PoC#9 Final Report

1 General

This document serves as report for the PoC#9, entitled Autonomous Network Slice Management for 5G Vertical Services-

Submission of this ENI ISG PoC Report as a contribution to the ENI ISG does not imply any endorsement by the ENI ISG of the contents of this report, or of any aspect of the PoC activity to which it refers.

2 ENI ISG PoC Report

2.1 PoC Project Completion Status

The PoC has been **completed** with some minor deviation with respect to the original plan, mainly due to the COVID-19 pandemic. The planned showcase with real public has been substituted with a remote integration of the different pieces of software that compose the system and their functional and performance validation with remote testing sessions.

2.2 ENI PoC Project Participants

These are the details of the PoC:

- PoC Project Name: Autonomous Network Slice Management for 5G Vertical Services
- Network Operator/Service Provider:TIM _____ Contact: Luca Pesando
- Manufacturer A: Samsung _____ Contact: Yue Wang
- Manufacturer B: WINGS _____ Contact: Vera Stavroulaki
- Additional Member: Nextworks _____ Contact: Gino Carrozzo
- Additional Member: University Carlos III of Madrid Contact: Marco Gramaglia

2.3 Confirmation of PoC Event Occurrence

The PoC has been demonstrated with a network service scenario deployed on the network infrastructure provided by Nextworks, one of the partners of the PoC. The chosen service is a Content Delivery Network (CDN) service, deployed in eMBB network slice. The CDN VNFs and Network Service descriptors, with the corresponding Network Slice Templates and Vertical Service Blueprints have been onboarded the Management and Orchestration Framework depicted in Figure 1 below. The on-demand deployment and the ENI-driven automation of the lifecycle of vertical services, network slices and NFV network services are handled through the combined actions of the Service & Slice Manager and the NFVO. The system has been developed in a modular manner, to demonstrate the applicability of the ENI concepts in different deployments. In particular, the PoC workflow has been demonstrated using two opensource NFV orchestrators: TIMEO[1] (as shown in the picture) and OSM[2]. A public remote demonstrator "Vertical's intent evolution at service runtime driving vCDN automated scaling", based on the OSM-based deployment, has been held during the OSM Ecosystem Day, on March 2021.

[1] https://github.com/nextworks-it/timeo
[2] https://osm.etsi.org/

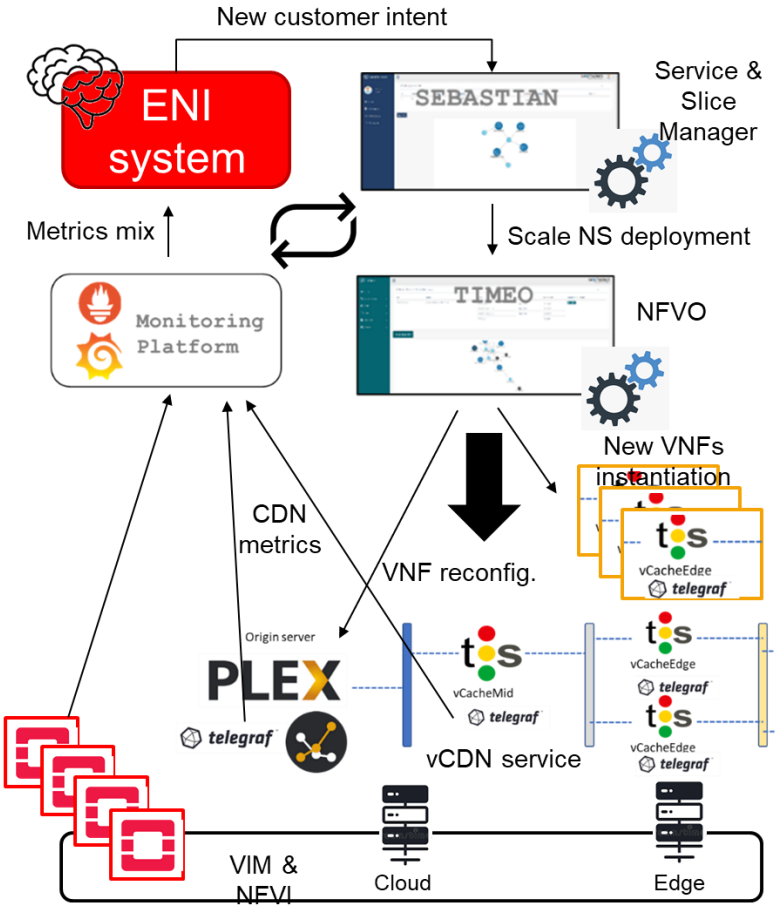

Figure 1: The Overall PoC#9 Software Infrastructure

The Management and Orchestration System offers a web-based portal and an API that is used for different purposes:

- Onboarding the network services in an automatic manner
- Getting the metrics of the network service, that are used by the ENI system (through the monitoring platform)
- Offer the possibility to the vertical to update the configuration of the Artificial Intelligence algorithm, to provide further configuration possibility to the vertical who is using the service provider.

