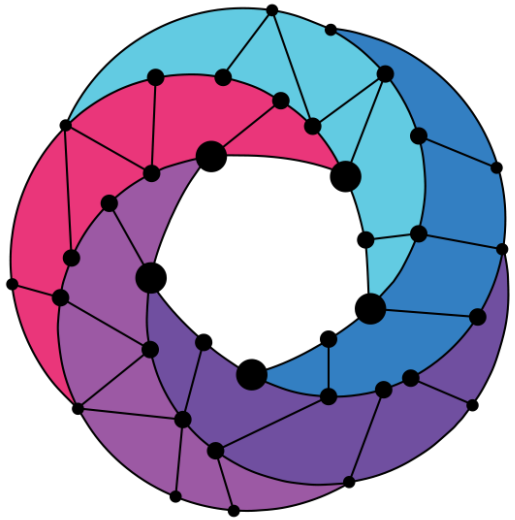


Open Source
MANO

OSM-MR10, OSM Ecosystem Day

On the use of OSM to allow for automated network slice scaling in multi-site environments

Jose Ordonez-Lucena (Telefónica)

PoC overview

- Presentation of ZSM PoC#2
  - **PoC objective**: Demonstrate the capacity to automatically scale out a deployed network slice instance across multiple administrative domains
  - **PoC timeline**: 01.01.2021– 31.03-2021 (3 months)
- Alignment with ZSM
  - ZSM001 (Use Cases) and ZSM003 (Network slicing features)

PoC Team

Telefonica

UNIVERSITY OF PATRAS

uc3m Universidad Carlos III de Madrid

telenor

openslice

PoC assets

5G-VINNI

5TONIC

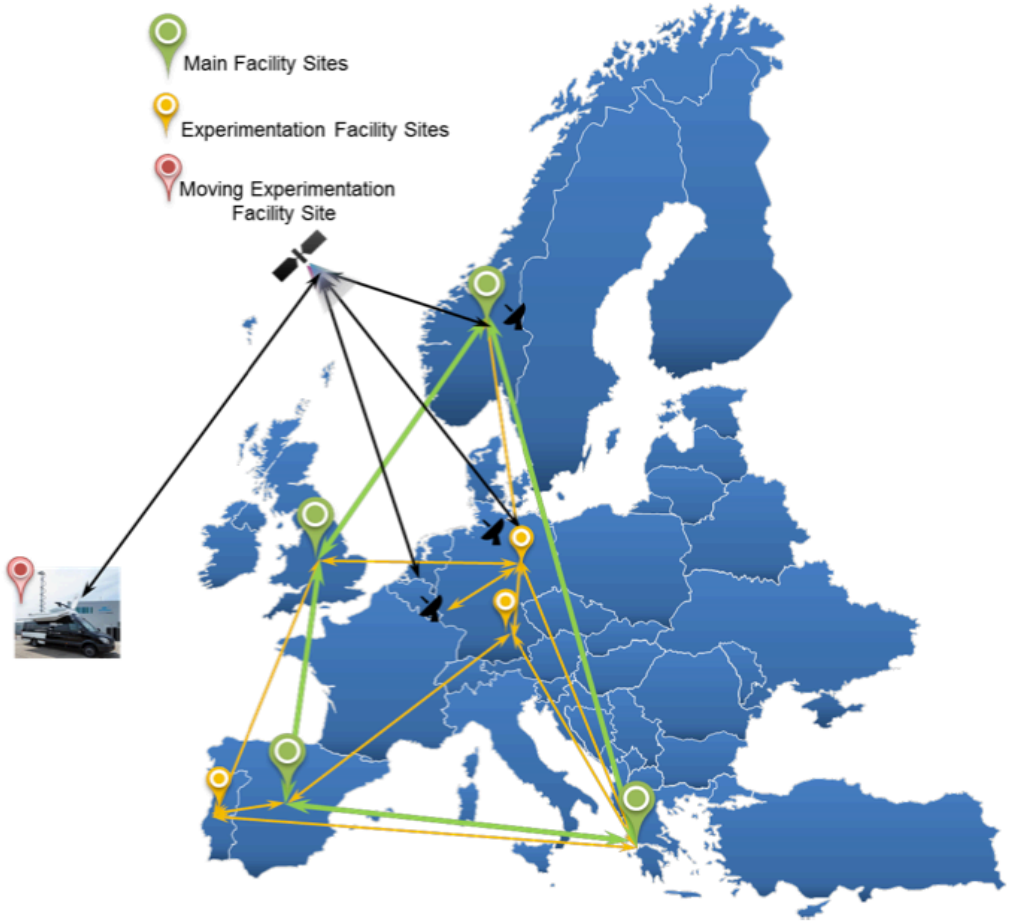
Open Source MANO

Agenda

- Background information
- PoC: Automated Network Slice Scaling In Multi-Site Environments
- PoC: OSM domain (steps 1-6)
- Concluding remarks

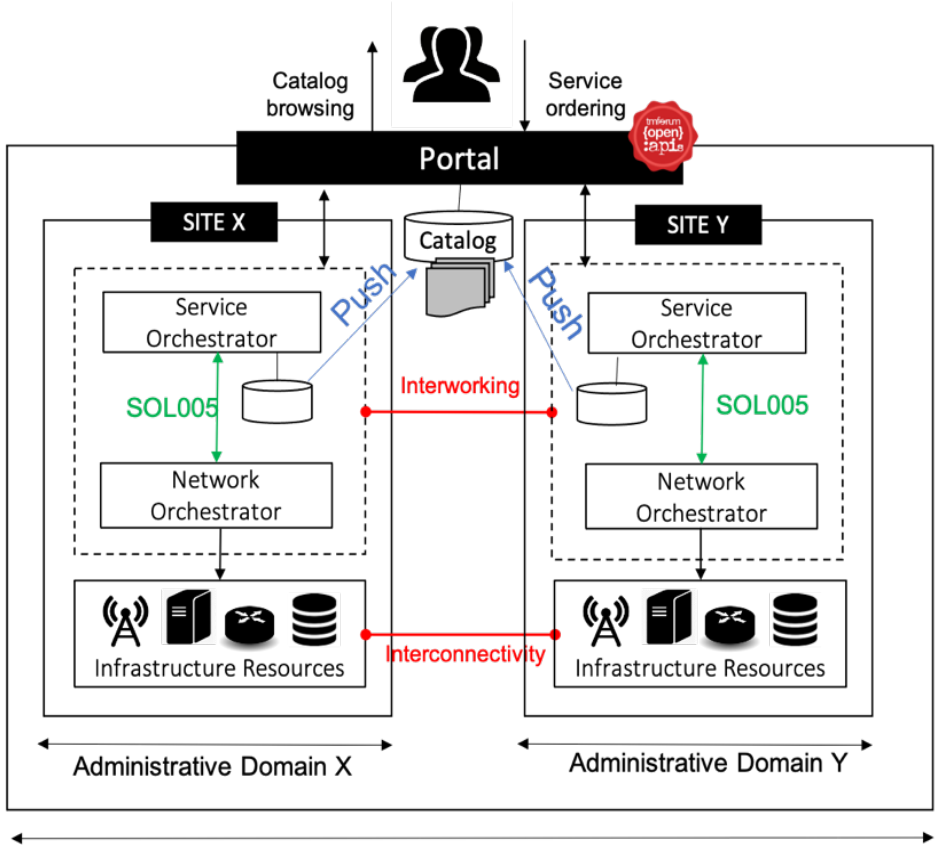
5G-VINNI project

- **5G-VINNI project's vision:** build an open large-scale 5G End-to-End (E2E) facility that can
  - demonstrate that they 5G network KPIs can be met
  - be validated, accessed and used by vertical industries to test use cases and validate KPIs.
- 5G-VINNI built out of a set of **interworking** facility sites
- **Main facility sites**: offer E2E 5G network capabilities to real-world verticals, with well-defined SLAs
  - Oslo-Kongsberg (Norway); Martlesham (UK); Madrid (Spain); Patras (Greece)
- **5G-VINNI experimentation facility site**s: provide environment for advanced experimentation and testing possibilities on (combination of) elements of the E2E model
  - Aveiro (Portugal); Berlin (Germany); Munich (Germany)
- **Moving experimentation Facility site**: satellite connected vehicle

