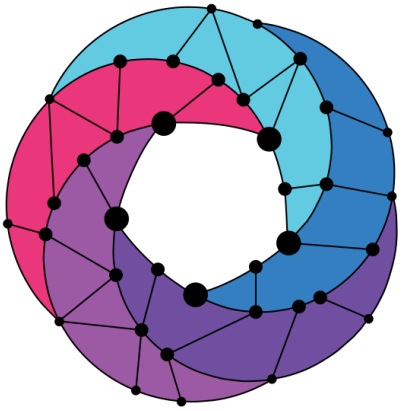


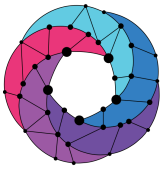
Open Source MANO

OSM Release NINE Release Notes

Open Source MANO
Technical Steering Committee
December 2020

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

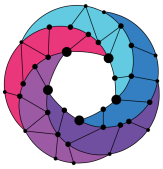
ETSI



Open Source
MANO

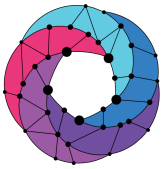
Authors

Name	Organization
Alejandro García	Whitestack
Alfonso Tierno	Telefónica
Andy Reid	BT
Antonio Marsico	BT
Atul Agarwal	Altran
Aurora Ramos	Atos
Barath Kumar R.	Tata ELXSI
David García	Canonical
Fabian Bravo	Whitestack
Francisco-Javier Ramón	Telefónica
Francisco Rodriguez	Indra
Gerardo García de Blas	Telefónica
Gianpietro Lavado	Whitestack
Isabel Lloret	Indra
José Miguel Guzmán	Whitestack
Mark Beierl	Canonical
Mark Shuttleworth	Canonical
Pål Grønsund	Telenor
Preethika P	Tata ELXSI
Ramesh Ramanathan	Tata ELXSI
Silvia Almagia	ETSI
Subhankar Pal	Altran



Content

- Introduction
- OSM and Standards Alignment
- Kubernetes, an integral part of OSM DNA. Ready for the Edge and Open RAN
- New Features in Release NINE
  - Alignment with SOL006
  - Support of Helm version 3
  - Centralized VCA for KNFs
  - Distributed Proxy charms
  - Kubernetes VCA in K8s Installation
  - RBAC external integrations
  - Multi-tenancy in Grafana
  - Data Plane Broker (DPB) WIM Connector
  - Simple strategy to allow address pairs for virtual IP
- Conclusions
- To go further



Introduction

We are happy to introduce OSM Release NINE! The 10th Release of an open source project that announced its Release ZEROin 2016 with the vision of filling the gap between the traditional way of building telecom networks, and the cloud technologies that were emerging at that time.

Nine releases and more than 600K lines of code later, OSM has evolved from that initial vision to a platform that not only fulfills the needs of Telecom Operators willing to build their horizontal virtualization environments, but also includes the most recent technologies (such as Kubernetes) with different options for deploying (such as Juju and Helm), and bundling a suite of other components that complete the offering of the most advanced open source NFV orchestrator in the market.

In this release NINE, one of the main goals was to complete the alignment with the ETSI NFV information model, NFV-SOL006, allowing OSM to consume standard network functions and services descriptors, and bringing the benefits of standardization to an expanding ecosystem.

In addition to that, a number of very important features were contributed by leading organizations and enthusiastic individual contributors who are convinced that the Telecommunications industry should not further delay the adoption of these new technologies.

So, thanks to our great community that made this new release possible!

OSM and Standards Alignment

One of the main objectives of the OSM project, in addition to providing a concrete MANO stack implementation, is to ensure a large and diverse NFV ecosystem. A lot of our initiatives are based on using and reusing existing successful technologies (such as Kubernetes, Juju, Helm, etc), facilitating the mechanisms of VNF onboarding (such as our recently launched VNF Catalogs), and adopting industry standards.

