminal via VNC or SPICE. For single-VM VNFs, access to the VNF console is equivalent to access to the VM console. For multi-VM VNFs there is usually a VM for management purposes. This is typically the Operations Administration and Maintenance (OAM) VM. Access to the VNF console is then equivalent to access to the OAM VM. Support is also provided for accessing the console of every VM of a multi-VM VNF. The latter feature is most useful during a debug activity.

Support for this feature in a deployment is subject to the VIM being capable of creating terminals based on VNC or SPICE. The VIM must build a URL for accessing the terminal through the hypervisor. When this is available, OSM will expose this URL up to the GUI.

cloud-init Support

In OSM, the VNF Configuration and Abstraction (VCA) module is responsible for managing VNF configuration. As with previous releases of OSM, VCA has an interface to Juju. Juju is an open source application modelling tool that the OSM community is leveraging to manage application configuration at a scale required by operators. The recommended model for specifying application configuration is via charm definitions that can be included as a portion of the VNF package.

Juju can support cloud-init in a post-boot scenario. However, there are use cases when part of the information that can be described in cloud-init scripts should be provided to the VM pre-boot. In OSM, passing pre-boot configuration data to the VIM is the responsibility of the Resource Orchestrator (RO). The intent of this functionality is to support a very limited set of configuration information that cannot reasonably be provided to the applications via the VCA path. For Release TWO, SSH key information can be leveraged from the cloud-init scripts and provided to the VMs. This is supported across the full suite of supported VIMs.

This RO functionality opens the door for VNF vendors that have defined configuration data in cloud-init to pass all of that information to the VIM for further relay to the VM, subject to the VIM supporting cloud-init. By opening this configuration path, the OSM community is acknowledging that a number of VNF vendors prefer to support configuration data in this manner. However, this is not the recommended method with OSM, and the OSM community will continue to encourage and support VNF vendors to upgrade to a managed configuration environment (described via charms) that can take full advantage of automated service delivery with OSM.

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 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. HIVE helps to ensure that OSM inter-operates successfully with multiple VIMs, SDN Controllers and NFV Infrastructure while helping to minimize barriers for community engagement.

During the Release TWO development cycle the OSM community welcomed the addition of two new remote lab environments. VMware has connected a VMware vCloud Director-based VIM environment and Wind River has connected a Wind River Titanium Server based environment. Both are now available for leveraging by the community to leverage as a part of the OSM CI/CD testing pipeline.
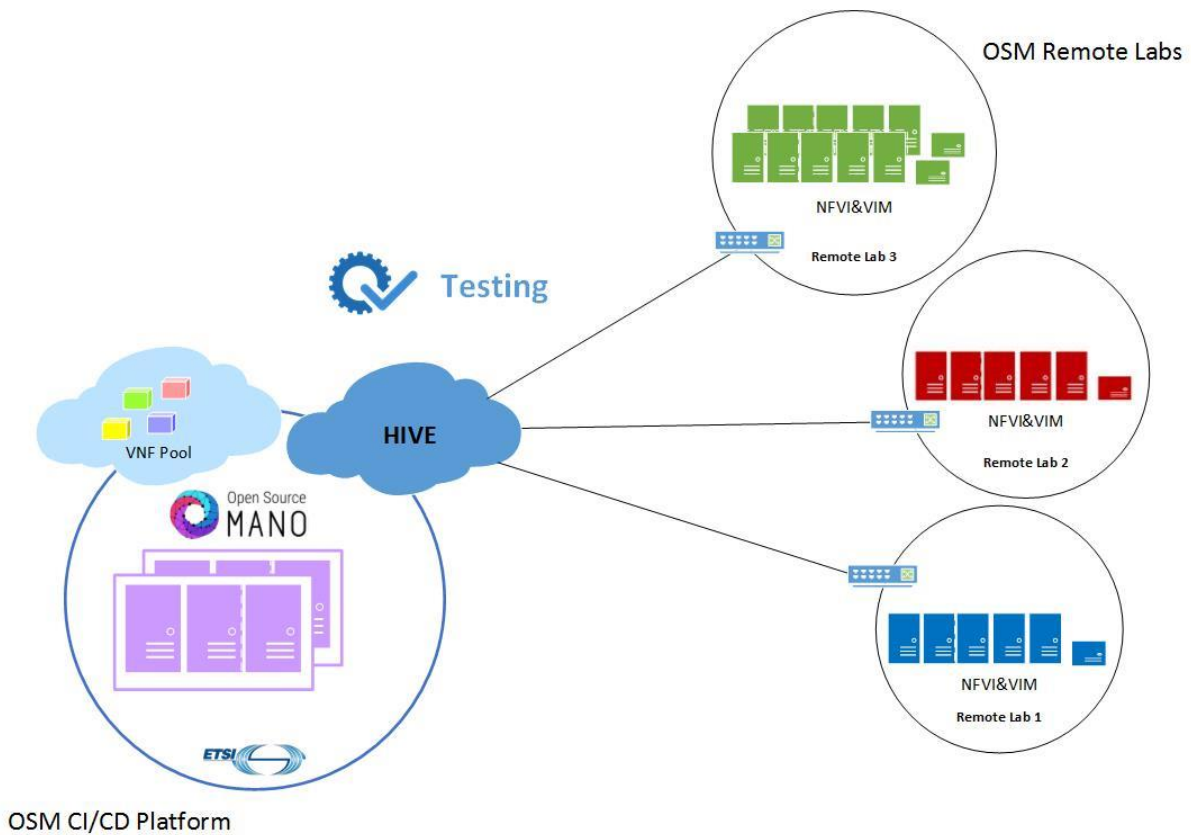
Figure 2: OSM Remote Labs Network

Common Logging & Exception Handling Extension

With OSM Release ONE the community made positive steps towards a common method for capturing detailed feedback when deploying and configuring VNFs. Error Message flow from the RO & VCA was also improved so that the messages propagated to the UI/Logs files would be more cohesive to simplify the troubleshooting of issues when using OSM.