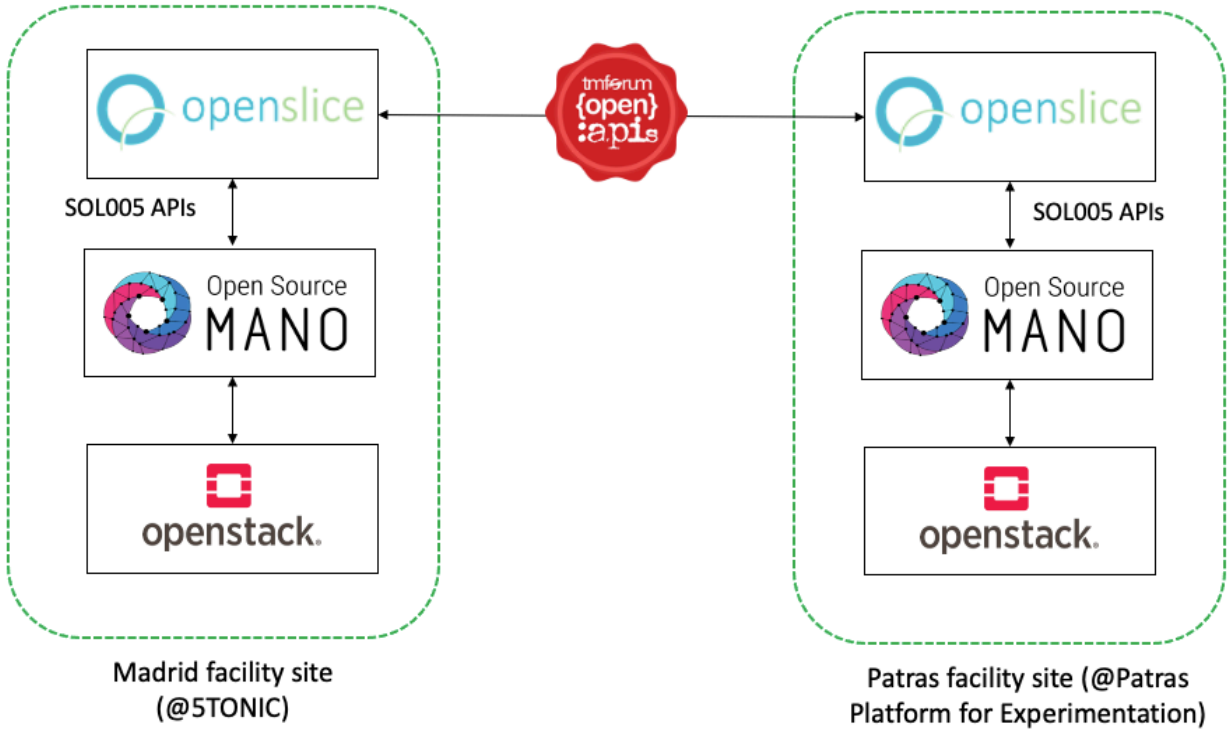


5G-VINNI facility architecture

From a generic 5G-VINNI facility blueprint…



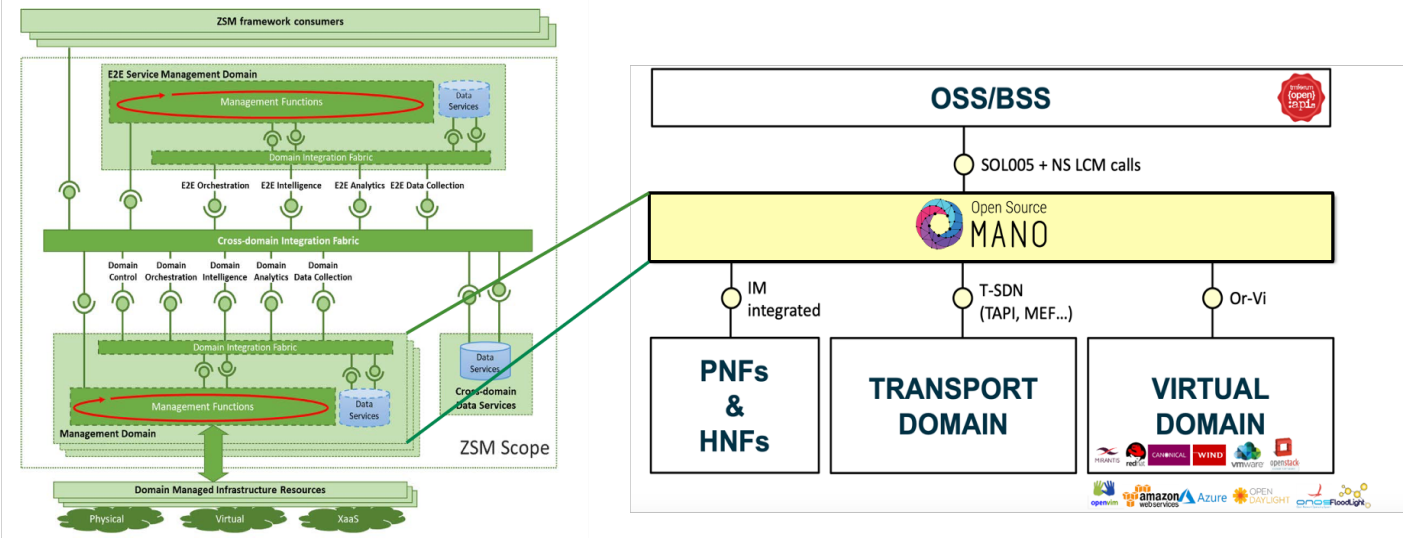
…. to a specific 5G-VINNI facility realization

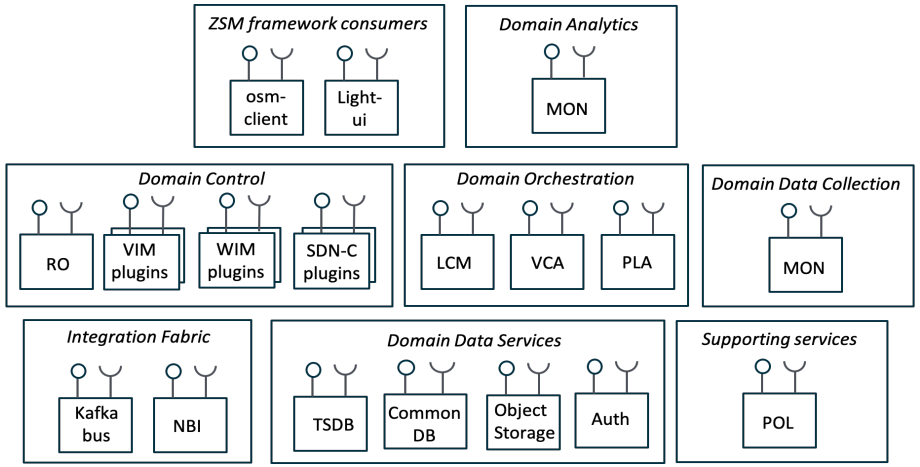


The PoC makes use of the assets from these two 5G-VINNI facility sites

OSM – Release EIGHT

- Alignment with ETSI ISG ZSM architecture (cf. ZSM002)
  - OSM can be mapped to a ZSM management domain
  - OSM stack aligned with ZSM architecture principles (SBMA, modularity, scalability, zero-touch,…)
- From OSM perspective, OpenSlice behaves as upper OSS/BSS layer
  - OpenSlice (Service Orchestrator) consumes OSM NBI capabilities.

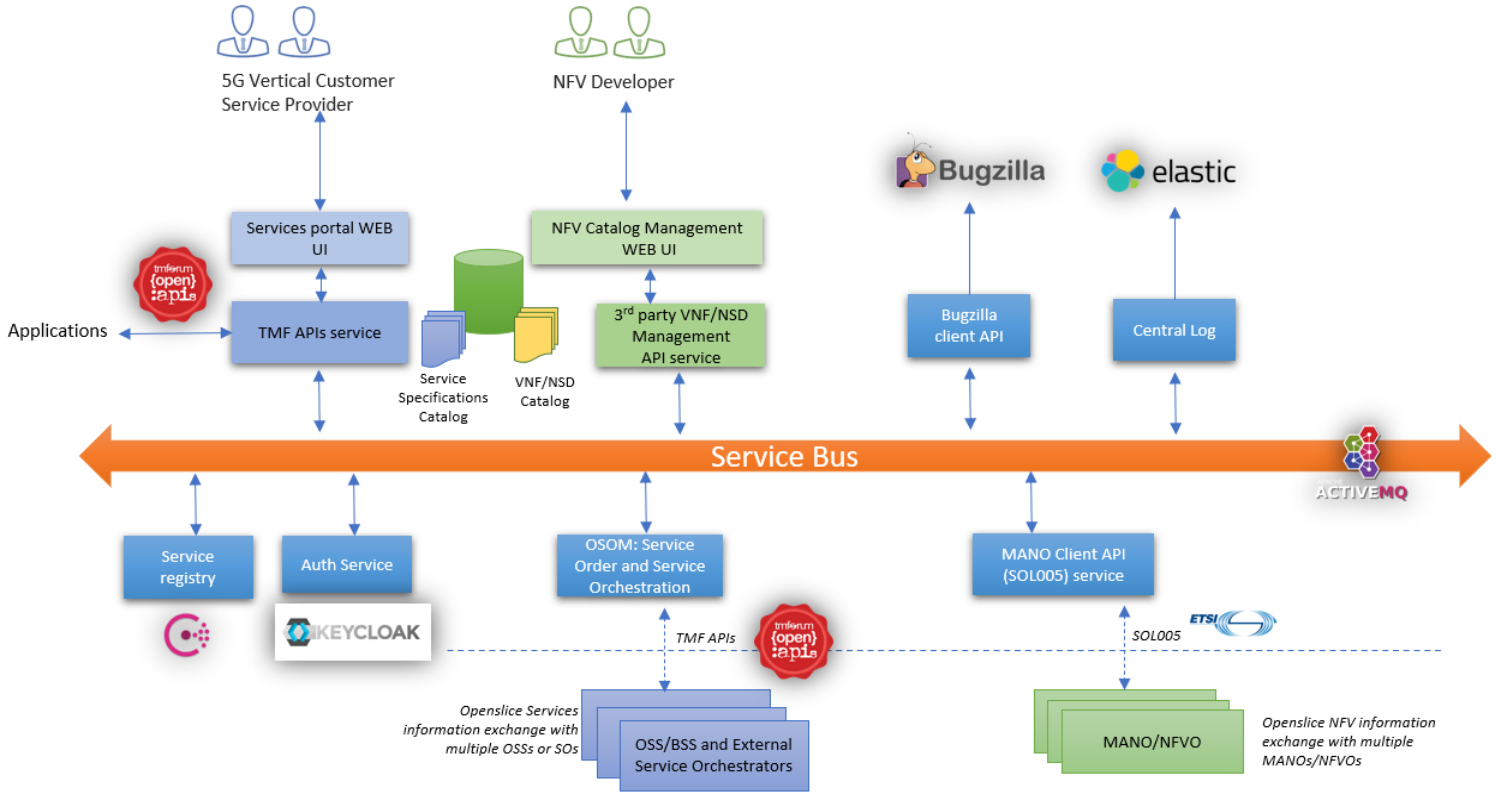




On the mapping of OSM modules with ZSM services

OpenSlice

- **Open-source** operations support system (OSS) solution providing **Service Orchestration** functionality
  - Including both service fulfillment and assurance lifecycle phases
- **CFS viewpoint**: user-friendly web portal for the interaction with the vertical customers:
  - Browse slice templates (VINNI-SB's) in the Service Catalog.
  - Issue and capture service orders
  - Retrieve PM/FM data on deployed slices.
- **RFS viewpoint**: consumption of OSM NBI capabilities to deploy and operate the virtualized components of the slices.