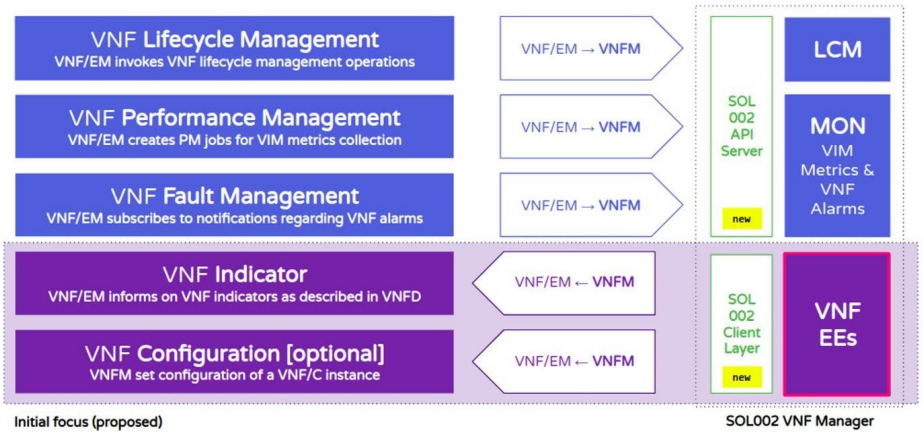
With Release NINE we are announcing native support of NFV-SOL006 as OSM Data Model, following the alignment with ETSI NFV standards in previous releases.

NFV standardization comes mainly from ETSI ISG NFV, founded in November 2012 by some of the world's leading telecom network operators, making ETSI the home of Network Functions Virtualisation (NFV).

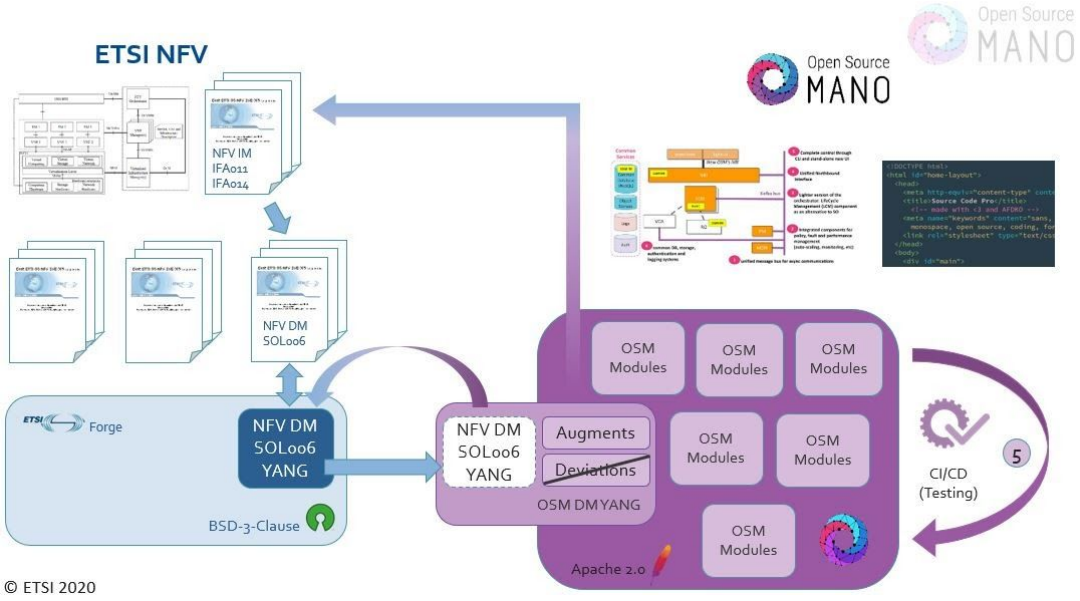
OSM, along with other projects and commercial solutions, has been progressively adopting those NFV standards as they were published:

- NFV-SOL005, that allows OSS/BSS to interact with the NFV Orchestrator (NFVO) by leveraging open protocols (RESTful APIs), is supported by OSM Northbound API since Release FOUR.

