Open Source MANO

OSM Hackfest – Session 8
OSM Service Assurance

Gianpietro Lavado
(Whitestack)

ETSI

Open Source MANO

Current Architecture & Features

OSM Service Assurance

Service Assurance MDG

Open Source MANO

Main components

MON

- Covers the basic uses cases, with a solid architecture to expand them easily.
- Opportunities to enhance usability.

POL

- Designed around the autoscaling use case.
- Starting to cover VNF alarms.
- Architecture needs a revisit based on expected use cases.

Prometheus

- OSM's TSDB for metrics since REL5
- Opportunities to enhance multi-tenancy to match new RBAC capabilities.

Auxiliary/Optional Tools

Grafana

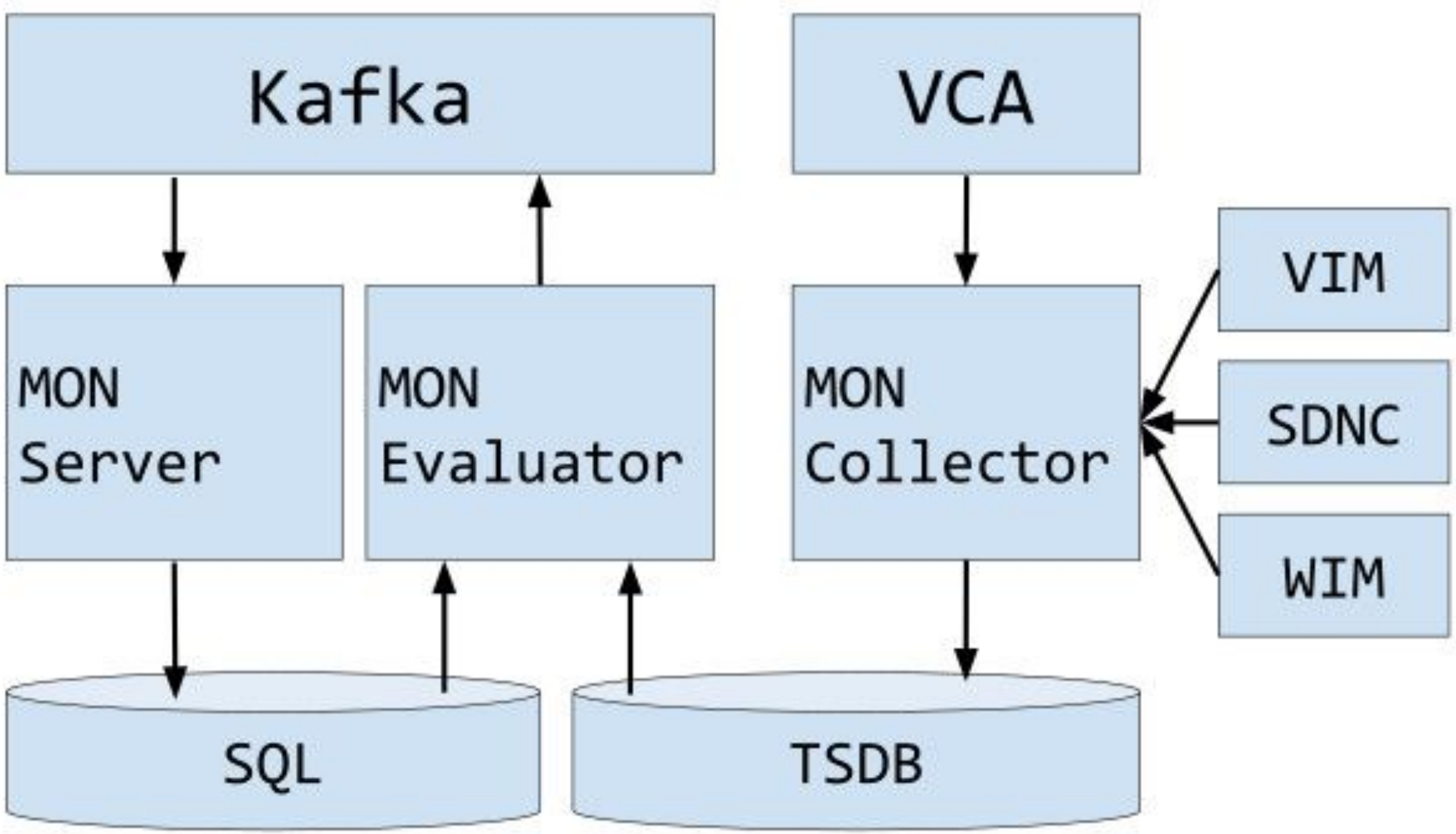
- Integrates seamlessly with Prometheus.
- Great tool for enhancing usability of the system's Service Assurance → should be included by default?