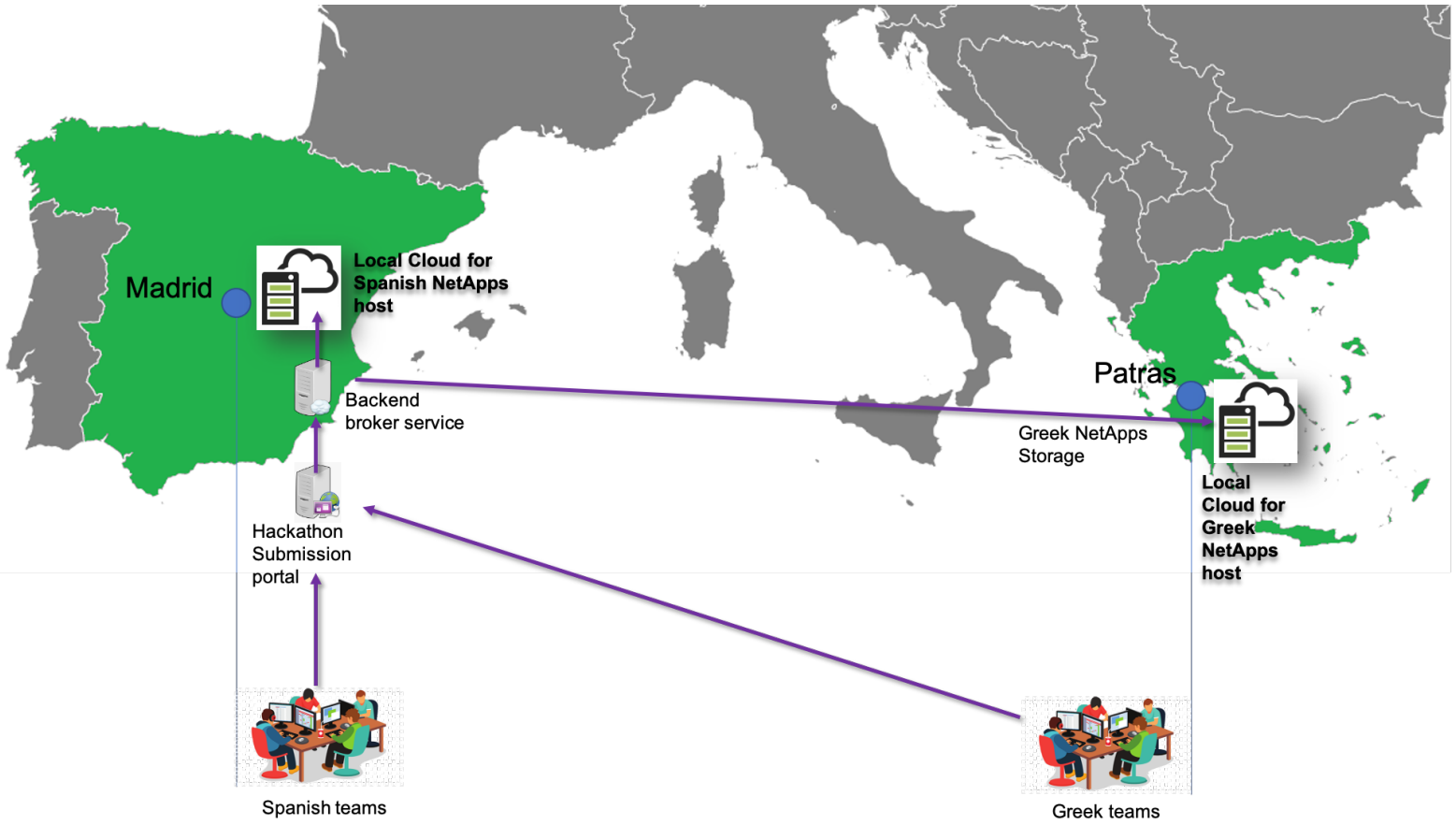
Link: https://openslice.io

Open Source MANO

Agenda

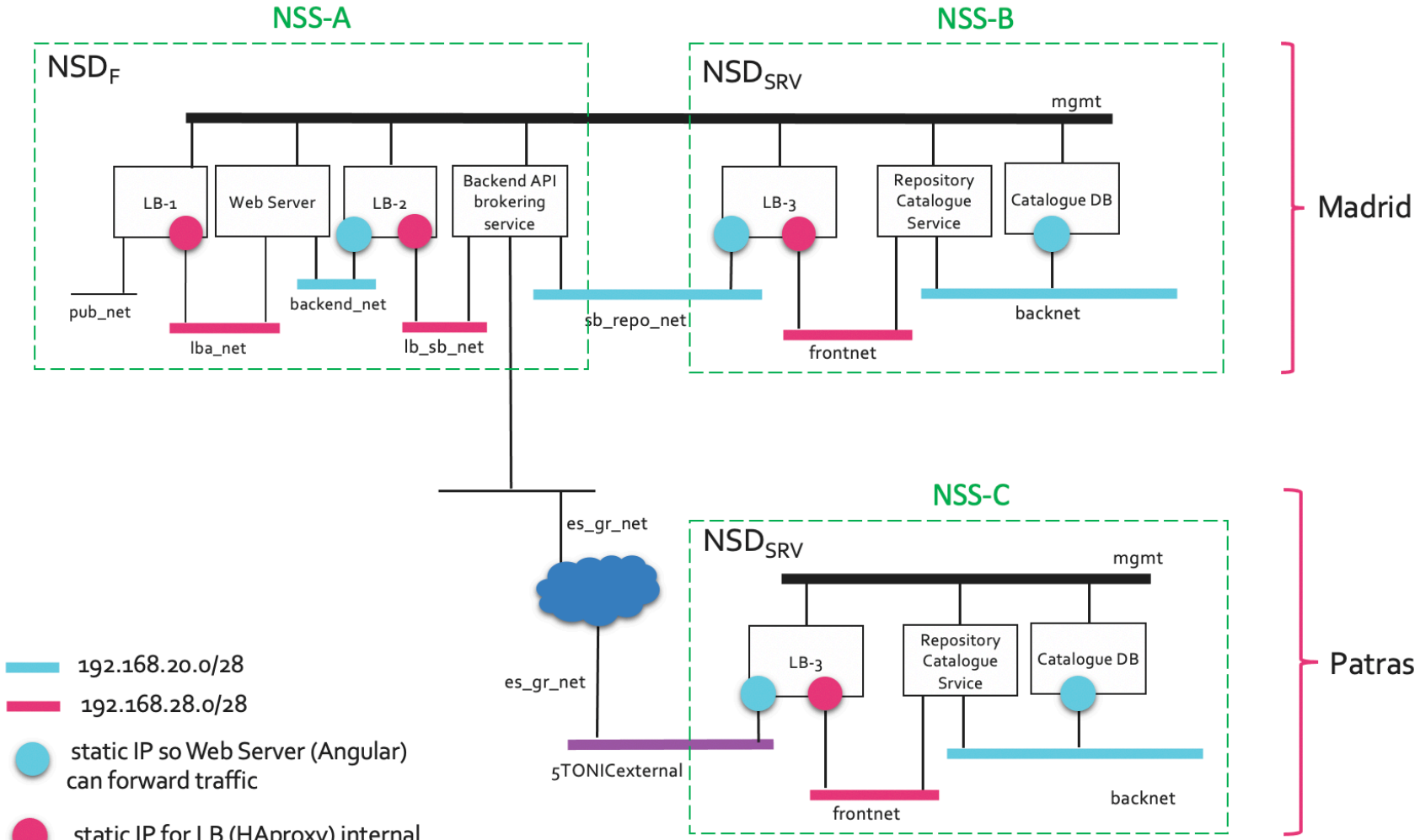
- Background information
- PoC: Automated Network Slice Scaling In Multi-Site Environments
- PoC: OSM domain (steps 1-6)
- Concluding remarks

Scenario description



- Vertical industry (e.g., PPDR, e-Health) NetApps hackathon
  - Involving developers from Spain and Greece
  - NetApp submission service hosted in Madrid facility site
  - Re: GPDR policy, NetApps binaries and data must be hosted in the home country.
- Network slice scaling out operation:
  - Automatically triggered in Madrid, because of unexpected load surges -> **reactive correction action**
  - Propagated towards Patras, due to forecasting reasons -> **pro-active corrective action**

Multi-site NW slice – design & deployment



VNFs

LB-1, LB-2, LB-3: LB_VNF (by OSM)

Web Server: Wserver_VNF:

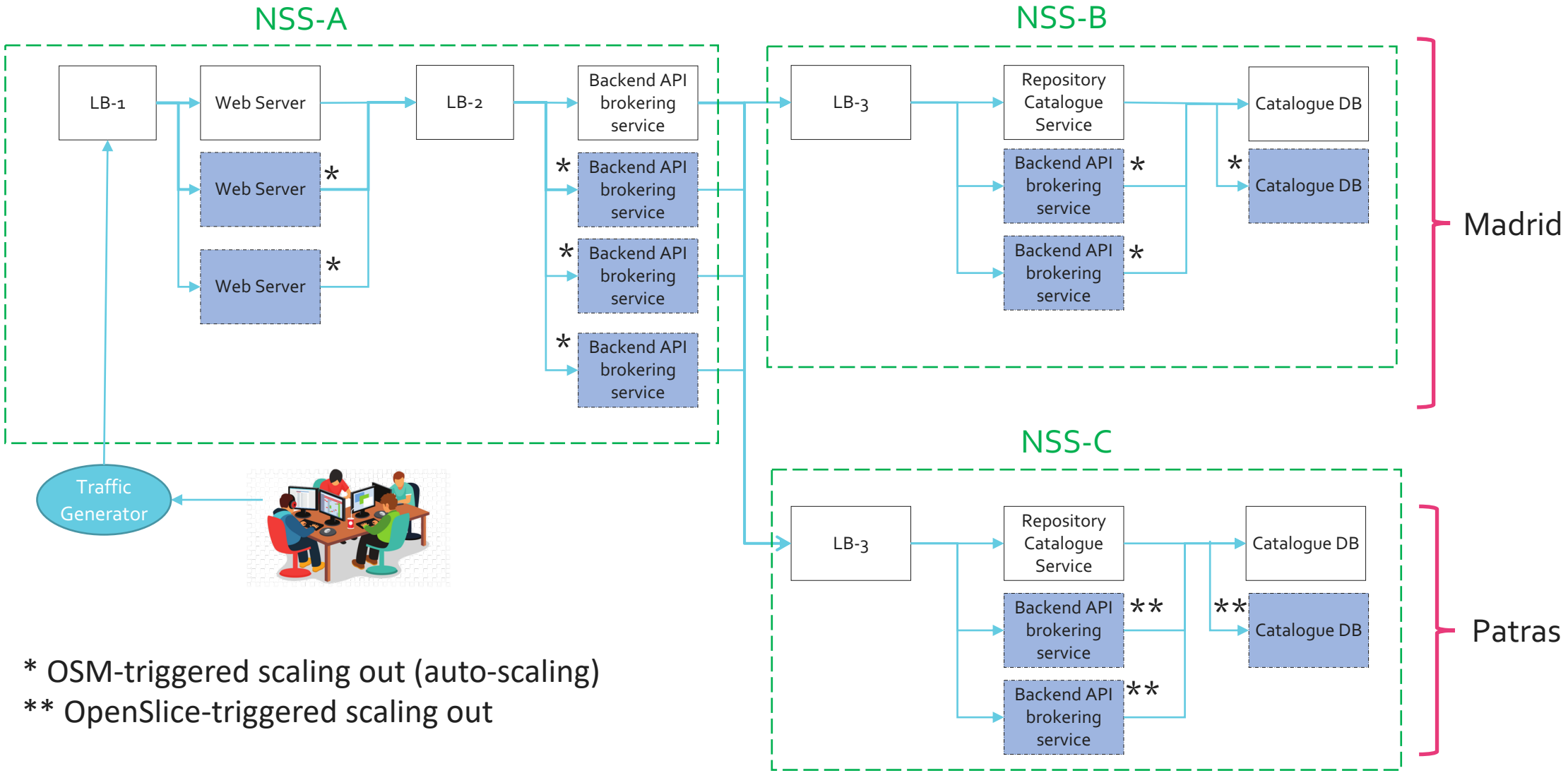
- Scalable – VNF metrics, granularity 5m

Backend API brokering service: Wserver2_VNF

- Scalable – VNF metrics, granularity 5m)

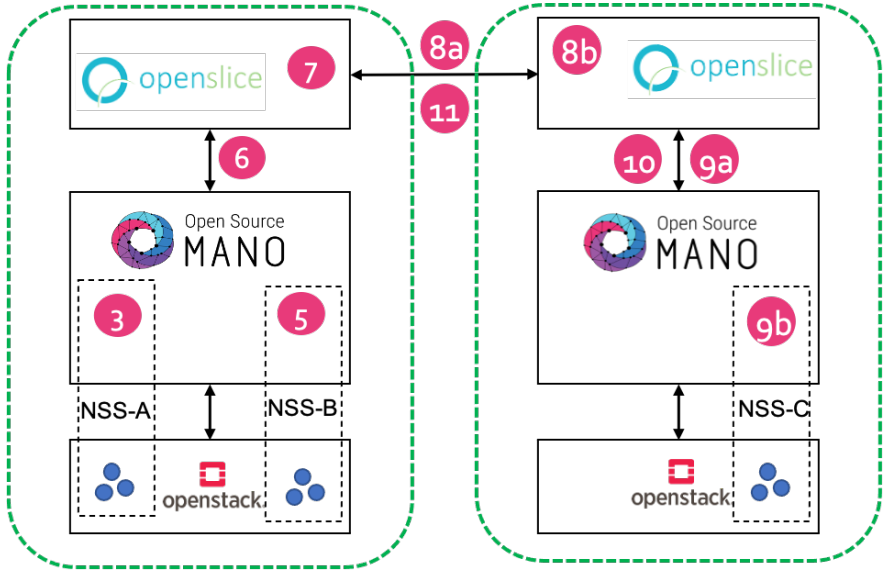
Catalogue DB: Generic_VNF (Vanilla Ubuntu 20.04)

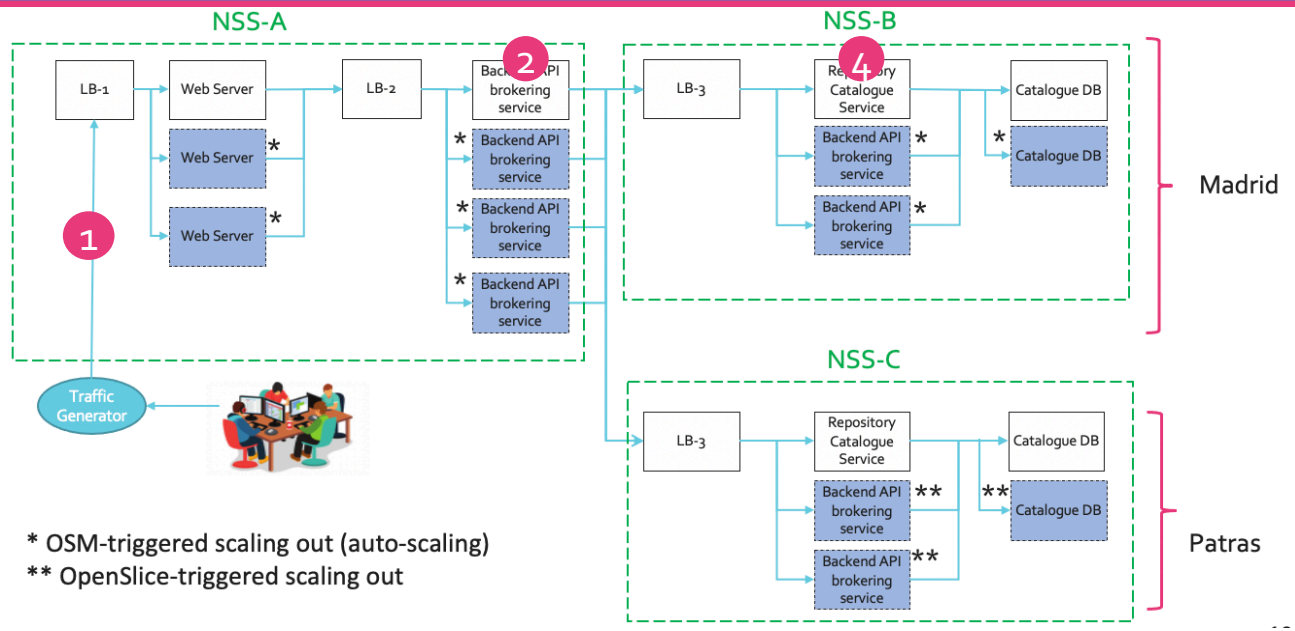
Multi-site NW slice – operation



* OSM-triggered scaling out (auto-scaling)
** OpenSlice-triggered scaling out

PoC user story





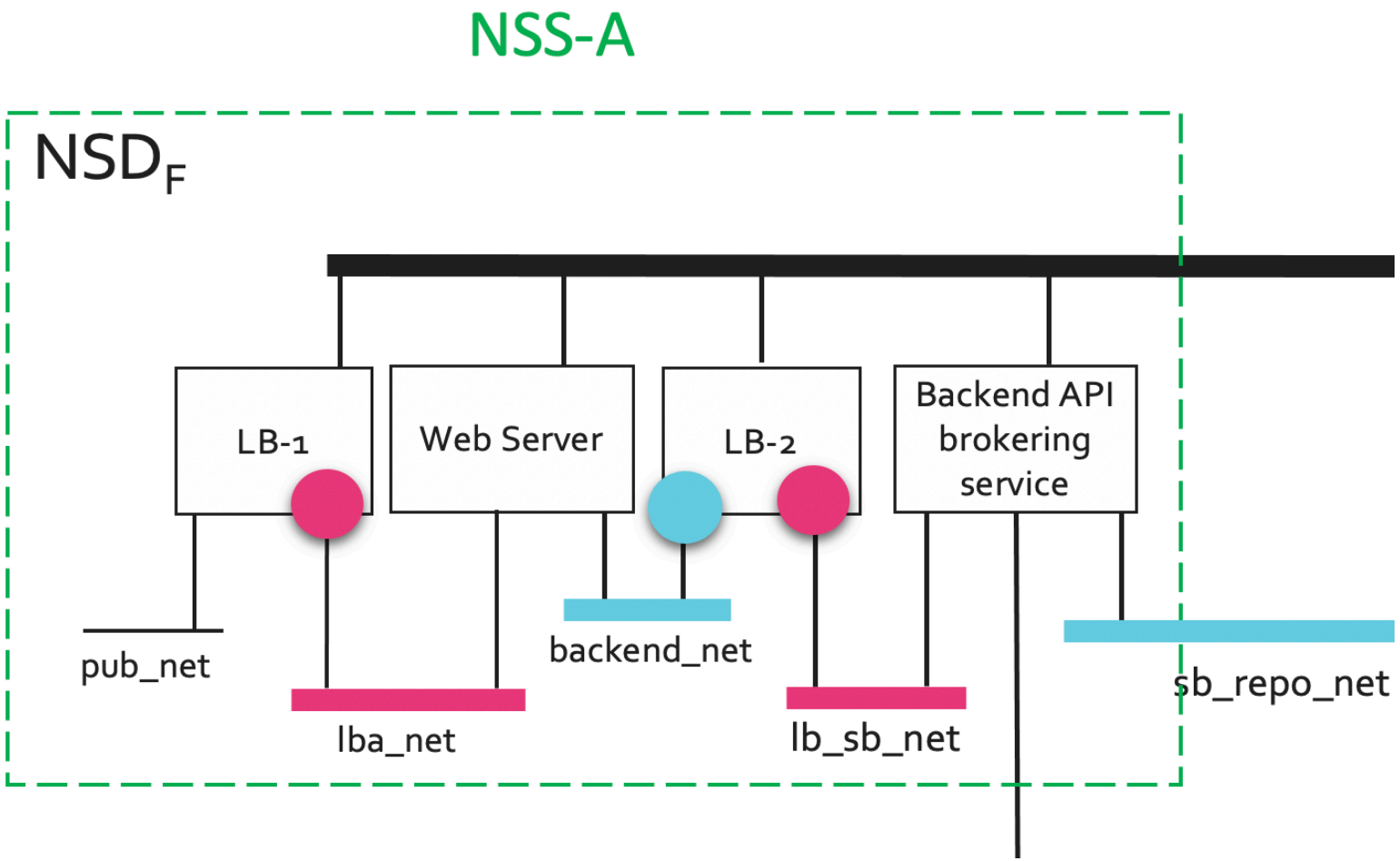
1. There is a sudden high demand of portal interaction at Madrid facility site (HTTP requests represents a traffic load surge with 3:1 ratio)
2. NSS-A's backend API brokering service collapses, being not able to forward traffic to etiher NSS-B or NSS-C
3. Based on day-2 activities, "Madrid-OSM" triggers **NSS-A auto-scaling** -> Web server (2 x scale out), LB-1 (reconfiguration), LB-2 (reconfiguration)
4. NSS-A's backend API brokering service back on normal operation, and starts sending traffic to NSS-B through LB-3. NSS-B's VNFs collapse.
5. Based on day-2 activities, "Madrid-OSM" triggers **NSS-B auto-scaling** -> Repository catalogue (2x scale out), DB (1 x scale out), LB-3 (reconfiguration)
6. "Madrid-OSM" notifies "Madrid-OpenSlice" of successul steps 3 and 5
7. "Madrid-OpenSlice" decides that NSS-C needs to be scaled out as NSS-B did, in order to avoid collapse as in Madrid -> OpenSlice is aware of UC semantics
8. "Madrid-OpenSlice" issues NSS-C scaling request to "Patras-OpenSlice", using TMF's APIs. "Patras-OpenSlice" checks this request.
9. "Patras-OpenSlice" forwards the request to the "Patras-OSM" for enforcement. Unlike step 3 and 5, here **there is no NSS-C auto-scaling**
10. "Patras-OSM" notifies "Madrid-OpenSlice" of NSS-C scaling out.
11. "Patras-OpenSlice" notifies "Madrid-OpenSlice" of successful NSS-C scaling out.