  NFV-SOL002 defined capabilities are supported since Release SIX, with focus on VNF Indicator collection and VNF Configuration. VNF Configuration Interface is implemented through VCA, while for Indicator collection, the system offers an Execution Environment Framework that supports SNMP Collection since Release EIGHT, and can be easily expanded to cover SOL002 indicator collection calls.

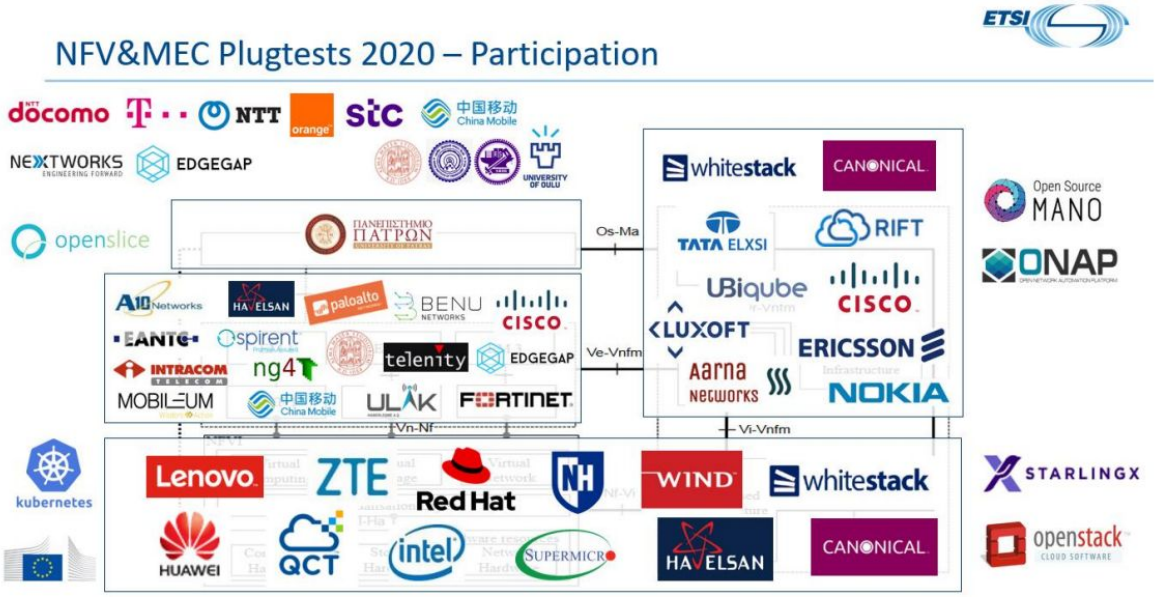


- NFV-SOL006 has been adopted natively in this Release NINE, allowing OSM to on-board and orchestrate VNFs and NS described by using ETSI-SOL006 descriptors, and replacing the previous OSM specific VNFDs and NSDs. When needed, OSM has augmented the SOL006 model to ensure no functionality was lost during the migration. These augments are being documented and sent back to NFV ISG for their potential inclusion in future releases of the specifications. No deviation to the NFV model was required, making the transition from old OSM descriptors to new NFV-SOL006 descriptors straight forward.



Standards alignment is at the heart of the OSM project vision, as the means to nurture and develop a healthy and diverse ecosystem of interoperable components.

In that regard, OSM has been actively involved in the ETSI NFV Plugtests Programme activities since its launch in 2017, a series of events and activities organised by ETSI Centre for Testing and Interoperability aiming to validate the interoperability of NFV & MEC standards implementations in real integration conditions. The latest ETSI NFV&MEC Plugtests Report 2020, reports on a growing number of vendors, including several OSM distributions, validating multi-vendor Interoperability and Conformance to NFV API Specifications.



NFV API Conformance testing is automated with a Robot Framework Test Suite delivered by ETSI NFV together with the NFV API Conformance Testing Specification (ETSI GS NFV-TST 010). This test suite offers solution providers a powerful tool to self-evaluate their conformance with NFV APIs Specifications and identify potential gaps in their implementations. If you want to learn more about API Conformance testing for NVF, take a look at this recent post: Insights on API Conformance testing for NFV-MANO by Pierre Lynch, NFV TST WG Chairman and Rapporteur of TST010.

