ZSM ISG PoC Report Template

1 Abstract

The ZSM ISG PoC#2, entitled “Automated network slice scaling in multi-site environments”, has the aim of demonstrating the capacity to automatically scale out a deployed network slice instance across multiple administrative domains. This will be achieved using the 5G assets of 5G-VINNI, which is large-scale, end-to-end facility composed of distributed sites, each deployed at a different geographic location and defining a single administrative domain. The management and orchestration capabilities of individual facility sites, and the enablers allowing for the interworking across them, are aligned with ZSM architectural design principles.

The present document constitutes the first PoC#2 report. Its mission is to provide an overall description of the PoC, including in-scope use case and the related user story. The pre-/post-conditions and workflow detailed in the PoC user story will define the acceptance criteria to be used for PoC execution.

2 ZSM ISG PoC Report

2.1 PoC Project Scope

The ISG ZSM POC#2 fits the end-to-end (E2E) service management scenario category captured in ZSM 001, considering the network slicing features specified in ZSM003.

2.2 PoC Project Status

The table below captures the status of the PoC project, relating them to the set of milestones defined in the original PoC proposal. This report allows addressing P.P.2 “PoC user story detailed”.

PoC Milestone	Milestone description	Target Date	Additional Info	Status
P.P.1	PoC Presentation	02/12/2020	Presentation to ZSM NOC	Completed
P.S	PoC Proposal submission	15/12/2020	Official PoC proposal submission	Completed
P.P.2	PoC Public Announce	15/01/2021	Public Web announce in 5G-VINNI media (web, twitter, etc.). *Once it is approved.	Completed
P.PU	PoC user story detailed	22/01/2021	Detailing use case, specifying actors, pre-conditions & post-conditions and exceptions.	Completed
P.PT*	PoC Test Plan	~~03/03/2021~~ 17/03/2021	Testbed setup and running	Pending
P.D1*	PoC Demo	~~17/03/2021~~ 01/04/2021	Demo for showcasing at ETSI endorsed Webinar	Pending
P.C1*	PoC Expected Contribution	~~17/03/2021~~ 01/04/2021	Propose contributions to several topics at ZSM meeting (ZSM-Interim#08-e).	Pending
P.R	PoC Report	01/04/2021	PoC-Project-End Feedback	Pending
P.E	PoC Project End	01/04/2021		Pending

(*) The target dates for milestones P.PT, P.D1 and P.C1 have been delayed to weeks with respect to the original Poc proposal.

2.3 PoC Project participants (TEF)

The table below provides an up-to-date glance at the PoC Team. The only change from the original PoC proposal is the replacement of a member at the UC3M side.

	Organisation name	ISG ZSM participant (yes/no)	Contact (Email)	PoC Point of Contact (*)	Role
1	Telefónica S.A.	Yes	Jose Ordonez-Lucena joseantonio.ordonezlucena@telefonica.com Diego R. López diego.r.lopez@telefonica.com	X	Network/ service provider
2	Telenor ASA	Yes	Min Xie min.xie@telenor.com		Network/ service provider
3	Universidad Carlos III (UC3M)	No	Carmen Guerrero carmen.guerrero@uc3m.es Iván Vidal ivan.vidal@uc3m.es Borja Nogales bdorado@pa.uc3m.es		University / Supplier
4	University of Patras (UoP)	No	Spyros Denazis sdena@upatras.gr Dimitris Giannopoulos dimit.giannopoulos@upnet.gr Panagiotis Papaioannou papajohn@upatras.gr		University / Supplier
5	Openslice	No	Christos Tranoris tranoris@ece.upatras.gr Kostis Trantzas ktrantzas@upnet.gr		Open source project

3 PoC Technical Details

3.1 Overview

The present PoC focuses on the management of a network slice when deployed across multiple administrative domains. Specifically, PoC#2 aims at demonstrating how to automatically scale out a network slice instance in multi-site environments. The rationale of the demonstration is as follows:

- There is an existing (running) network slice instance. This instance is deployed across two different facility sites: Madrid (Spain) and Patras (Greece), each hosting a portion of the entire network slice.
  - From a functional viewpoint, the network slice consists of multiple NFV network services, each corresponding to a network slice subnet.
  - From an operational viewpoint, the network slice is deployed as a multi-site network slice instance.
  - From a network viewpoint, there exists L3 connectivity between Madrid and Patras, so that in-slice connectivity can be ensured along the entire data path.
- The behavior of existing (running) network slice instance is continuously monitored
  - Policy-based performance management on individual facility sites
  - There are pre-defined policy rules that allow triggering the need for scaling out operation based on collected metrics.
- When certain policy rules are met at Madrid facility site, a scaling out operation is triggered. This operation applies to the entire network slice instance.
  - This means that although the scaling out operationally is triggered at Madrid facility site, this operation needs to be propagated to Patras facility site accordingly.

- Consistency is a must: increasing capacity of one network slice subnet on one facility site requires modifying the capacity of the network slice subnet accordingly.