ELK

- Proved seamless integration with OSM.
- Main use case remains at log processing where stack is used.

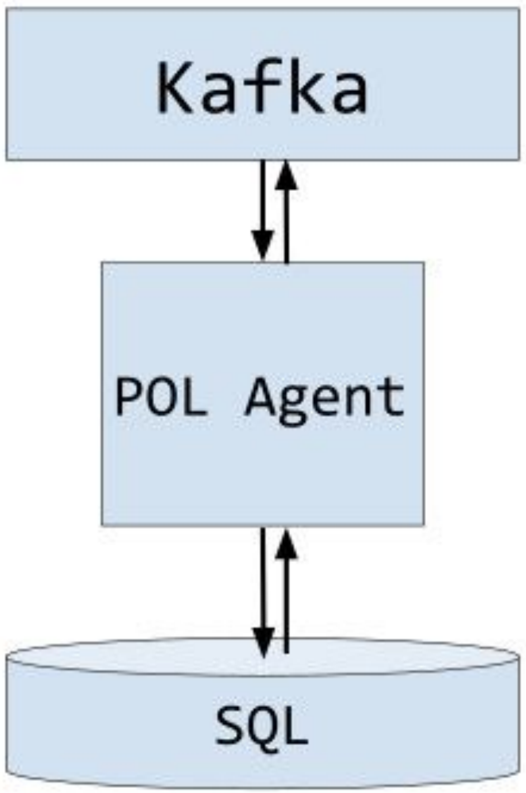
MON Architecture

Formal documentation: https://osm.etsi.org/gitlab/osm-architecture/osm-arch-doc/blob/master/04-mon.md



POL Architecture

Formal documentation: https://osm.etsi.org/gitlab/osm-architecture/osm-arch-doc/blob/master/05-pol.md



Metrics collection

Open Source MANO

Prometheus collects the following metrics from “MON Exporter”

Metrics Collection @ OSM				
Metric	**Collection type**	**Behavior**	**KPI**	**Labels**
VIM Status	Infrastructure	By default	status (up/down)	vim_id
SDNC Status			status (up/down)	sdnc_id
VM Status	VNF		status (up/down)	nsr_id, vnf_member_index, vdu_name
VDU CPU Utilization		Enabled by descriptor	utilization, rate, etc.	
VDU Memory Utilization				
VDU Packet forwarding				
VNF Metrics through Juju *(to be deprecated)*				

Metrics collection

- VDU Metric Collection from VIM

```
vdu:
  id: hackfest_basic-VM
  ...
  monitoring-param:
        - id: "cpu_util"
          nfvi-metric: "cpu_utilization"
...
monitoring-param:
-   id: "vnf_cpu_util"
    name: "vnf_cpu_util"
    aggregation-type: AVERAGE
    vdu-monitoring-param:
      vdu-ref: "hackfest_basic-VM"
      vdu-monitoring-param-ref: "cpu_util"
```

nfvi-metric corresponds to a OSM metric name which maps to the corresponding metric in each supported VIM

Autoscaling

- Scaling descriptors can be included and be tied to automatic reaction to VIM/VNF metric thresholds. An internal alarm manager is supported, so that both VIM and VNF metrics can trigger threshold-violation alarms and scaling actions.

```
scaling-group-descriptor:
- name: "vdu_autoscale"
  min-instance-count: 0
  max-instance-count: 10
  scaling-policy:
  - name: "cpu_util_above_threshold"
    scaling-type: "automatic"
    threshold-time: 10
    cooldown-time: 120
    scaling-criteria:
    - name: "cpu_util_above_threshold"
      scale-in-threshold: 10
      scale-in-relational-operation: "LT"
      scale-out-threshold: 60
      scale-out-relational-operation: "GT"
      vnf-monitoring-param-ref: "vnf_cpu_util"
  vdu:
  - vdu-id-ref: hackfest_basic_metrics-VM
    count: 1
```