OSM has started integrating these TST010 Robot Framework Test Suites in its CI/CD to ensure that new features and code changes do not compromise API Conformance.

Kubernetes, an integral part of OSM DNA. Ready for the Edge and Open RAN

Kubernetes is becoming a de-facto standard for the deployment of containerized applications with built-in scalability, high availability, fault tolerance and disaster recovery in the IT world. Network is not an exception to this IT revolution, and some of the Edge and Open RAN use cases are already incorporating Kubernetes as the platform to deploy and operate the so-called container-based network functions (CNF).

The OSM project follows closely the growing adoption of Kubernetes in the network industry, specifically in ETSI MEC and the O-RAN Alliance, and the evolution of the Kubernetes ecosystem in the IT world, so that OSM can easily support all the new use cases that are appearing. In this regard, one year ago, Release SEVEN introduced the support of Kubernetes-based CNF, and now with Release NINE we are continuing that work by incorporating a set of features, mainly:

- The support of Helm version 3, the new version of the well-known package format for Kubernetes applications, guaranteeing the support of thousands of existing applications and the future ones that will come in the new format.
- The capability to operate distributed applications in multiple Edge locations through the support of distributed proxy charms.
- The default installation of OSM on top of Kubernetes, making it the default choice both in the community installer and the charmed installer.
- The deployment of the VCA (Juju) in the same Kubernetes cluster as the rest of the OSM components.

Kubernetes is already an integral part of OSM's DNA, and next releases will keep incorporating new features as required to enable the different use cases where OSM can help in the deployment and operation of CNFs.

New Features in Release NINE

Release NINE brings a set of new features:

- Alignment with SOL006
- Support of Helm version 3
- Centralized VCA for KNFs
- Distributed Proxy charms
- Kubernetes VCA in K8s Installation
- RBAC external integrations
- Multi-tenancy in Grafana
- Data Plane Broker (DPB) WIM Connector
- Simple strategy to allow address pairs for virtual IP

The following sections describe in further detail these major new features coming with Release NINE.

Alignment with SOL006

```
vnfd:
  - id: vEPC_vnfd
    product-name: vEPC_vnfd
    description: Generated by OSM package generator
    provider: OSM_VNFONB_TF
    version: "1.0"
    mgmt-cp: spgwmme-mgmt-int

    virtual-compute-desc:
      - id: spgwmme-compute
        virtual-cpu:
          num-virtual-cpu: 2
        virtual-memory:
          size: 4
      - id: hss-compute
        virtual-cpu:
          num-virtual-cpu: 1
        virtual-memory:
          size: 2

    virtual-storage-desc:
      - id: spgwmme-storage
        size-of-storage: 10
      - id: hss-storage
        size-of-storage: 10

    sw-image-desc:
      - id: nextepc-spgwmme-base
        name: nextepc-spgwmme-base
        checksum:
          hash:
```

The adoption of ETSI NFV-SOL006 data model allows OSM to onboard and orchestrate VNFs and NS described with NFV-SOL006 descriptors, and replacing the previous VNF and NS descriptors, which used a very rich, but OSM-specific data model.

In particular, OSM not only adopts the complete NFV-SOL006 specification, but also extends it (through standard YANG augment mechanisms) to allow the modelling of features that are not yet addressed by NFV standards.

OSM Release NINE is also including a translator tool, so that former OSM-specific descriptors can be easily migrated to NFV-SOL006, thus making the transition as smooth as possible.

By adopting NFV-SOL006, OSM expands its VNF Ecosystem and ensures a standard based development and maintenance of VNF Packages.

Support of Helm version 3

OSM supports the deployment of Kubernetes workloads since Release SEVEN. In that release, OSM incorporated the Kubernetes workloads using the KDU concept. In OSM, a Network Function consists of a collection of Deployment Units (DU) that can be either Virtual (VDU), Physical (PDU), Kubernetes-based (KDU), or even a combination of them, leading to any form of Hybrid Network Function.