According to this rationale, the PoC#2 requires a use case that justifies (i) having a multi-site network slice instance; (ii) the triggering of scaling out operation at Madrid facility site, and (iii) the need to propagate the scaling out operation to the Patras facility site. The selected use case is based on vertical industry related (e.g. e-Health, PPDR) NetApps hackathon involving developers from Spain and Greece. For this short-lived event, a network slice instance is deployed.

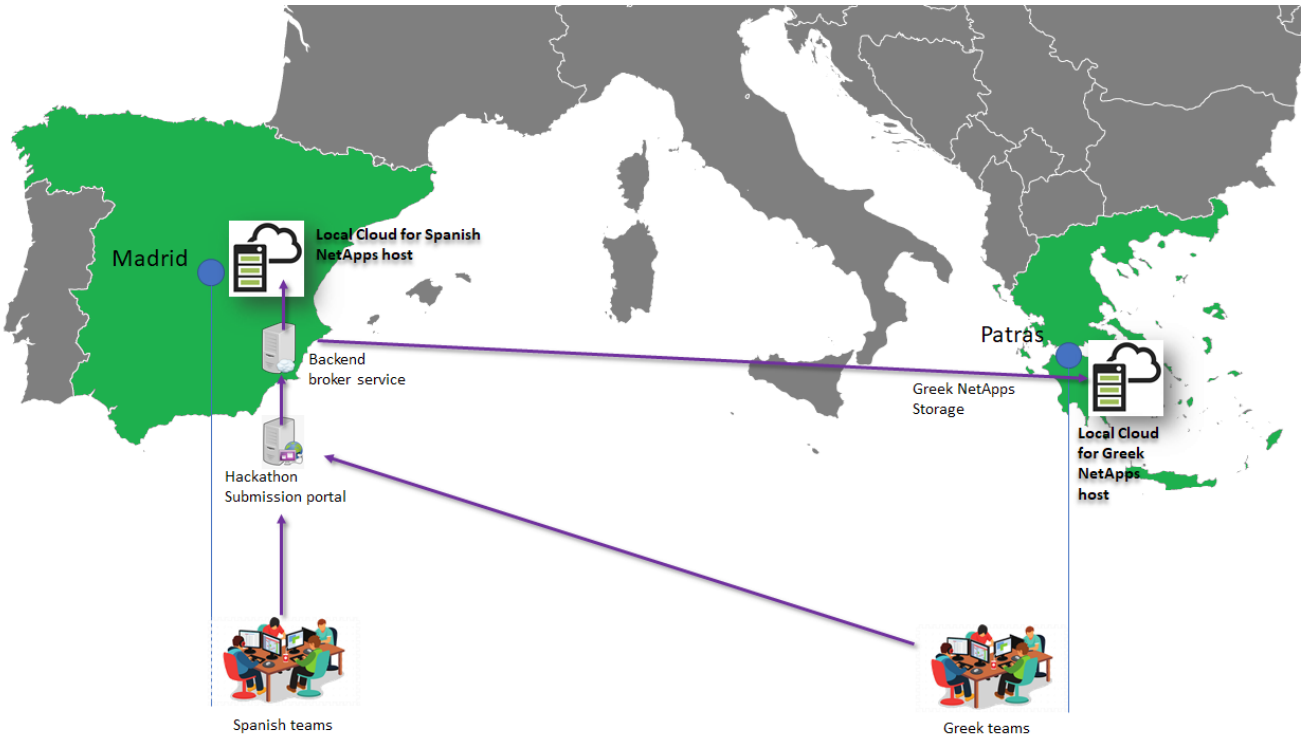

Figure 1: In-scope use case for PoC#2

The logic of the in-scope use case, illustrated in Figure 1, is as follows:

- There is a NetApp submission service where developers continuously upload their solutions. The NetApps submission portal and the backend service broker are hosted in Madrid facility site.
- Due to EU defined General Data Protection Regulation (GDPR) policy, NetApps binaries and data must be hosted in the home country. Therefore, the services for managing the NetApps catalogue repositories need to be located at both Madrid and Patras facility sites.
- During the hackathon days, there is a sudden high demand of portal interaction, due to an unexpected prize to winner developers. The demand is first detected in Madrid, thus the backend hosts of NetApps catalogue repository will be automatically scaled there.
- The scaling out operation triggered in Madrid is propagated to the Patras facility site, since this sudden high demand of portal interaction is also expected at Greece side. Unlike Madrid, where the scaling out was a reactive corrective action, the scaling out operation triggered at Patras facility site is a pro-active corrective action (due to forecasting reasons).

3.2 Resources

The present PoC may leverage the resources provided by 5G-VINNI. 5G Verticals INNovation Infrastructure (5G-VINNI) is a large-scale, end-to-end facility providing advanced 5G capabilities that can be accessed and used by vertical industries for use

case trialling [1]. The mission of 5G-VINNI is to provide a realistic 5G test and experimentation environment that is open to verticals and which supports rapid and agile testing of real-world use cases. To that end, 5G-VINNI facility leverages the use of Network Slice as Service (NSaaS) model. This means that facility provides every vertical with an isolated service experimentation platform, deployed in the form of a slice. Each vertical will then use the provided slice to set up one or more use cases, assessing their KPIs under different load conditions through the execution of test campaigns.

