OSM Release ELEVEN Webinar

Francisco-Javier Ramón (Telefónica, ETSI OSM Chair)
Gerardo García (Telefónica, TSC Chair)
Ramesh Ramanathan (Tata Elxsi, TSC Member)
Mark Beierl (Canonical, TSC Member)
What does OSM do...

... and how?
OSM provides a platform to create **Networks as a Service** and to manage them conveniently later.
... on different types of infrastructure and across different locations...

... with VNFs composed of VMs, containers and/or physical elements...

a) All VMs

VM

VM

b) All Containers

Container Network Function (CNF)

KBs

KBs

c) All Physical

Phy

Phy

Phy

d) Hybrid cases

Hybrid Network Function (HNF)

KBs

KBs

KBs
... and ready for network-specific workloads whenever needed

Huge Pages

NUMA Topology Awareness

CPU Pinning

Data Plane assignment

Line rate with all frame sizes

x100

x100
All in OSM is model-driven to make VNFs and scenarios as portable and reusable as possible.

**NS PACKAGES / SLICE PACKAGES:**

Upon instantiation, you just need to decide:
- The target VIM (or VIMs)
- Values for the parameters (IP addresses, keys, etc.)

**DEPLOYED INSTANCES:**

- Instance #1 based on NS “A”
- Instance #2 based on NS “A”
- Instance #3 based on NS “A”
All these OSM packages are oriented to maximize reusability for multiple scenarios.

Can be easily customized upon instantiation

Parametrized

Can be easily customized upon instantiation

OSM Package

Parametrized

Can be easily customized upon instantiation

Model agnostic to infrastructure

VNF vendor does not need to know the details of our infrastructure upfront

Multi-VIM Multi-SDN

Can be easily customized upon instantiation

Models include full lifecycle

Recurrent operations are greatly simplified

Day-2 can be run from OSM
Open Source MANO

A vibrant and thriving community
OSM community is really **LARGE AND DIVERSE**, with 150 members today, but always **OPEN** to new participants.

- 15 Global Service Providers
- Leading IT/Cloud players
- VNF providers

(*) Names & brands may be claimed as the property of others
OSM Ecosystem

Companies listing their products and offers related to OSM (like “OSM Yellow pages”)

• Searchable by potential customers looking for OSM-related products
• Only with demonstrable OSM-related products/offers
• Opt-in process, continuously open

https://osm.etsi.org/wikipub/index.php/OSM_Ecosystem
... and the new release!

Release ELEVEN
Release ELEVEN brings new features to foster current and new deployments

**SOL004 and SOL007 package formats**

**Brand-new support for Google Cloud**
- Completing the infrastructure support for 3 largest public clouds

**Fine-grained operations in CNFs**
- Start and stop services
- Run one-shot commands
- Files API

**Better coordination across PNFs, VNFs, and CNFs**
- Enhanced data exchange between NFs in the NS.

**CNF monitoring from Kubernetes metrics**
- Metrics collection from K8s clusters in centralized Grafana dashboard.

**Enhanced installation process**
- Support of Ubuntu 20.04 and better tracking of the installation process.

... and other improvements in usability and stability derived from the learnings of latest OSM production deployments

Available at: osm.etsi.org
... which are added on top of an already long set of features...
At this point, it is becoming easier explaining OSM features in practice

**MAGMA EPC DEMO (2020)**

11 teams onboarding 8 NFs in just one week!

**OSM#11 Hackfest**

Asterisk

- freeRADIUS
- PowerDNS
- OPC UA
- NetNumber

**Multi-Cloud Deployments**

**OSM-MR#11 Hackfest**

Release TEN Webinar
Edge orchestration with OSM

https://osm.etsi.org/gitlab/vnf-onboarding/osm-packages/tree/master/magma

© ETSI
Frictionless Multi-Cloud Deployments

Gerardo García de Blas (Telefónica, OSM TSC Chair)
Taking as starting point a service onboarded for production...