KDUs can be modeled in OSM through helm-charts and juju-bundles, both of them well-known package formats for Kubernetes applications. For helm-charts, Helm version 2 was used up to Release EIGHT, but this version reached end of life in November 2020.

OSM Release NINE includes the support of Helm version 3, which is the current version of Helm and the default helm-chart version now in OSM. Being aware that both versions will

need to co-exist for some time with existing commercial CNFs, Release NINE keeps backwards compatibility with former Helm v2 charts, so it is still possible to deploy them.

Supporting Helm v3 allows OSM to deploy thousands of new Kubernetes applications that are already publicly available in Helm chart repositories on the Internet and use the latest packaging choice.

Centralized VCA for KNFs

The VCA is the core component of OSM that deploys what we call operators, that can implement what in OSM are called *primitives*. These operators are implemented by Juju *charms*, and consist of a set of custom scripts that encapsulate the knowledge that a human operator of a workload would have in order to operate it (i.e. install, upgrade, do a backup, restore from a backup, integrate it with another application, etc.)

Since Release SEVEN, OSM can deploy Kubernetes workloads using charms, providing a unified experience when writing operators for Physical, Virtual, and Containerized Network Functions.

In this release, the experience of deploying Kubernetes workloads with charms is fully integrated to the VCA. The Juju controller, that is the Operator Lifecycle Manager for charm universal operators, is able to handle thousands of kubernetes clusters without having to deploy any pod as a prerequisite to the cluster.

Distributed Proxy charms

OSM can orchestrate different Network Functions (NF) that can be operated using charms (operators), which are operational packages (code) that allow the addition of day-1 and -2 primitives. There are two types of charms in OSM: native and proxy charms. Charms that handle the full lifecycle of a workload are called native charms, and these manage the application from the software installation, to upgrade and integration. In native charms, the operations code lives and is executed from the workload itself. On the other hand, when the workloads are fixed (i.e. PNFs), and no software related to the operator can be installed, the charm needs to live somewhere else - these are called proxy charms.

All charms are deployed by the VCA, which has the concept of *clouds*. Clouds are substrates in which the VCA can deploy charms to. By default in OSM, the VCA knows about two clouds for proxy charms: Kubernetes and LXD.

Up to Release EIGHT, proxy charms of a specific type (LXD or K8s), could only be deployed to one cloud.. This new feature in Release NINE allows the association of different clouds with different VIM accounts for proxy charm deployments. This is an important step towards production use cases, as it removes the constraint of having all the proxy charms in the same cloud, and can effectively distribute them across VIMs. Additionally, it allows the placement of the proxy charms closer to the workloads, decreasing delays and increasing the overall performance and user experience.