Open Source MANO

Agenda

- Background information
- PoC: Automated Network Slice Scaling In Multi-Site Environments
- PoC: OSM domain (steps 1-6)
- Concluding remarks

NSS-A Topology



- LB_VNF:
  - CPU: 4
  - RAM: 4 GB
  - Disk: 10 GB
  - Name in NSD-F: "haproxy_ubuntu" [image]
- **Wserver_VNF (Scalable – VNF metrics, granularity 5min):**
  - CPU: 1
  - RAM: 1 GB
  - Disk: 10 GB
  - Name in NSD-F: "ubuntu-20.04-server-cloudimg-amd64" [image]

Wserver_VNF: VNFD

- Collecting VNF metrics
  - A simple charm containing a metrics.yaml file at its root folder specifies the metrics to be collected and associated comments

```yaml
metrics:
    users:
      type: gauge
      description: "# of users"
      command: who|wc -l
    load:
      type: gauge
      description: "5 minute load average"
      command: cat /proc/loadavg |awk '{print $1}'
    load_pct:
      type: gauge
      description: "1 minute load average percent"
      command: cat /proc/loadavg  | awk '{load_pct=$1*100.00} END {print load_pct}'
```

```yaml
    vnf-configuration:
      initial-config-primitive:
      - name: config
        parameter:
        - name: ssh-hostname
          value: <rw_mgmt_ip>
        - name: ssh-username
          value: "username"
        - name: ssh-password
          value: "password"
        seq: '1'
      juju:
        charm: testmetrics
        proxy: true
      metrics:
        - name: users
        - name: load
        - name: load_pct
```

```yaml
    monitoring-param:
    -   id: "Wserver_users"
        name: "Wserver_users"
        aggregation-type: AVERAGE
        vnf-metric:
          vnf-metric-name-ref: "users"
    -   id: "Wserver_load"
        name: "Wserver_load"
        aggregation-type: AVERAGE
        vnf-metric:
          vnf-metric-name-ref: "load"
    -   id: "Wserver_load_pct"
        name: "Wserver_load_pct"
        aggregation-type: AVERAGE
        vnf-metric:
          vnf-metric-name-ref: "load_pct"
```

Wserver_VNF: VNFD

- Autoscaling

Metric	Value
Minimum instance count	0
Maximum instance count	3
CPU-scale-in-threshold (%)	20
CPU-scale-out-threshold (%)	80
threshold-time (s)	10
cooldown-time (s)	30

```
72  scaling-group-descriptor:
73  -   name: "Wserver_autoscale"
74      min-instance-count: 0
75      max-instance-count: 3
76      scaling-policy:
77      -   name: "Wserver_load_pct_above_threshold"
78          scaling-type: "automatic"
79          threshold-time: 10
80          cooldown-time: 30
81          scaling-criteria:
82          -   name: "Wserver_load_pct_above_threshold"
83              scale-in-threshold: 20
84              scale-in-relational-operation: "LT"
85              scale-out-threshold: 80
86              scale-out-relational-operation: "GT"
87              vnf-monitoring-param-ref: "Wserver_load_pct"
88      vdu:
89      -   vdu-id-ref: Wserver
90          count: 1
```

Open Source MANO

Agenda

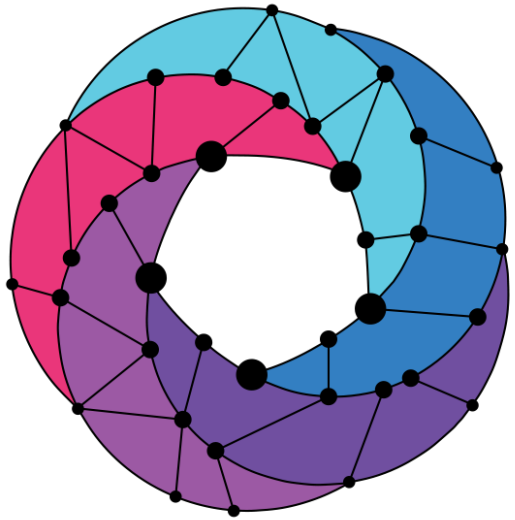
- Background information
- PoC: Automated Network Slice Scaling In Multi-Site Environments
- PoC: OSM domain (steps 1-6)
- Concluding remarks

Golden nuggets

- ZSM poC#2 pipeline, with all ingredients
  - Research + Experimentation + Standardization
  - Open-source communities (OSM and OpenSlice) along the entire path

- ZSM PoC#2 showcasing by early April
  - Current status: automated scaling at individual facility sites (steps 1-6)
  - Next steps: Cross-domain orchestration @OpenSlice (steps 7-11)

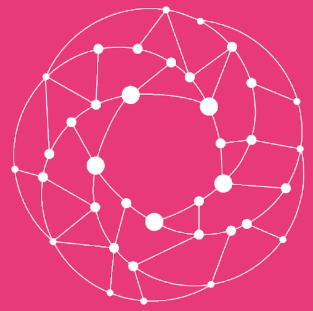
- PoC#2 wikipage: https://zsmwiki.etsi.org/index.php?title=PoC_2_Automated_Network_Slice_Scaling_in_Multi-Site_Environments



Open Source MANO

Find us at:

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



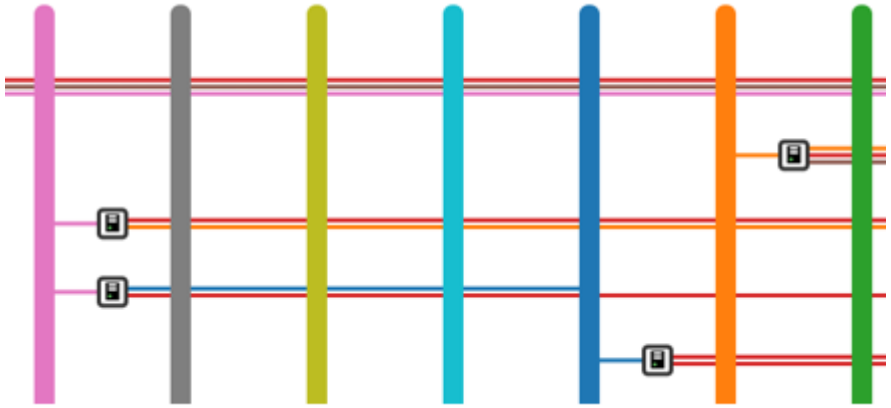
Open Source
MANO

Annex A

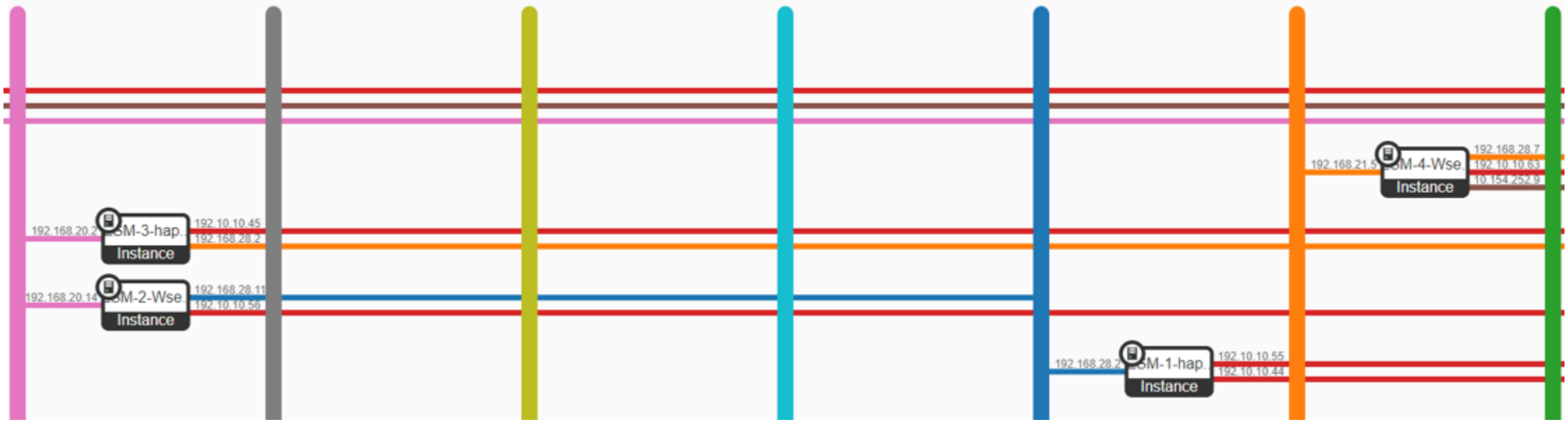
Scaling out in action

ETSI

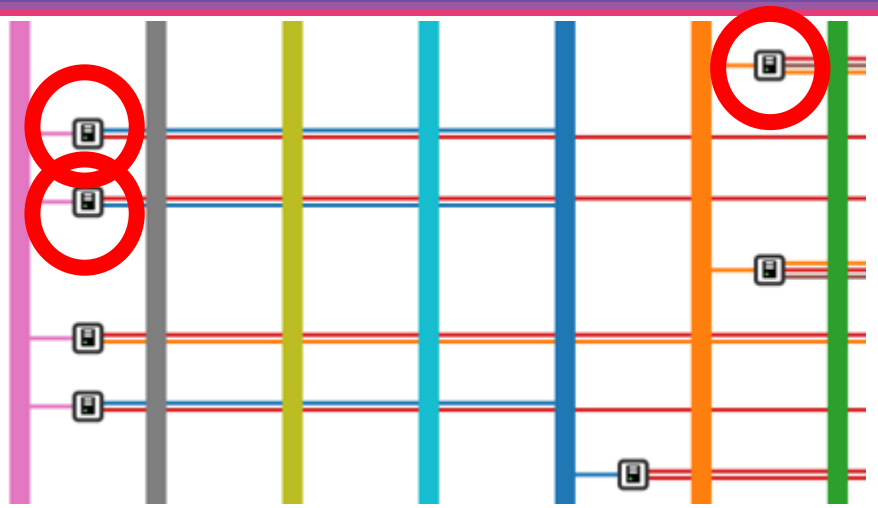
Scaling in action (pre-conditions)



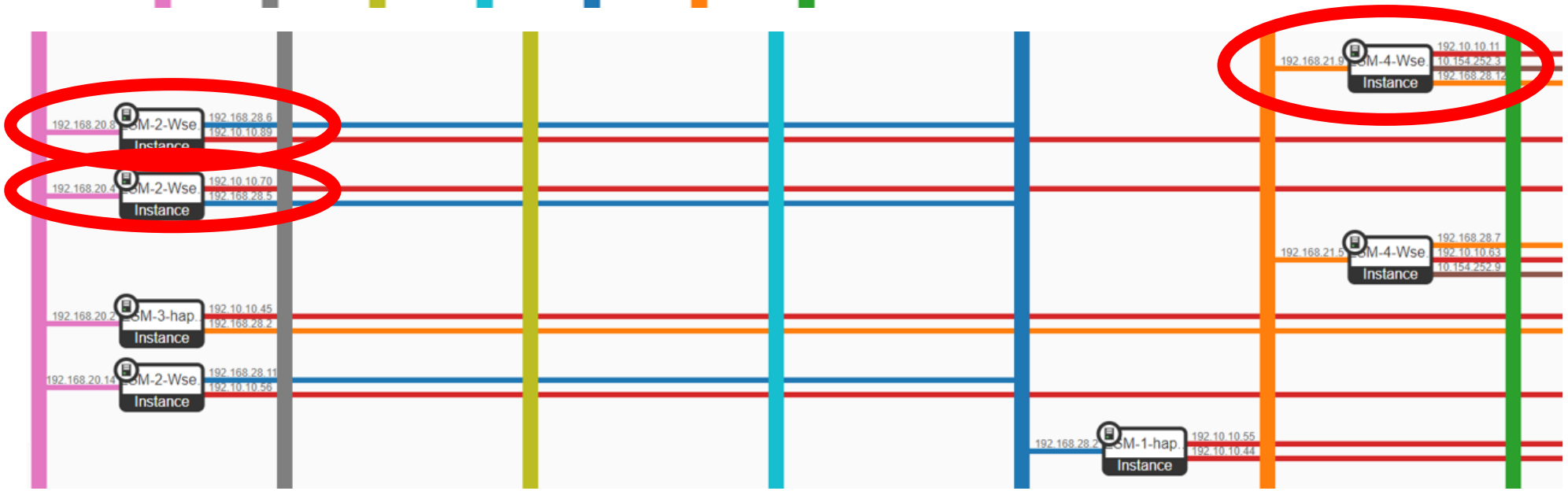
OpenStack topology view



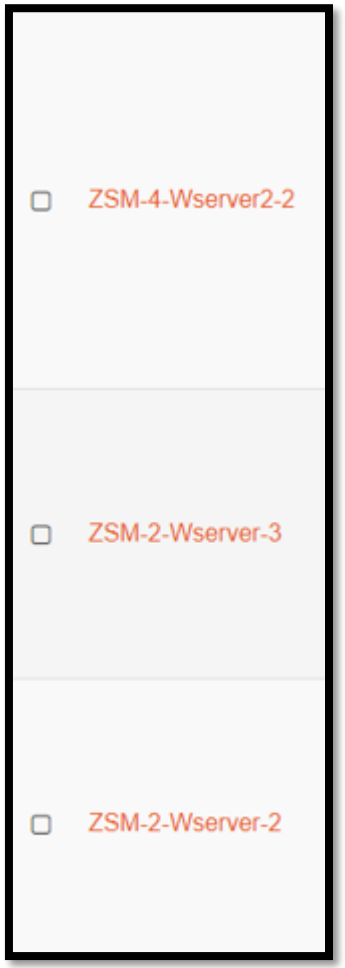
Scaling in action (post-conditions)



OpenStack topology view

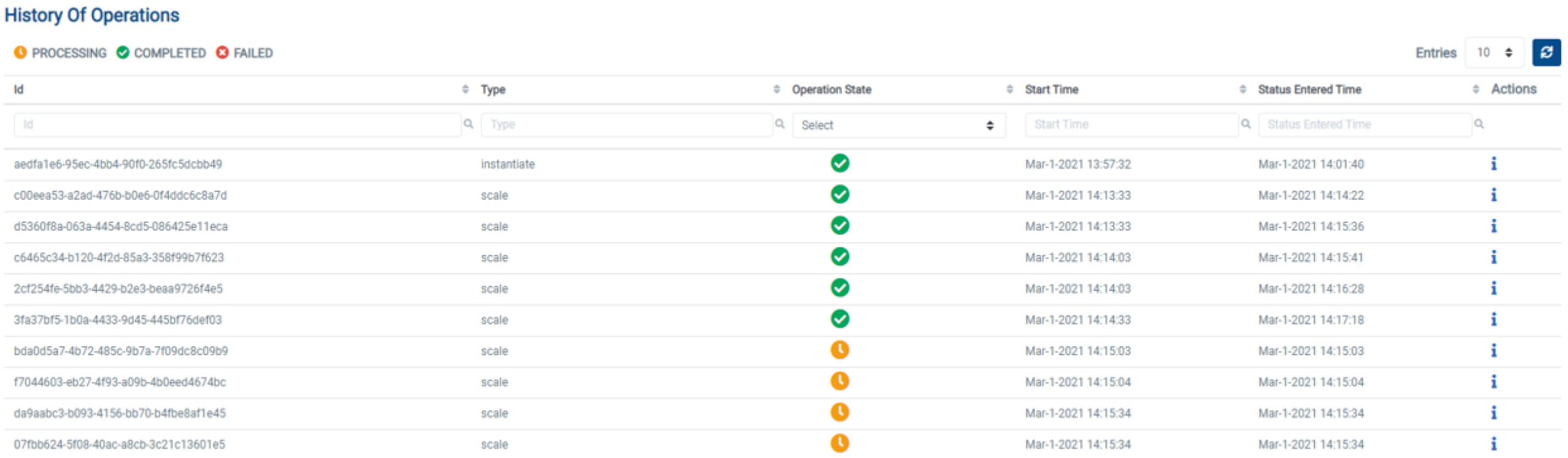


OpenStack instances view



OSM's History of Operations

Some scaling actions are completed successfully, and more are happening



LB-1 (HAProxy) Dashboard

The dashboard shows the new nodes (1 initial plus 2 scaled out)

HAProxy

Statistics Report for pid 19594

> General process information

pid = 19594 (process #1, nbproc = 1)
uptime = 0d 0h06m16s
system limits: memmax = unlimited; ulimit-n = 4054
maxsock = 4054; maxconn = 2000; maxpipes = 0
current conns = 1; current pipes = 0/0; conn rate = 1/sec
Running tasks: 1/18; idle = 100 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance
backup UP
backup UP, going down
backup DOWN, going up
not checked
Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:
- Scope :
- Hide 'DOWN' servers
- Refresh now
- CSV export

External resources:
- Primary site
- Updates (v1.5)
- Online manual

haproxynode

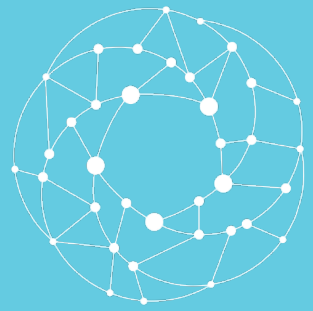
	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	0	-	0	0	2 000	0			0	0	0	0	0					OPEN								

backendnodes

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
node1	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	6m16s DOWN	L4CON in 0ms	1	Y	-	1	1	6m16s	-
node2	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	6m14s DOWN	L4TOUT in 2000ms	1	Y	-	1	1	6m14s	-
node3	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	2m37s UP	L4OK in 0ms	1	Y	-	1	1	3m37s	-
node4	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	1m24s UP	L4OK in 0ms	1	Y	-	1	1	4m51s	-
node5	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	6m14s DOWN	* L4TOUT in 2001ms	1	Y	-	1	1	6m14s	-
node6	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	6m15s DOWN	* L4TOUT in 2001ms	1	Y	-	1	1	6m15s	-
node7	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	6m15s DOWN	* L4TOUT in 2000ms	1	Y	-	1	1	6m15s	-
node8	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	6m13s DOWN	L4TOUT in 2000ms	1	Y	-	1	1	6m13s	-
node9	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	6m16s UP	L4OK in 0ms	1	Y	-	0	0	0s	-
node10	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	6m13s DOWN	L4CON in 0ms	1	Y	-	1	1	6m13s	-
node11	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	6m12s DOWN	* L4TOUT in 2001ms	1	Y	-	1	1	6m12s	-
node12	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	6m12s DOWN	* L4TOUT in 2001ms	1	Y	-	1	1	6m12s	-
Backend	0	0		0	0		0	0	200	0	0	?	0	0	0	0		0	0	0	0	6m16s UP		3	3	0		0	0s	

stats

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				2	2	-	2	2	2 000	2			0	0	0	0	0					OPEN								
Backend	0	0		0	0		0	0	200	0	0	0s	0	0	0	0		0	0	0	0	6m16s UP		0	0	0		0		



