OSM architecture
Gerardo García de Blas (TSC Chair, Telefónica)
OSM architecture - Whiteboard
OSM architecture

- **Unified Northbound Interface (SOL005-based), decoupled from LCM**
- **E2E orchestration through Life Cycle Manager (LCM)**
- **Message bus for async communications**
- **Common DB, auth and object storage**
- **Integrated components for policy, fault, performance management and placement**
- **Complete control through CLI and UI**
- **Execution environments for VNF configuration**
Identity & Assignment Operations

When dealing with the creation, modification or deletion of users, projects and roles, the interacting components vary according to the selected backend.
Uploading packages

When reading, uploading, modifying and deleting a Network Slice Template, Network Service Package or VNF Package, the following components interact.

CLI Example: `osm nfpkg-create myvnfpackage.tar.gz`
Adding VIM/SDNC

When registering new VIMs or SDN Controllers, the following components interact.

CLI Example: `osm vim-create --name myVIM --user myuser --password myprecious --auth_url http://172.21.7.5:5000/v3 --tenant mytenant --account_type openstack`
Adding a K8s Cluster

When registering new Kubernetes clusters, the following components interact.

CLI Example: `osm k8scluster-add --creds myCredentials.yaml --version '1.15' --vim myVIM --description "My K8s cluster" --k8s-nets '{"net1": "myVIMnet"}' myK8Cluster`
When launching a new instance of a Network Service or Slice Instance (n x VNFs), the following components interact.

CLI Example: osm ns-create --ns_name myNS --nsd_name myNSD --vim_account myVIM
When launching a Day-2 primitive over a NS, the following components interact.

CLI Example: `osm ns-action myNS --vnf_name 1 --action_name myAction`
Monitoring framework
What MON Collector Monitors?

- VIM Metrics (Openstack & VMware)
- VNF/VDU Infrastructure Metrics (Openstack & VMware)
- VNF Metrics/Indicators (VCA)
- VNF Specific Metrics (using Prometheus Exporters)
- SDNC Metrics
- K8s Monitoring (Introduced in Release 11)
For infrastructure metrics to be collected, your VIM should support a telemetry system.

From Rel 7.0 onwards, metric collection works with:

- OpenStack **Ceilometer** and **Gnocchi** based telemetry services.
- VMware vCD (**vCloud Director**) with **vROps** (**vRealize Operations Manager**)

Openstack metrics collected are -

- CPU utilization, memory utilization, disk read and write, packets sent and received, packets dropped etc.
MON collects VIM or VNF/VDU metrics (defined in VNFD) using telemetry client gnocchi

MON collects VIM or VNF/VDU metrics (defined in VNFD) using telemetry client gnocchi

Prometheus TSDB stores metrics exposed by MON

Provides analytics dashboards that shows metrics value from TSDB

Grafana

Dashboards

TSDB

Prometheus

MON

Prometheus Exporter

mon-collector
• Few VNF metrics can also be collected directly from VNFs using VCA (VNF Configuration & Abstraction), through the Juju Metrics framework. Granularity is fixed to 5 minutes.

• Few example metrics that can be collected are:
  • Number of users
  • Load average

• Not a preferred approach for large scale metric collection as Juju is not designed for telemetry.
VNF specific metrics collection is done by using **Prometheus Exporters** running as “**execution environments**”, which translate into PODs instantiated in the same K8s cluster where OSM runs. These PODs follow the VNF lifecycle and are dedicated to the collection of metrics. From Rel 8 OSM support has support for SNMP Exporters, to grab scalar provided by any SNMP MIB/OID.
K8s Monitoring

1. NBI → RO
   - CNF Details
   - K8S Cluster Address

2. RO → CNF
   - Launch NS/CNF

3. MON → OSM
   - Create Dashboard

4. MON → Grafana
   - Get container metric value (for alarm evaluation)

5. Grafana → Prometheus Operator
   - Get container metric value (for alarm evaluation)
   - Collect metrics from pods

6. OSM → Grafana
   - MON Details
   - K8S Cluster Address

Release 11

Future Release

© ETSI
Troubleshooting
A general approach for OSM Troubleshooting is to first look for error messages in “show” commands, as in:

```
osm ns-show [ns]
osm vim-show [vim]
```

Besides that, knowing which components interact for each operation, you can troubleshoot by looking at the logs of each component. All troubleshooting tips are being documented in the user guide, here: https://osm.etsi.org/docs/user-guide/09-troubleshooting.html
Open Source MANO

For more information:

osm.etsi.org
osm.etsi.org/wikipub
osm.etsi.org/docs/user-guide