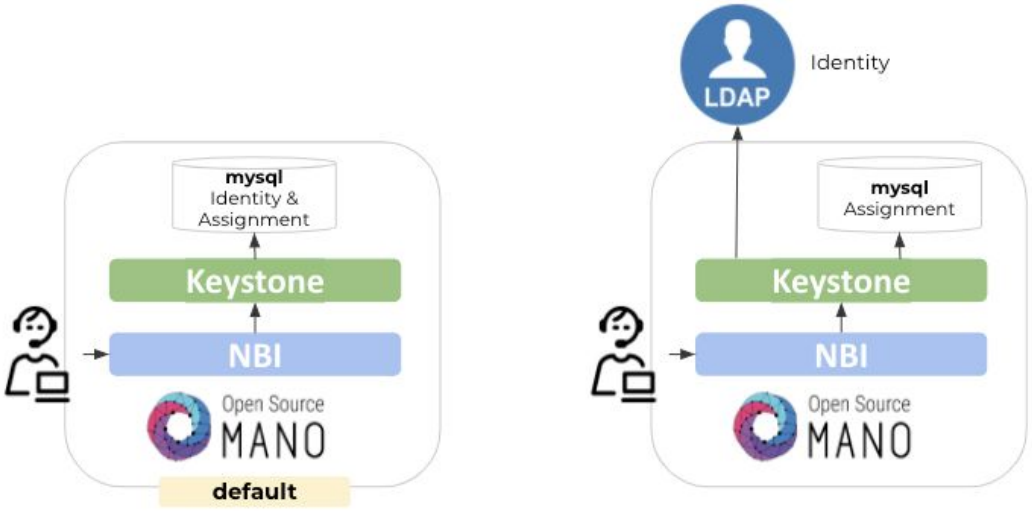
Kubernetes VCA in K8s Installation

From Release ONE, Juju has been the VCA of OSM, enabling operations to a diverse set of Network Functions (NF). Juju is an Operator Lifecycle Manager (OLM) that can deploy operators to different substrates such as Kubernetes, LXD, and Openstack.

Up to Release EIGHT, the VCA has been deployed by default to an LXD cloud, but from Release NINE, it is bootstrapped to Kubernetes (in the kubernetes-based installations - Charmed OSM and k8s). The main advantage of this feature is that the VCA comes up faster and can benefit from Kubernetes resiliency.

RBAC external integrations

Role-Based Access Control (RBAC) external integration is now available by default through Keystone, which provides out-of-the-box integrations (i.e. LDAP) and more advanced use RBAC cases.



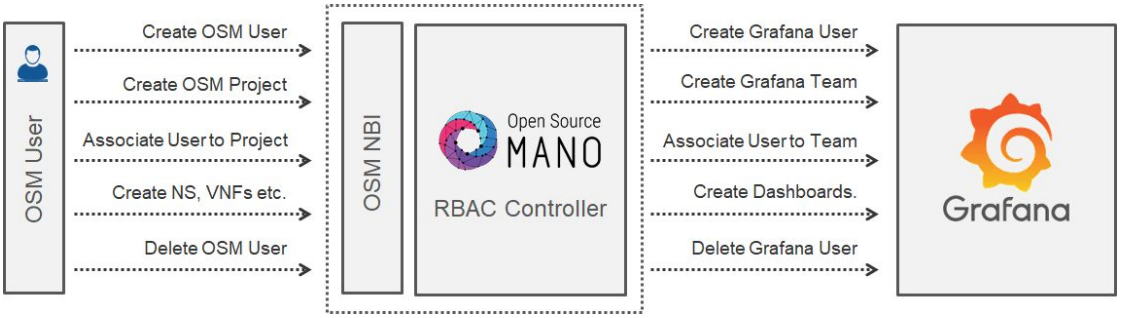
Multi-tenancy in Grafana

Role-Based Access Control (RBAC) in OSM provides different users and projects a controlled access to different resources. That means that, in order to obtain privileges to do something in OSM, you should have a user, belonging to a project, with a specific role or roles over the project.

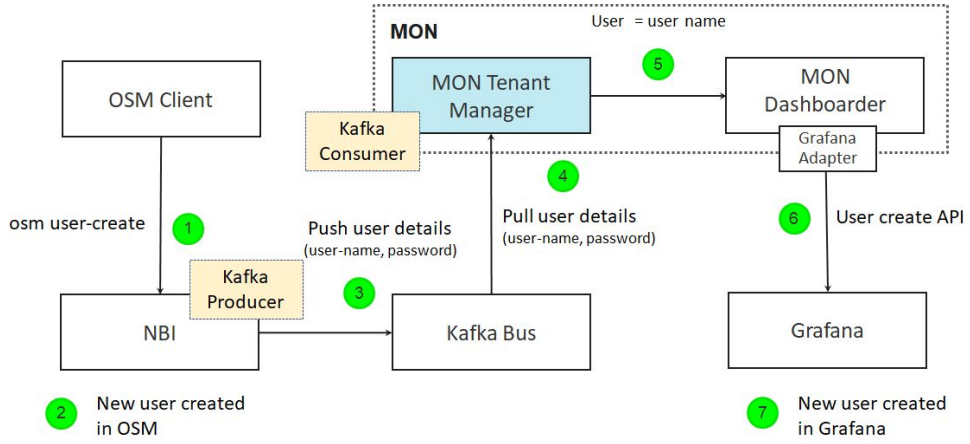
Grafana multi-tenancy extends the RBAC feature to OSM's Grafana and provides OSM users with controlled access to OSM dashboards. With multi-tenancy, users can login to Grafana with their OSM credentials instead of a common username as was the case in previous releases.