VNF Alarms (new)

- Alarms based on metric thresholds can be sent to webhooks

```
vdu:
-   alarm:
      -   alarm-id: alarm-1
          operation: LT
          value: 20
          actions:
            alarm:
              - url: https://webhook.site/1111
            ok:
              - url: https://webhook.site/2222
            insufficient-data:
              - url: https://webhook.site/3333
          vnf-monitoring-param-ref: vnf_cpu_util
```

Open Source MANO

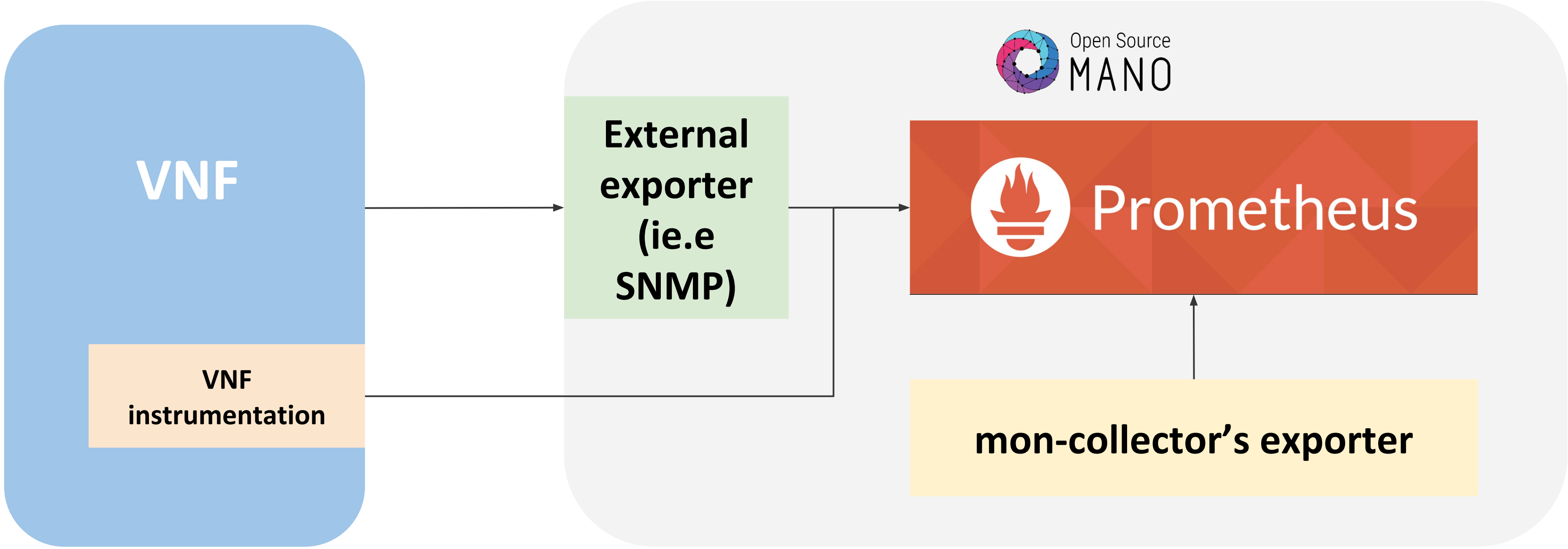
New Proposals

OSM Service Assurance

New methods for VNF Indicator Collection

Objective: Evolve the way OSM collects VNF indicators to allow for more compatibility with VNFs, real-time collection and standards alignment.

A first approach is using additional “Prometheus exporters”



OSM System Metrics

Open Source MANO

Objective: OSM Operators can install OSM and immediately and permanently know the health of the system.