3.2.1 Infrastructure resources

The 5G-VINNI facility is composed of several interworking sites, each deployed at a different geographic location and defining a single administrative domain. As shown in Figure 2, these include four main facility sites (Oslo/Norway, Suffolk/UK, Madrid/Spain and Patras/Greece) and three experimentation facility sites (Aveiro/Portugal, Munich/Germany, Berlin/Germany). In addition, there is a mobile experimentation facility site in the form of a rapid response vehicle for PPDR use cases.

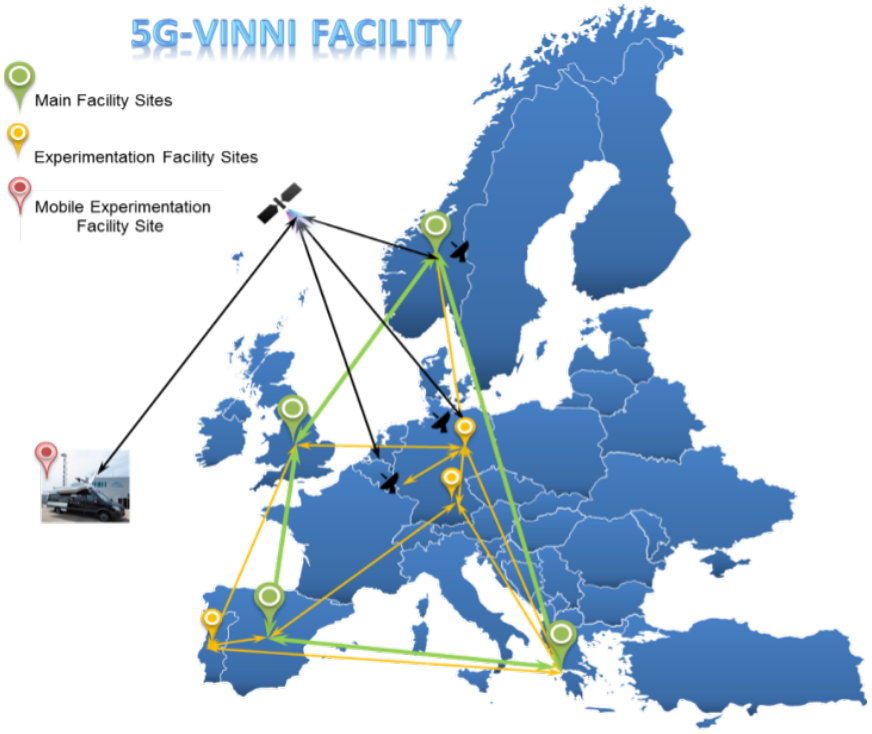

Figure 2: Madrid and Patras facility sites in 5G-VINNI facility infrastructure

For the present PoC, Madrid and Patras facility sites will be selected.

The 5G-VINNI facility in Madrid will be part of the 5TONIC lab (https://www.5tonic.org/). 5TONIC provides an open innovation hub for research in technologies, and their validation by equipment vendors, network service providers and vertical/digital service providers. For more information on the 5G capabilities of Madrid facility site, see [2].

The 5G-VINNI facility in Patras will be part of the Patras Platform for Experimentation (http://nam.ece.upatras.gr/ppe/), which is an exemplary Open Source 5G-IoT hub. This means that most of the installed components are offered as open source, together with dedicated components and services to support 5G-IoT scenarios. For more information on the 5G capabilities of Patras facility sites, see [3].

3.2.2 Management and orchestration resources

Individual 5G-VINNI facility sites have their own management and orchestration stack. This stack includes the following functional components:

- A Service Orchestrator, taking care of the lifecycle management of provided network slices at the application layer, i.e., network slice semantics.
- An NFV Orchestrator (NFVO), which deploys and operates provided networks slices at the virtualized resource layer.
- A Virtualized Infrastructure Manager (VIM), which manages the virtual deployment units building up the cloud execution environment.

The Madrid and Patras facility sites make use of the same stack, consisting of OpenSlice (Service Orchestrator) + OSM (NFVO) + OpenStack (VIM).

On the one hand, OSM [4] is an E2E Network Service Orchestrator aligned with ETSI NFV specification. It is an ETSI-hosted initiative, which is currently formed by more than 120 contributors, that aims to develop a solution that facilities the use and maturation of NFV technologies, gives access to a huge ecosystem of VNF vendors, and allows testing and monitoring between the orchestrator and the rest of the elements (NFVI, VIM, VNFs, PNFs). As captured in Figure 3 , OSM:

- Exposes management capabilities to external management systems (e.g., legacy OSS/BSS, Service Orchestrator) through a unified North Bound Interface (NBI). This NBI provides a superset of ETSI NFV SOL005 APIs together with the ability to handle network slice from a resource management viewpoint (i.e., the ability to handle network slice instances as a concatenation of network slice subnet instances, each deployed as an exclusive or shared network service).
- Interacts with underlying infrastructure resources through the OSM's South Bound Interface (SBI). This SBI includes plugins towards virtual and transport domains (e.g., VIM, WIM and SDN-C plugins) as well as configuration interfaces towards individual physical (and hybrid) network functions.