The user now only has access to dashboards of the resources (e.g. VNF and NS), they also have access rights for OSM. Also, users do not have access to other users' dashboards.

For every OSM user and project, the corresponding user and team are created in Grafana, as shown in figure below. When an OSM user is associated with an OSM project, the corresponding Grafana user is associated with the corresponding Grafana team.

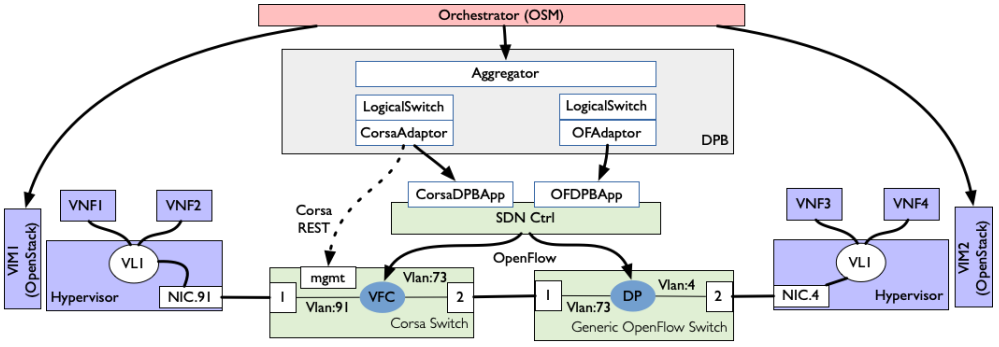


The figure below shows the example of the user creation scenario and describes how different OSM modules interact to realize this use case:



Data Plane Broker (DPB) WIM Connector

The DataPlane Broker (https://dataplanebroker.github.io/) is an open and extensible service providing WIM management capabilities over SDN networks. The framework uses a hierarchical architecture that allows scalable and hierarchical network management.



OSM Release NINE includes a new WIM Connector for integrating with DPB, as one of the options for the OSM WIM (WAN Infrastructure Manager).

Simple strategy to allow address pairs for virtual IP

Openstack VIMs allow two ways to access virtual IPs inside VMs:

1. Per-IP/MAC or prefix, with allowed-address-pairs

2. Full access, by disabling port-security

In previous releases, OSM only supported the second option which might not be enabled in some VIMs. With this new feature in Release NINE, OSM models the use of “allowed-address-pairs” instead of just the “port-security” strategy, in order to enable access to virtual IPs inside VMs, thus now supporting both options

Conclusions

Release NINE brings a number of new features required to adapt to different deployment environments and is also the first open source orchestrator to adopt natively the ETSI NFV-SOL006 standard. SOL006 allows vendors to describe their VNFs by using a standardized model, creating an abstraction layer that, in addition to supporting any type of VIM, ensures interoperability across orchestrators.

Also, this release coincides with the announcement of new commercial deployments, which is a consequence of OSM not only producing the code, but also building a healthy and diverse ecosystem of distributions, integrators, consulting and training companies, that are working together with PNF, VNF, and CNF vendors, OSS/BSS solution providers and research entities to consolidate OSM as the orchestrator of choice for operators deploying virtualized networks in production.

ETSI OSM is an open community in its fifth year of growth, that welcomes the contribution of people and organisations sharing the vision of open source orchestration. If you want to know how to join us, please visit How To Get Involved In OSM

To go further

For more information, please see:

- Open Source MANO's documentation
- Open Source MANO's developer's guide
- Open Source MANO's VNF Onboarding guide
- OSM Ecosystem
- OSM Members and Participants