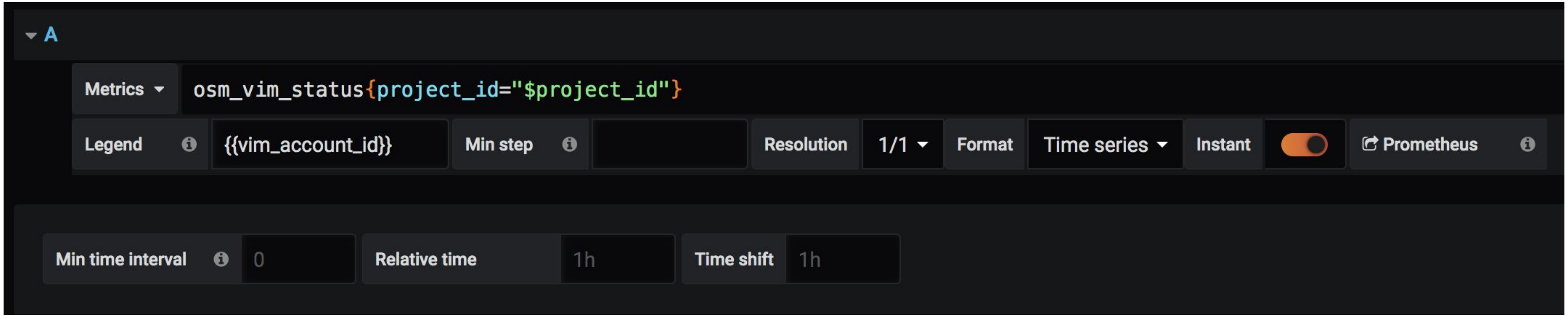
	Feature 7898	Feature 8132
Coverage	OSM on Kubernetes	OSM on Docker Swarm
Additional components	● Prometheus Operator Chart (New prometheus instance, Grafana and different exporters: node, cadvisor, etc.) ● Other charts: MongoDB, MySQL and Kafka exporters	● Grafana promoted to OSM stack. ● Node exporter ● CAdvisor exporter
Implements	Multiple Grafana dashboards for a comprehensive health check of the system.	Single Grafana dashboard with the most important system metrics.

Check it out! (beta) → http://172.21.248.14:3000/d/gDHmpHbWk/osm-system-metrics

Project-scoped VIM/VNF Metrics

Objective: Follow RBAC structure for metric consumption.

- Prometheus does not support multi-tenancy, other projects need to be explored (e.g. Cortex)
- **Short-term proposal is to add a label for project_id in all Prometheus metrics**



A

Metrics: `osm_vim_status{project_id="$project_id"}`

Legend: {{vim_account_id}}

Min step:

Resolution: 1/1

Format: Time series

Instant

Prometheus

Min time interval: 0

Relative time: 1h

Time shift: 1h

Find your dashboard! (beta) → http://172.21.248.14:3000/dashboards

Grafana Dashboard Automation

Objective: adding to the previous feature, a new “MON Dashboarder” component will take care of dashboard “lifecycle”.

Updates in...	...automates these dashboards...	...and these Grafana resources
OSM installation	System Metrics, Admin Project-scoped	Admin-privileges
OSM Projects	Project-scoped (Grafana “team” privileges)	Grafana “team” privileges
OSM Users	-	Grafana users to teams
OSM Network Services	NS-scoped	-

`...Let’s play with the prototype!`

Open Source MANO

Hands-on!

OSM Service Assurance

Let’s play with metrics and (auto)dashboards!

1. From your SSH console, download new descriptors and upload them to OSM

```
wget http://osm-download.etsi.org/ftp/osm-6.0-six/8th-hackfest/packages/hackfest_basic_metrics_vnfd.tar.gz
wget http://osm-download.etsi.org/ftp/osm-6.0-six/8th-hackfest/packages/hackfest_basic_metrics_nsd.tar.gz
osm vnfd-create hackfest_basic_metrics_vnfd.tar.gz
osm nsd-create hackfest_basic_metrics_nsd.tar.gz
```

2. Create your VIM & instantiate your NS

```
osm vim-create --name whitecloud_XX --user osm_hackfest_XX --password osm_hackfest_XX --auth_url
http://172.21.247.1:5000/v3 --tenant osm_hackfest_XX --account_type openstack

osm ns-create --ns_name hfmetrics_XX --nsd_name hackfest_basic-ns-metrics --vim_account whitecloud_XX --config
'{vld: [ {name: mgmtnet, vim-network-name: osm-ext} ] }'
```