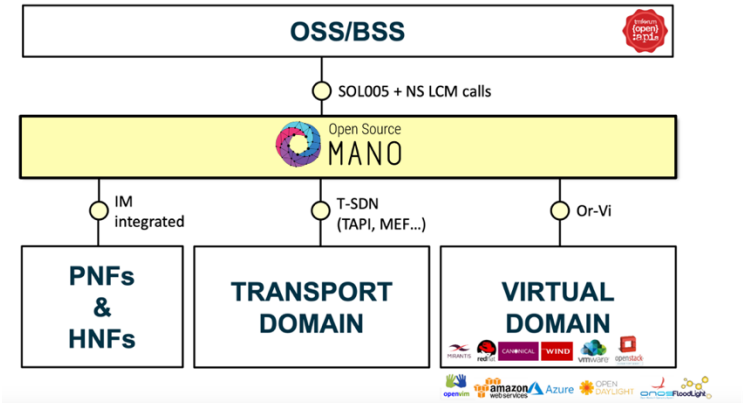

Figure 3: OSM and its role of Network Service Orchestration

Figure 4 illustrates the components building up the OSM software architecture. The different components are designed according to the architectural principles of modularity (ZSM principle 01) and resiliency (ZSM principle 07), and are based on a model-driven approach, with open interfaces (ZSM principle 04). As shown in the figure, individual components interacting with each other through a Kafka bus, which provides integration fabric functionality to allow for extensibility (ZSM principle 02) and scalability (ZSM principle 03) in OSM architecture.

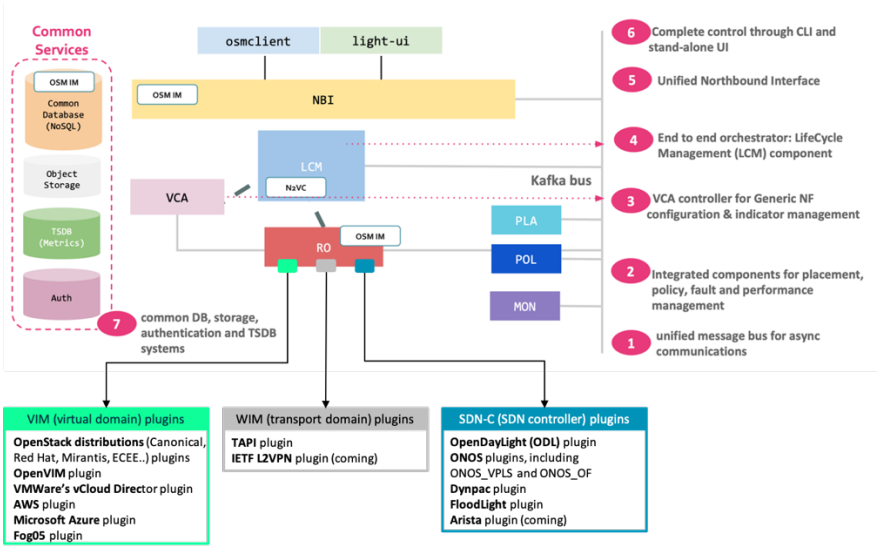

Figure 4: OSM Release EIGHT architecture

On the other hand, OpenSlice [5] is an open-source, operations support system (OSS) solution providing Service Orchestration functionality, including both service fulfilment and assurance lifecycle phase. From a customer-facing viewpoint, OpenSlice defines a user-friendly web portal that allows managing (e.g., authorization, authentication) the interaction with vertical customers, capturing their service orders and keeping them informed about the status of the network slices, which their ordered services are hosted on. From a resource-facing viewpoint, it interacts with the NFVO, consuming SOL005 exposed capabilities to deploy and operate the virtualized components of the network slice.