DEPLOYMENTS:
- 5G NF #1 (x2 sites)
- 5G NF #2 (1 site)
.. we can easily re-deploy it in 5 different types of clouds (and several sites)

**DEPLOYMENTS:**
- 5G NF #1 (x2 sites)
- 5G NF #2 (1 site)

**PRIVATE TELCO CLOUD**

(PRODUCTION)
.. we can easily re-deploy it in 5 different types of clouds (and several sites)
Oracle’s Cloud Native Core Policy

- Convergent solution for 5G Policy Control Function (PCF) and 4G Policy and Charging Rules Function (PCRF)
- Containerized Network Function (CNF) with different tiers, all of them fully based on containers:
  - Core Policy
  - DB tier
  - Monitoring tier
The deployment was prepared, for convenience, as a set of NF and NS.

DEPLOYMENTS:
- 5G NF #1 (x2 sites)
- 5G NF #2 (1 site)
This brings obvious operational benefits out of the box...

- Huge cost savings in testing and validation.
- Time to market for second deployments is minimal.
- Minimization of errors.
- No lock-in with specific clouds.
- Package sharing between OBs becomes possible, regardless their underlying infrastructure.

... while enables advanced deployment scenarios

- Ability to move workloads between clouds.
- Easier growth in capacity with different clouds.
- Advanced HA schemes.
OSM and ETSI Architecture

Ramesh Ramanathan (Tata Elxsi, TSC Member)
OSM, Standardization and Open Platforms

Virtualized Infrastructure Managers

- OpenStack
- Azure
- GCP
- AWS
- VMWare
- K8S Clusters

Plugins
- LDAP
- DNS
- DHCP
- NTP

Repository
- NSD
- VNFD

TCP/IP

Os-Ma

OSS / NMS

REST API

Big Data

SDNC

WIM

© ETSI
Coordination Between CNFs, VNFs, PNFs

Mark Beierl (Canonical, TSC Member)
Simple Network Service
Firewall must open ports and forward traffic to Load Balancer

Load Balancer needs IP addresses of backend CNFs

CNF units can be scaled individually
Data Exchange via Relationship

Establish relationship
Exchange data

Requires:
- Information on ports to open
- Target IP address for forwarding traffic

Provides:
- Its IP address and port
- Requested firewall port to open
Defining a Relationship

relation:
  - name: firewall-loadbalancer
    provider:
      endpoint: firewall-rule
      vdu-profile-id: load-balancer
    requirer:
      endpoint: firewall-rule
      vdu-profile-id: firewall
Relation Events

- **Relation Joined**
  - Notification that a new relation is going to be formed

- **Relation Changed**
  - Primary means of data exchange
  - Used to send and receive data between deployment units

- **Relation Departed**
  - Notification that relation is being removed due to deployment unit being terminated

- **Relation Broken**
  - Sent after all units have been terminated as the final message
Create Network Service With Relationship

Execution Environment

Relation Joined

Execution Environment
Data Exchange via Relationship

Execution Environment

Open port with target data from event

Relation Changed

Execution Environment

Port to open
My IP Address
My IP Port
Removing the Network Service

Execution Environment

On Stop Event

Execution Environment

Port to close
My IP Address
My IP Port
Removing the Network Service

Close port with target data from event

Execution Environment

Relation Changed

Execution Environment
Removing the Network Service

Execution Environment → Relation Broken → Execution Environment
def on_firewall_loadbalancer_relation_changed(self, event):
    event.relation.data[self.model.unit]["port-to-open"] = "443"
    event.relation.data[self.model.unit]["target-address"] = self.ip_address
    event.relation.data[self.model.unit]["target-port"] = self.listen_port

def on_firewall_loadbalancer_relation_changed(self, event):
    port=event.relation.data[self.model.unit]["port-to-open"]
    target_address=event.relation.data[self.model.unit]["target-address"]
    target_port=event.relation.data[self.model.unit]["target-port"]
    add_firewall_rule(port, target_address, target_port)
Further Information

https://juju.is/docs/sdk/relations