For Release TWO, OpenVIM has upgraded to the common framework for logging and exception handling. This feature helps with the overall cohesion of the OSM project and paves the way for even more structured logging to follow.

Python Based OSM Client

OSM now provides a simplified CLI client that can be used to remotely interact with the OSM Northbound REST API. There is also a new Python functional library that can be used to programmatically interact with OSM remotely (via REST). This is the first revision of this capability and as such the focus has been on enabling the most commonly used northbound commands such as uploading VNF packages, creating/delete/view VIM accounts, and creating/deleting/viewing network services.

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.

- Lowers 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 TWO the OSM NSD and VNFD Data Models continue with broad alignment with the ETSI NFV phase 1 MANO NSD and VNFD Information Models [Ref 5].

During the development of this release, the community assessed the ETSI NFV IFA011 VNFD [Ref 6 and ETSI NFV IFA014 NSD [Ref 7] 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 the goal of 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.

Security Details

Unfiltered Interfaces

When Virtual Link Descriptors (VLDs) are created as overlay networks on an NFVI, some VIMs such as OpenStack implement a number of security policies that apply L2-L4 filtering rules to the frames/packets in those networks. By default, when creating VLDs (networks in

VIMs) and when attaching VDUs to the VLDs (attaching a VM to a network in a VIM), those security policies are created in the VIM. Depending on the VIM, the security policies can be disabled at different levels of abstraction such as a network level, a port level, or a group level.

A new feature in OSM adds the capability to enable or disable port security per VDU/VM interface. This is supported in OpenStack and OpenVIM environments. With VMware-based environments, the security settings will need to be adjusted directly via the VMware VIM interfaces.

This feature is particularly useful for VNFs that need to have access to the complete set of network traffic that could be transferred over the interface.

Data Plane Configuration Details

Underlay Network Management from RO

VIMs do not always provide management of native L2 underlay connections as may be required when deploying certain E-Line or E-LAN configurations. In the case of OpenStack, there are vendor specific solutions based on Neutron ML2 plugins for the data plane switch management related to the connectivity and configuration of SR-IOV interfaces. However, those solutions are not suitable for every type of use case such as when there is a requirement to interconnect multi-tenant VNFs (using VLAN tags per tenant) with single tenant VNFs where the VLAN tag must be automatically removed/added by the network.

This new feature in OSM Release TWO enables the Resource Orchestrator (RO) to manage underlay networks directly through an SDN Controller (SDNC), to be used when the related feature is not provided by the VIM. In such a deployment scenario, the VIM network management functionality will be used for the creation of VM instances and the creation of overlay networks, while the SDNC, driven by the RO in OSM, will be responsible for the creation and deletion of underlay networks.

After a VM is instantiated, the RO retrieves information from the VIM related to the physical allocation of the SR-IOV or full PCIe pass-through interfaces. Subsequently, the RO then directs the SDNC to connect these interfaces in E-Line or E-LAN network configurations.

OVS Support added to OpenVIM

For OSM Release ONE OpenVIM used Linux bridges to create the overlay networks used to interconnect the VMs. The Linux bridges were built on a host interface and associated with a VLAN tag which was used to transport frames between VMs located in different hosts across a switching infrastructure.

Linux Bridge-based environments work well but have some limitations. For example, VLAN tagging is the only mechanism for achieving isolation and they also lack advanced capabilities such as L3 filtering, IP tunneling, support of Openflow rules, etc.

OSM Release TWO has added Open vSwitch (OVS) [Ref 8] support for OpenVIM. OVS brings a suite of advanced networking capabilities such as those mentioned above and is better aligned with future OSM needs and use cases. Leveraging the Data Plane Development Kit

(DPDK) performance enhancement capability included with the latest OVS releases is likely to be a topic for future consideration.

Service Assurance Details

Network Service Scaling (Experimental)

This new experimental feature in OSM Release TWO supports 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.

A Network Service Descriptor can be used to articulate VNFs that need to be scaled simultaneously, termed Scaling Groups. A Scaling Group contains one or more VNFs. A scaling action is performed on the Scaling Group. The VNFs that are added 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 Northbound API. Auto-scaling support is expected to be added in a future release.

References

1. https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/1st_ETSI_NFV_Plugtests_Report_v1.0.0.pdf
2. OSM Troubleshooting Wiki: https://osm.etsi.org/wikipub/index.php/Logs_and_troubleshooting_%28Release_TWO%29#Software_upgrade
3. Virtual Network Computing (VNC)
4. Simple Protocol for Independent Computing Environments (SPICE), https://www.spice-space.org/.
5. ETSI GS NFV-MAN 001 V1.1.1 (2014-12)
6. ETSI NFV IFA011
7. ETSI NFV IFA014
8. http://www.openvswitch.org/

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