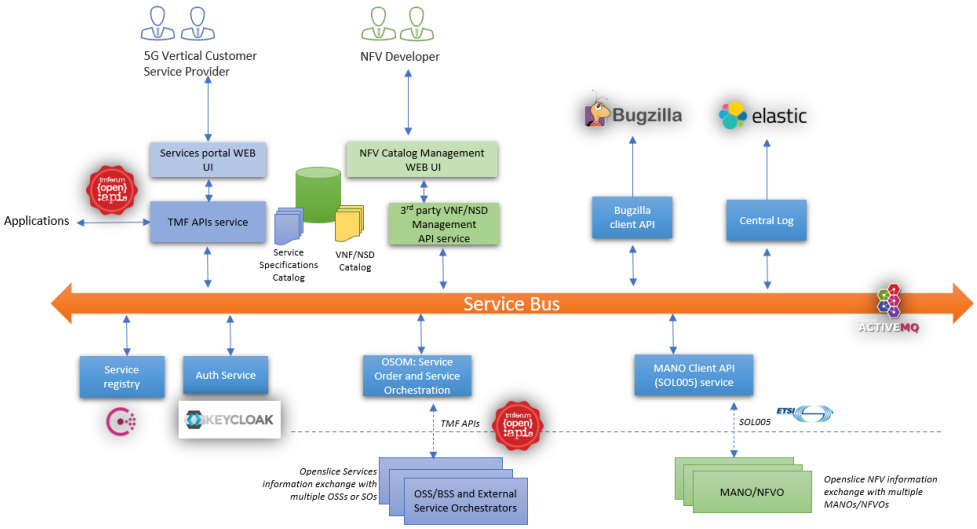

Figure 5: OpenSlice architecture

Figure 5 illustrates the architectural framework of OpenSlice. Roughly, the solution includes a set of loosely coupled modules exchanging messages via an ActiveMQ service bus, following microservice architecture. Table 1 provides a summary of the main microservices building up OpenSlice.

Table 1: OpenSlice services

µService	Description
Auth	Keycloack server to authenticate/authorize verticals using OAuth 2.0 schemes.
Service registry	One-stop solution for typical procedures in µservice architectures, including service (self) registration, discovery, key-value store and load balancing. It is implemented using Consul.
Bugzilla client API	Offers interface to Bugzilla, which is a ticketing tool that allows issue tracking (fault alarms, service orders) and reporting (to verticals, facility operators, etc.) via tickets.
Central Logging	Logs all distributed actions into an Elasticsearch cluster.
TM Forum APIs	Offers TM Forum's OpenAPIs to allow consumption of service catalogue exposed capabilities. These open APIs include Service Catalog, Ordering and Inventory APIs.
MANO client APIs	Offers SOL005 API services (e.g. NSD/VNFD on-boarding, NS instantiation/termination requests, etc).
3rd party VNFD Mgmt APIs	Offers NFV APIs to manage 3rd party VNFDs (e.g. on-boarding, updating). These APIs allow verticals to bring their own VNFs to 5G-VINNI, to validate their KPIs.
Service Order and Service Orchestration	Referred to as OSOM, captures service ordering requests triggered by verticals and propagates them to external systems, including legacy OSS/BSS or other Service Orchestrators.

According to the above rationale, the management and orchestration setup for the PoC is captured in Figure 6. As seen, both facility sites have their own stack:

- 5TONIC: Madrid hosted OpenSlice instance (hereinafter referred to as "Madrid-OpenSlice") + Madrid hosted OSM instance (hereinafter referred to as "Madrid-OSM") + Madrid hosted VIM instance (hereinafter referred to as "Madrid-VIM")
- Patras Platform for Experimentation: Patras hosted OpenSlice instance (hereinafter referred to as "Patras-OpenSlice") + Patras hosted OSM instance (hereinafter referred to as "Patras-OSM") + Patras hosted VIM instance (hereinafter referred to as "Patras-VIM")

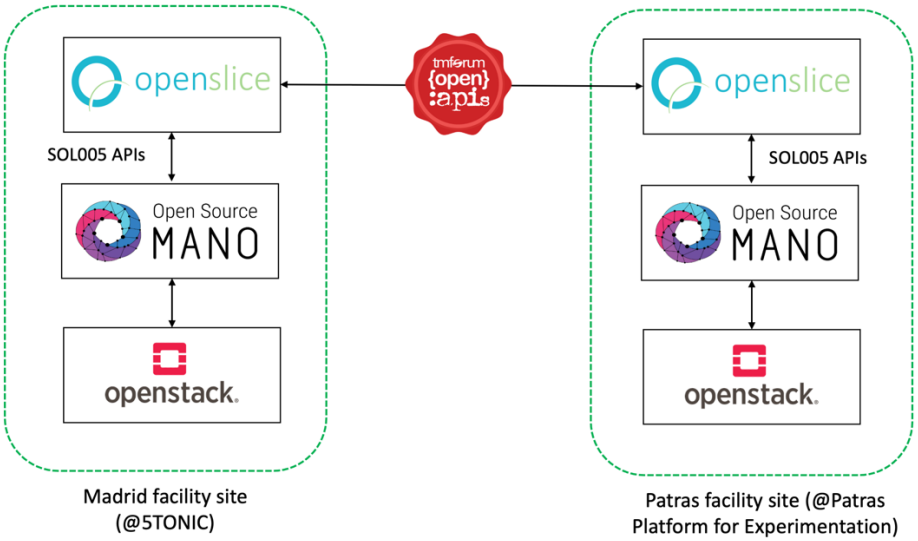

Figure 6: Management and orchestration stack used in the PoC.