Open Source
MANO

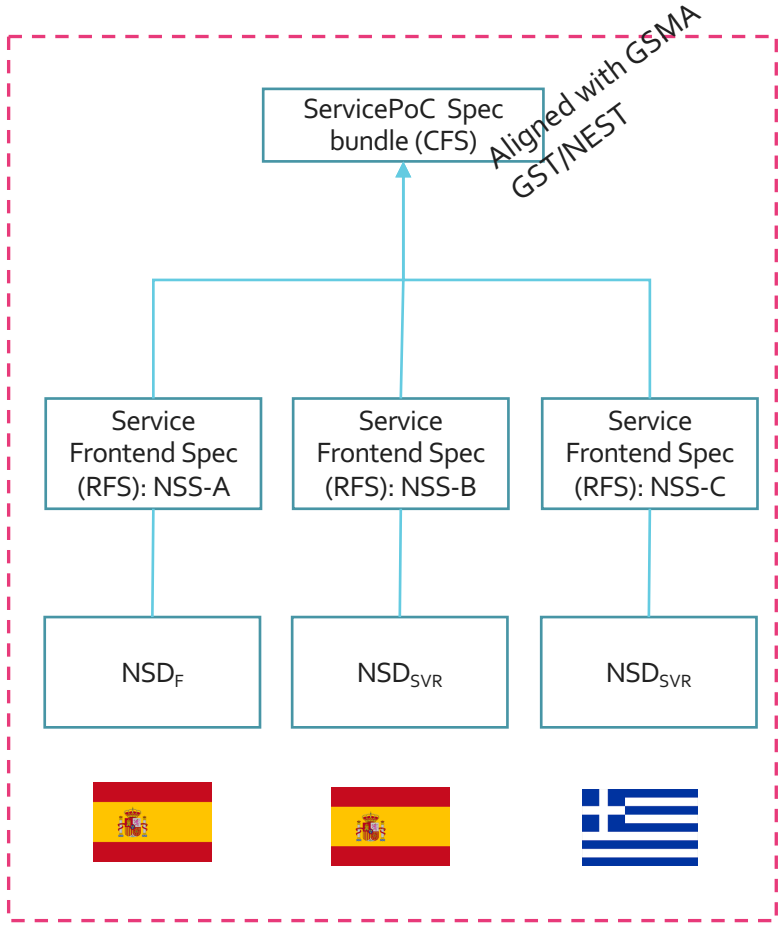
Annex B

The role of OpenSlice in the PoC

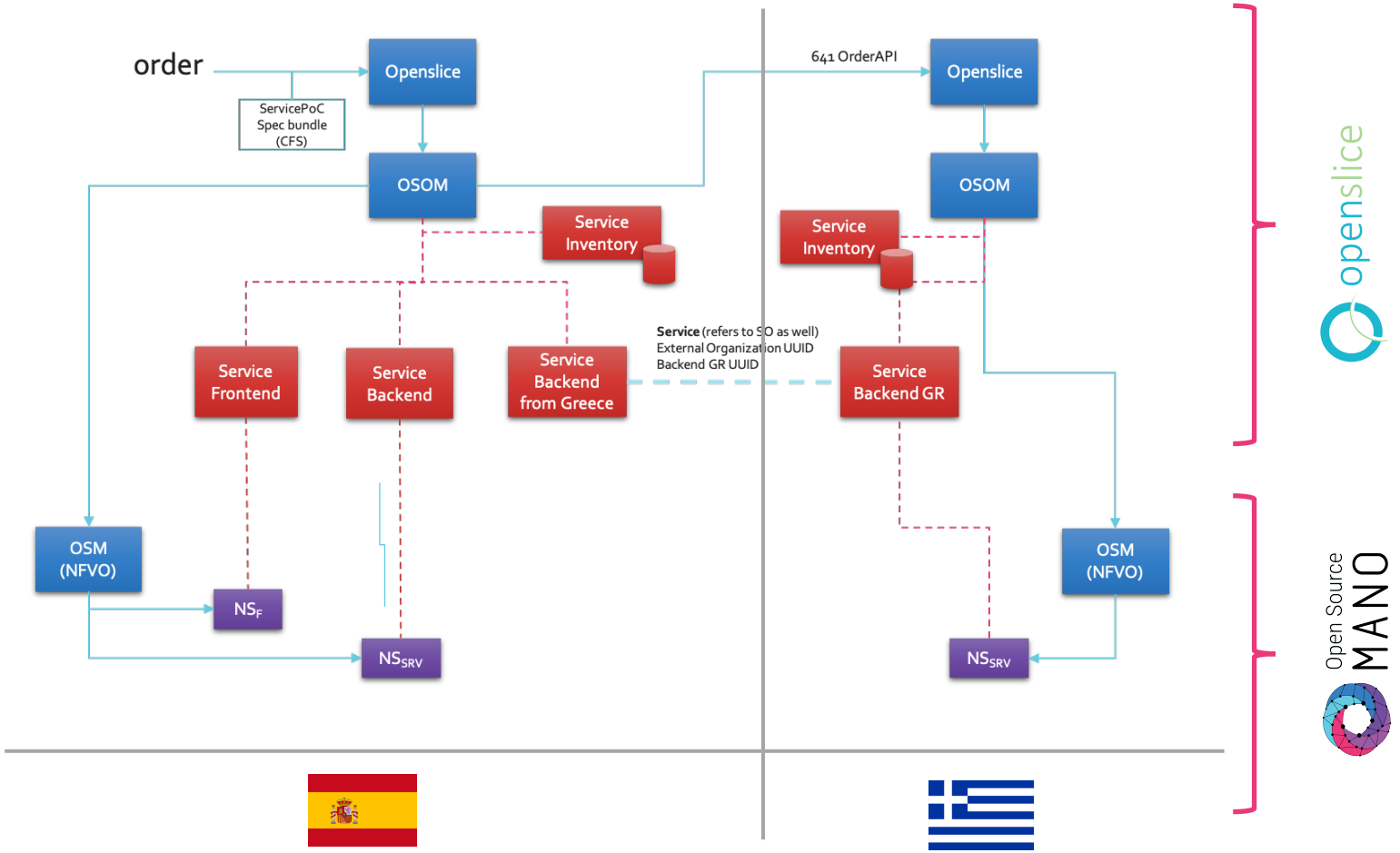
ETSI

OpenSlice – service fulfillment

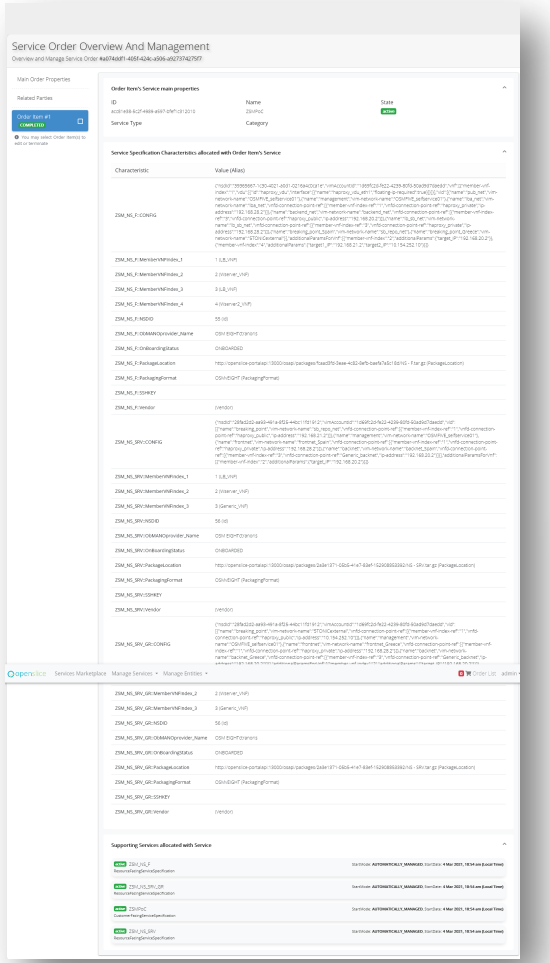
Service specification tree



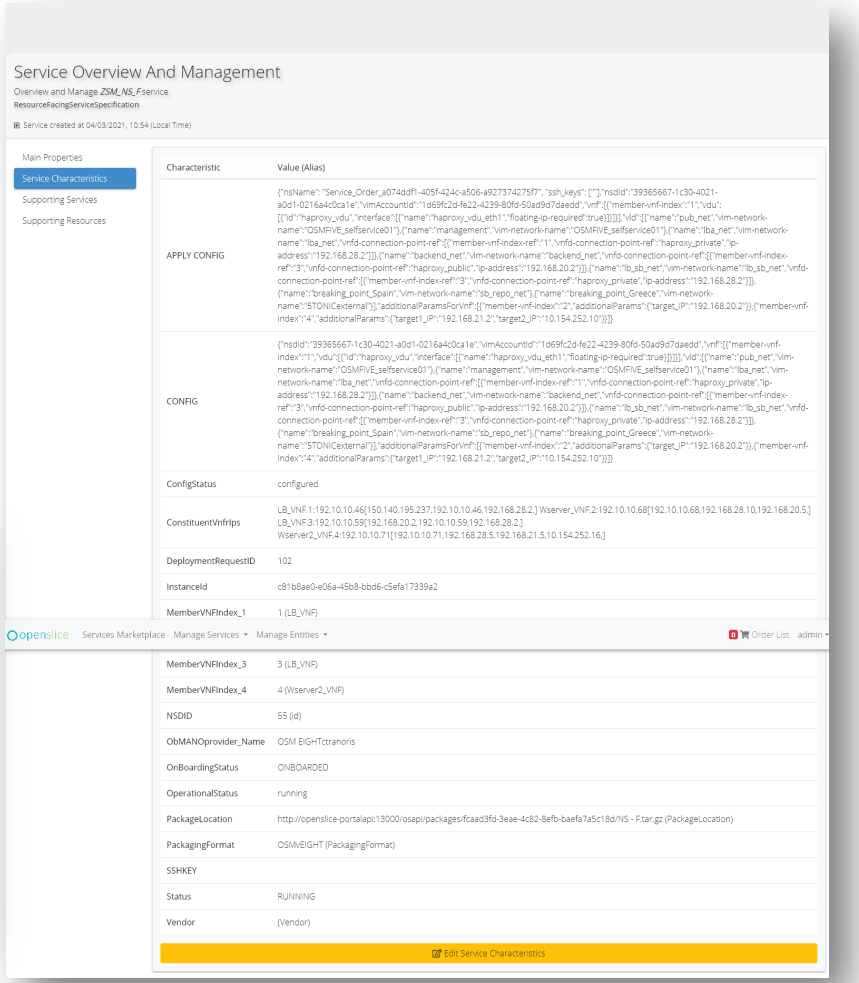
Service ordering



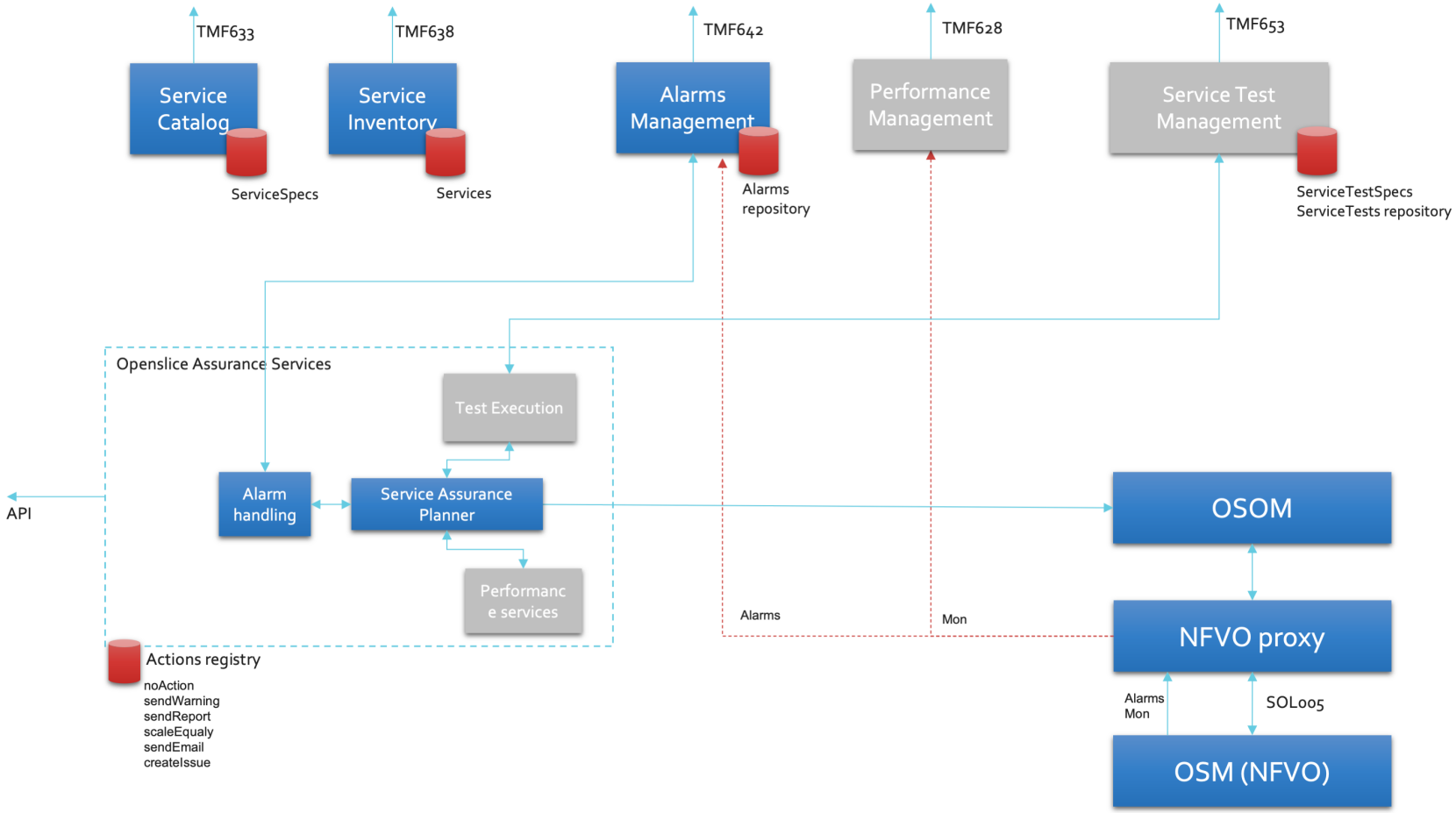
OpenSlice – service fulfillment