3. Go and check how your own “project dashboard” starts to be populated.

   Then, look for a new dashboard dedicated to your NS! → http://172.21.248.14:3000/dashboards

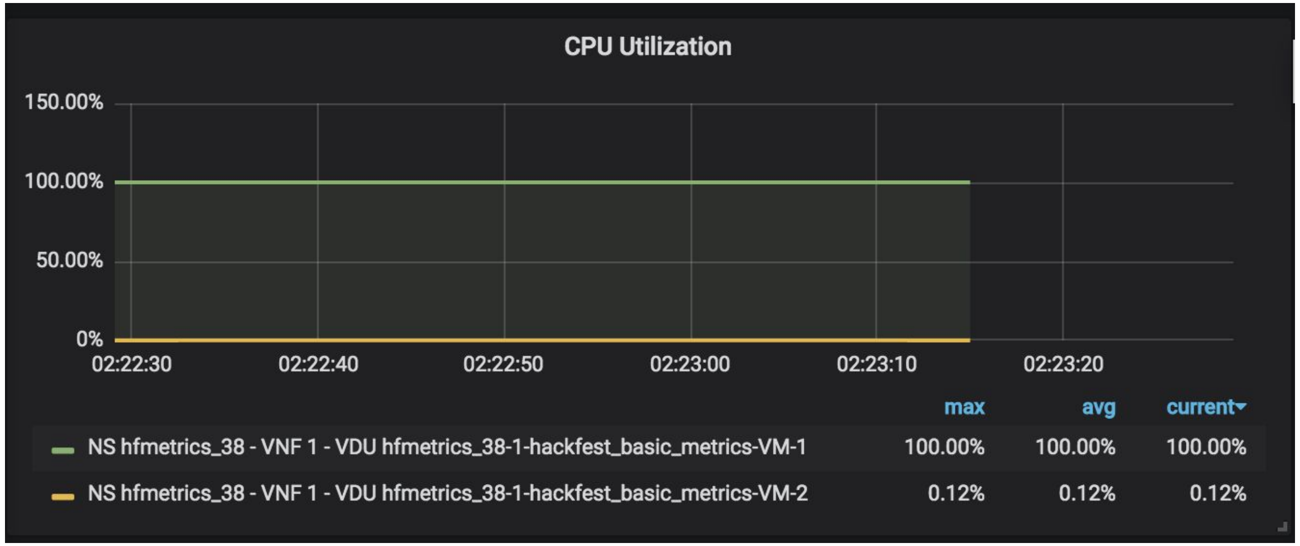
Let’s play with autoscaling!

1. Access your VM and stress it out!

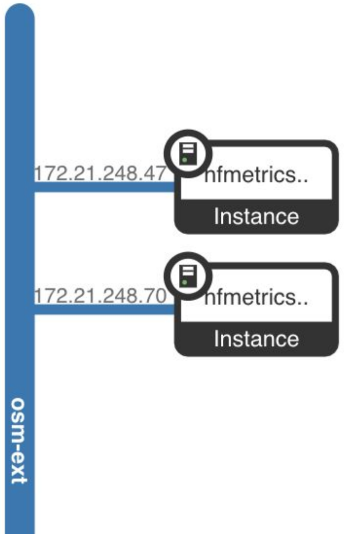
```
osm vnf-list                    # to find the IP address
ssh ubuntu@172.21.248.93        # password: osm4u

yes > /dev/null &    # 4 or 5 times!
```

2. Wait for a bit (5 to 10 minutes due to current collection period), and watch it scale!



	max	avg	current▾
🟩 NS hfmetrics_38 - VNF 1 - VDU hfmetrics_38-1-hackfest_basic_metrics-VM-1	100.00%	100.00%	100.00%
🟨 NS hfmetrics_38 - VNF 1 - VDU hfmetrics_38-1-hackfest_basic_metrics-VM-2	0.12%	0.12%	0.12%



Open Source MANO

Find us at:

osm.etsi.org
osm.etsi.org/wikipub

ETSI