To allow for multi-site network slice orchestration, interworking between both stacks is a must. In the PoC, this interworking occurs at the Service Orchestration layer, with “Madrid-OpenSlice” and “Patras-OpenSlice” communicating using TMF Forum Open APIs.

3.2.3 Network slice – design and deployment

The NetApps hackathon event captured in Figure 1 requires deploying a network slice across Madrid and Patras facility sites. This means that each facility site hosts a portion of the slice. The internal composition of this slice and the geographical distribution of their functional components is illustrated in Figure 7. As seen, the network slice consists of three network slice subnets, each modelled as a separate Network Service Descriptor (NSD).

- **Network Slice Subnet A (NSS-A),** deployed in Madrid according to NSD_F. The NSD_F is composed of four VNFs, including two Load Balancers (LB-1 and LB-2), one Web Server and one backend API brokering service.
- **Network Slice Subnet B (NSS-B),** deployed in Madrid according to NSD_{SRV}. The NSD_{SRV}, consisting of three VNFs, including one Load Balancer (LB-3), one repository catalogue and one catalogue DB.
- **Network Slice Subnet C (NSS-C),** deployed in Greece according to NSD_{SRV}. Like NSS-B, NSS-C holds the repository catalogue and its supported DB, together with the LB-3 as an entry point of requests.

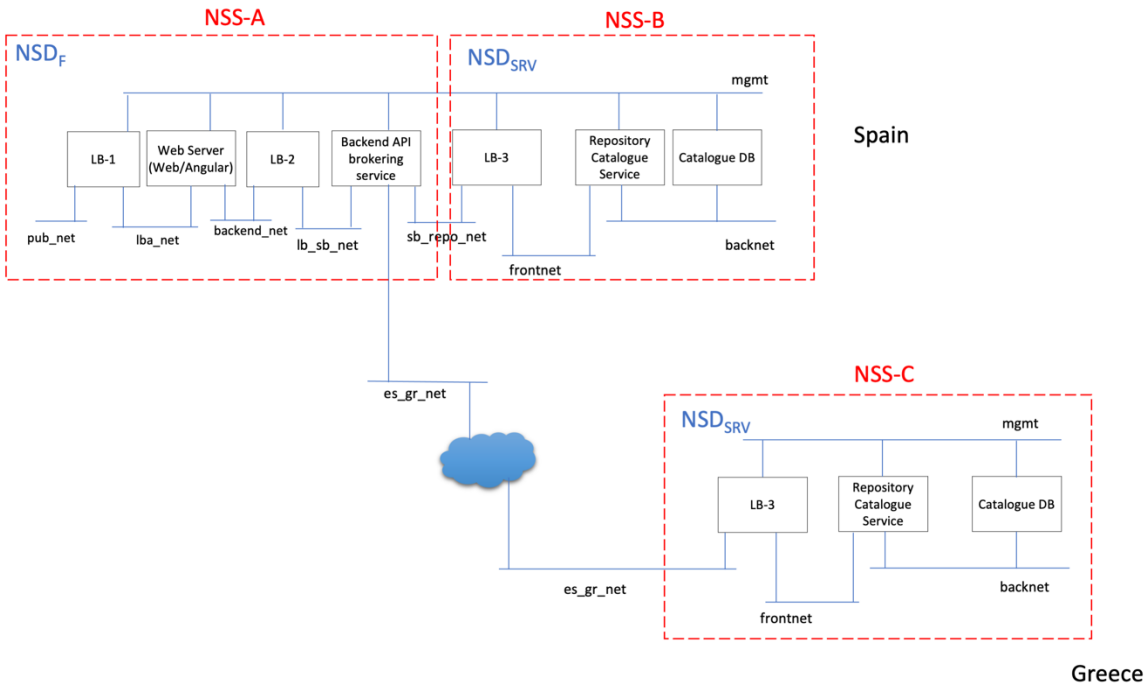

Figure 7: Network slice design

According to the network slice design, the traffic flow is as follows: the *LB-1* acts as an entry point, sending incoming traffic to the *Web Server* that serves the web pages to the developers. The web pages issue requests to the *backend API brokering service*, using *LB-2* for scalability reasons. Upon capturing these requests, the *backend API brokering service* (in NSS-A) decides the equivalent *repository catalogue service* to communicate with, either to Madrid (NSS-B) or to Patras (NSS-C). If Madrid is selected, LB-3 from NSS-B handles incoming traffic to the API endpoint of the repository catalogue service. This service is supported by a database deployed in a High-Availability (HA) cluster. If Patras is selected, it is LB-3 from NSS-C which does the process.

The Figure 8 and Figure 9 illustrates the impact of the scaling operation over the running slice instance. Figure 3 captures the slice instance as originally deployed for the NetApps hackathon, while Figure 9 shows the state of the slice instance after having been scaled out.