PoC#2 service prototypes running in OpenSlice

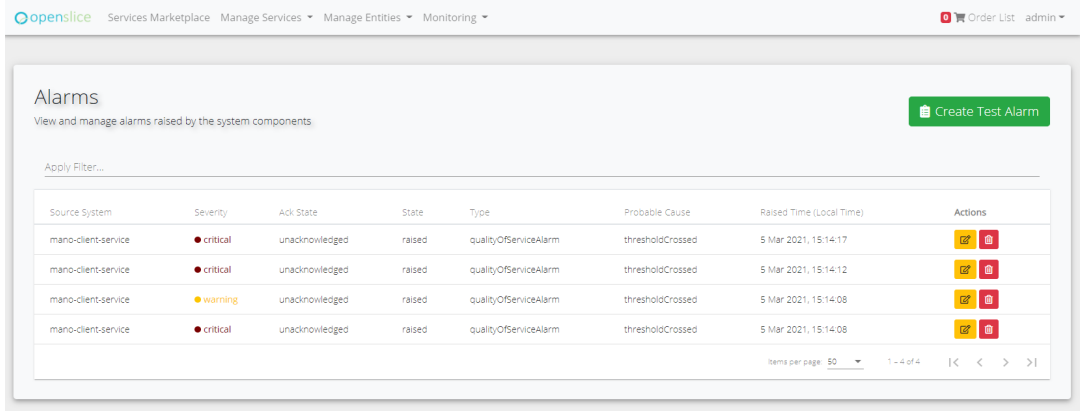


OpenSlice – service assurance

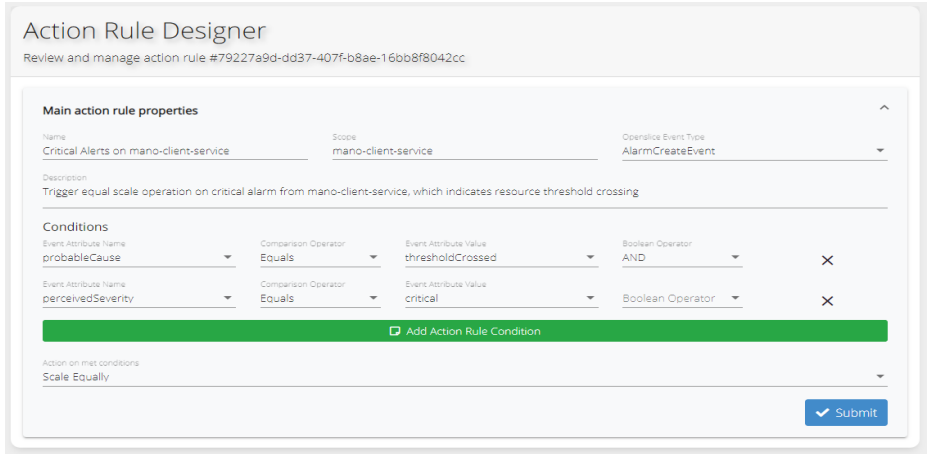


OpenSlice – service assurance

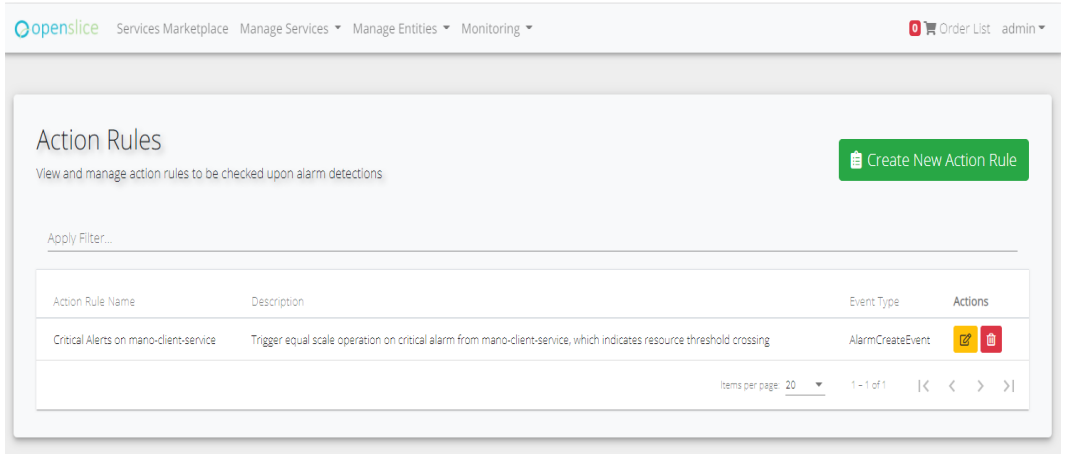
PoC#2: alarm management



PoC#2: policy design



PoC#2: policy management



PoC#2: cross-domain orchestration (steps 7-11)