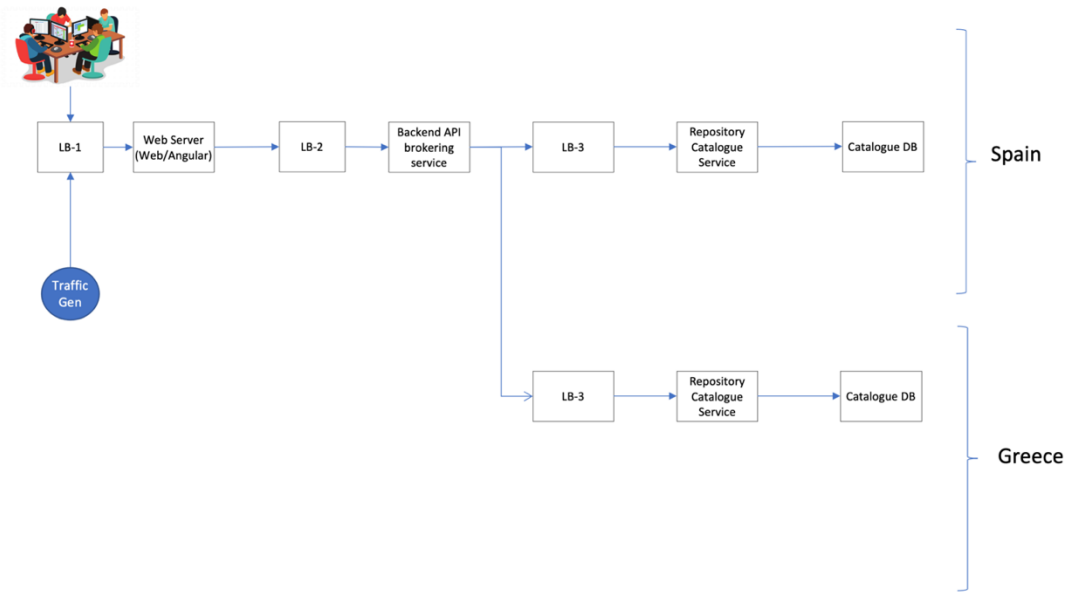

Figure 8: Network slice instance before the scaling out operation

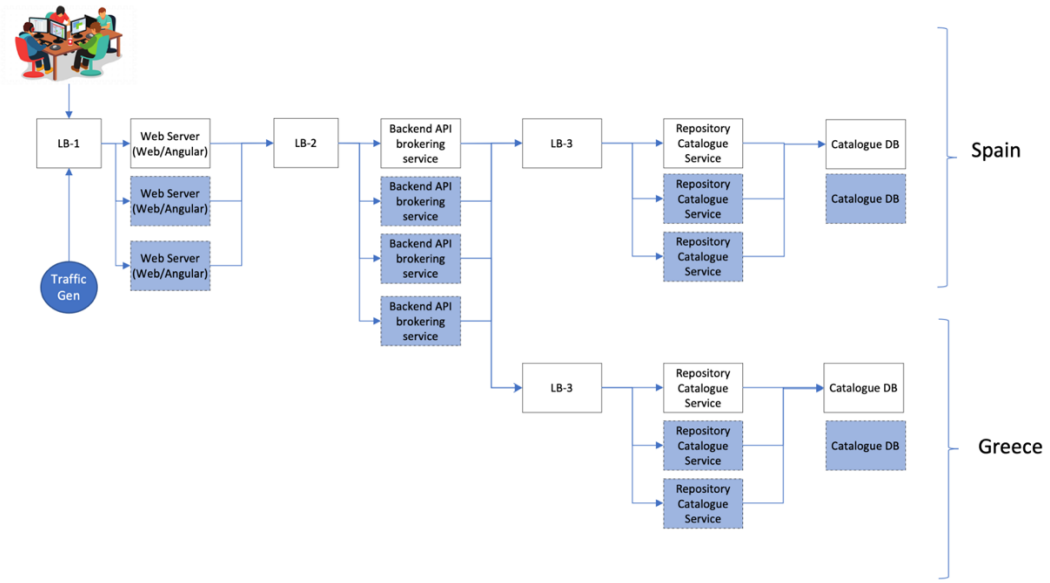

Figure 9: Network slice instance after the scaling out operation.

3.3 User story

This section details the PoC#2 user story.

3.3.1 Pre-conditions

The following conditions need to be true before the scaling out is triggered.

- An instance of OpenSlice+OSM+VIM stack is installed (and operative) in Madrid facility site.
- An instance of OpenSlice+OSM+VIM stack is installed (and operative) in Patras facility site.
- There exists L3 based connectivity between Madrid and Patras facility sites. This allows for data and control plane communication between both facility sites.
- "Madrid-OpenSlice" can communicate with "Patras-OpenSlice", using TMForum Open APIs. This allows for management plane communication between both facility sites.
- The NSDs (and constituent VNF Packages) building up the network slice are on-boarded to individual OSM instances, NSD_F is on-boarded to "Madrid-OSM", allowing for NSS-A. NSD_{SRV} is on-boarded to both OSM instances, including "Madrid-OSM" (allowing for NSS-B) and "Patras-OSM" (allowing for NSS-C).
- Service ordering is triggered by "Madrid-Openslice". For NSS-A and NSS-B, "Madrid-OpenSlice" forwards SOL005 calls to "Madrid-OSM". For NSS-C, "Madrid-Openslice" forwards TMForum OpenAPI calls to "Patras-Openslice", which in turns translates them into SOL005 calls for the local OSM instance.
- NSS-A and NSS-B are deployed on "Madrid-VIM", with day-0 and day-1 configuration on individual VNFs.
  NSS-C is deployed on "Patras-VIM", with day-0 and day-1 configuration on individual VNFs.
- Each service communicates with its sibling services successfully. The backend API brokering service can connect to both repository catalogue service in Greece and Spain.
- "Madrid-OSM" performs day-2 operations on NSS-A and NSS-B, comparing run-time performance measurements and fault alarms (collected from MON) against PM/FM policies (defined in POL).
- "Patras-OSM" performs day-2 operations on NSS-C, comparing run-time performance measurements and fault alarms (collected from MON) against PM/FM policies (defined in POL).

3.3.2 Steps/workflow

1. There is a sudden high demand of portal interaction in Madrid facility site (see Section 3.1). To model this, a traffic generator sends quite a high number of HTTP requests to the web server.
2. The HTTP requests represent a traffic load surge with a 3:1 ratio, which results in higher I/O processing and CPU usage of NSS-A VNFs. This scenario collapses backend API brokering service, which is able to forward traffic to either NSS-B or NSS-C.
3. "Madrid-OSM" detects this load surge as part of day-2 activities, by comparing the MON collected metrics against the POL defined policies. Based on this comparison, it triggers a scaling out operation on NSS-A.
4. "Madrid-OSM" scales NSS-A out, by
    a) creating two additional Web server instances and three additional instances of the backend API brokering service (see Figure 9). The new VNF instances are deployed on the local VIM, and day-0 and day-1 configuration are completed.
    b) Reconfiguring LB-1, so that the incoming traffic can also be forwarded to the newly instantiated Web Servers.
    c) Reconfiguring LB-2, so that the web page requests can also be forwarded to the newly instantiated backend API brokering services.

   NOTE: After NSS-A scaling, the backend API brokering service works again.
5. The load surge originated in step 1 makes backend API brokering service send traffic to NSS-B through LB-3. The increased traffic collapses repository catalogue service together with its supported DB.
6. "Madrid-OSM" detects this load surge as part of day-2 activities, by comparing the MON collected metrics against the POL defined policies. Based on this comparison, it triggers a scaling out operation on NSS-B.
7. "Madrid-OSM" scales NSS-B out, by:
    a) creating two additional repository catalogue instances and one additional DB instance (see Figure 9). The new VNF instances are deployed on the local VIM, and day-0 and day-1 configuration are completed.
    b) Reconfiguring LB-3, so that the incoming traffic can also be forwarded to the new repository catalogue instances.
8. "Madrid-OSM" notifies the local OpenSlice instance of both scaling operations (steps 4 and 7).

9. OpenSlice detects the scale change in repository catalogue service in Madrid, based on step 8 notification. Since OpenSlice is aware of the use case semantics, it decides a similar scenario may occur in Patras. Therefore, it takes the decision that NSS-C needs to be also scaled out as NSS-B did, in order to avoid collapse as in Madrid.
10. "Madrid-OpenSlice" propagates the scale change to the "Patras-OpenSlice", using TMF Open API(s).
11. "Patras-Openslice" checks this NSS-C scaling out request, and forwards it to the "Patras-OSM".
12. "Patras-OSM" scales out NSS-C out, by:
    a) creating two additional repository catalogue instances and one additional DB instance (see Figure 9). The new VNF instances are deployed on the local VIM, and day-0 and day-1 configuration are completed.
    b) reconfiguring LB-3, so that the incoming traffic can also be forwarded to the new repository catalogue instances.
13. "Patras-OSM" notifies the local OpenSlice instance of this scaling operation.
14. "Patras-OpenSlice" informs "Madrid-OpenSlice" of the successful NSS-C scaling out operation, in response of step 10 request.

3.3.3 Post-conditions

The following conditions need to be true after the completion of the scaling out operation:

- The NSS-A and NSS-B have been successfully scaled out at Madrid facility site.
- The NSS-C has been successfully scaled out at Patras facility site.
- The day-2 metrics (run-time performance measurements and fault alarms) are as good as were before the load surge.

4 References

[1] 5G-VINNI project [Online]. Available: https://www.5g-vinni.eu/
[2] 5G-VINNI - Spain Main Facility Site. Available: https://www.5g-vinni.eu/spain-main-facility-site/
[3] 5G-VINNI - Greece Main Facility Site. Available: https://www.5g-vinni.eu/greece-main-facility-site/
[4] ETSI Open Source MANO (OSM) [Online]. Available: https://osm.etsi.org
[5] OpenSlice project [Online]. Available: https://openslice.readthedocs.io/en/stable/