The service is onboarded using the API provided by the SEBASTIAN Service and Slice Manager. This software is available Open Source [3] and is being developed in the context of the 5G-EVE[4] and 5GROWTH[5] projects. The

[3] https://github.com/nextworks-it/slicer
[4] https://www.5g-eve.eu/
[5] https://5growth.eu/

onboarded service is composed by a Content Delivery Network VNF instances, whose lifecycle is ultimately handled by the NFV Orchestrator (NFV-O), i.e., TIMEO or OSM.

```
{
    "vsd":{
        "name":"VSD_TEST_SMALL",
        "version":"0.2",
        "vsBlueprintId":"9",
        "sst":"EMBB",
        "managementType":"PROVIDER_MANAGED",
        "qosParameters":{
            "users":"50"
        }
    },
    "tenantId":"nextworks",
    "isPublic":true
}
```

Figure 2: Virtual Service Descriptor of the eMBB CDN Service

Vertical Service and Network Slice Template Onboarding

During the onboarding phase, the service provider compiles the definition of the service required by the vertical and the characteristics of the related network slices and NFV network services. The resulting network slice templates and vertical service blueprints are onboarded in the form of JSON files in the Service and Slice Manager through the web GUI.

Starting from the blueprint, the vertical specifies the application-level requirements for the desired services, providing some high level intents such as the network slice type (i.e., eMBB in this case) or the expected number of users for the service. This information is processed at the Service and Slice Manager, which generates the virtual service descriptor corresponding to the declared intent. The resulting JSON is depicted in Figure 2.

Vertical Service Provisioning

The provisioning phase involves the steps to create the network slice and instantiate the network service required to run the vertical service. From the Service and Slice Manager web GUI, the vertical can request the instantiation of a vertical service selecting the ID of its descriptor and providing the configuration parameters of the CDN application. This request may be also sent using the REST API of the system, with the JSON depicted in Figure 5. Internally, the Service and Slice Manager maps the vertical intents into a concrete definition of the network slice and triggers the instantiation of the network service on the NFV Orchestrator, with the deployment of the needed VNFs and the setting of important configuration parameters such as the IP address of the instance and the running operation port. Once these procedures are successfully accomplished, the overall MANO system is reporting that the service is correctly instantiated and ready to run (including the overall ENI system, as discussed next and showcased in Figure 3 and in Figure 4).

Figure 3: Vertical Service instantiation on the SEBASTIAN platform (Service and Slice Manager)

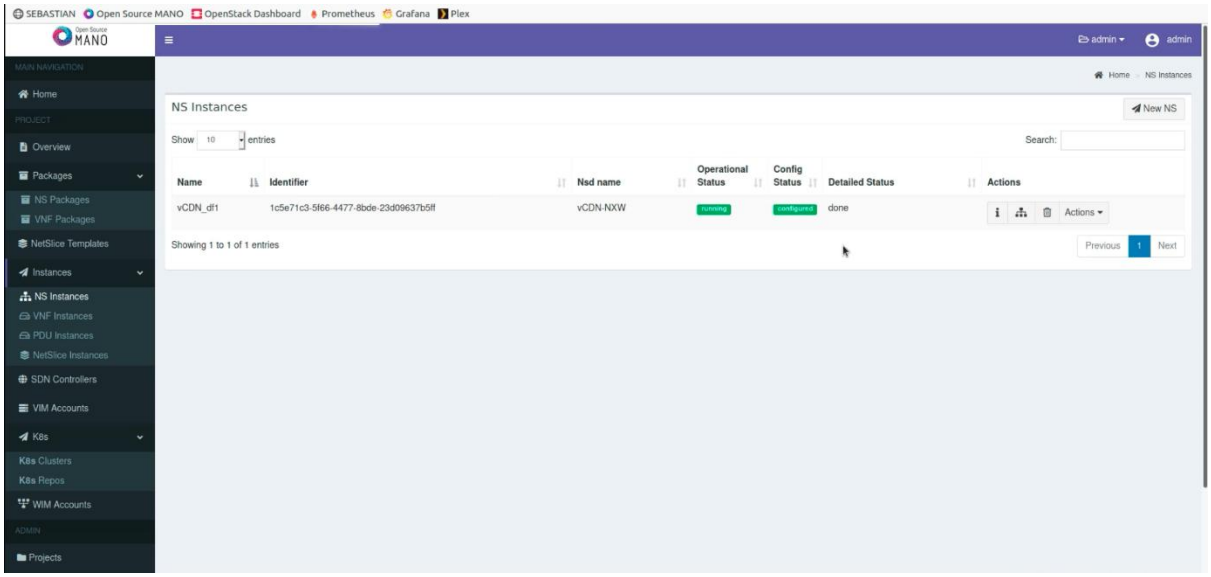

Figure 4: Network Service instantiation on the OSM platform (NFV Orchestrator)

```
{
    "name":"CDN_small",
    "description":"Bluespace CDN service",
    "vsdId":"25",
    "tenantId":"nextworks",
    "userData": {
        "uservnf.origin_address": "10.30.8.78",
        "uservnf.origin_fqdn": "bluespace.lab",
        "uservnf.origin_port": "32400",
        "uservnf.rm_dns_hosts": "true",
        "uservnf.add_dns_hosts": "true"
    }
}
```

Figure 5: Instantiation request for a vertical service

Network Service Monitoring

In order to take decisions such as the one envisioned by this PoC (e.g., scaling decisions when the target QoS is not met) the monitoring framework is fundamental. For the purposes of our PoC we utilize the CPU consumption metric as the main driver for the scaling decisions. Being the proof of concept not deployed in a production system, we emulate the system load by offering load according to well defined pattern. Specifically, we adopt the methodology provided in [6] to generate real word demands for a given service. In this case, we selected the Youtube service, given the similarities with the CDN VNFs running in the eMBB netwok slice used for this PoC. Hence, we created a simulated load that provides a proportional number of requests to the load recorded in the same work.

The monitoring framework included in the overall system is depicted in Figure 6 below, which depicts the overall location of the different probes and the different functionalities, which are further discussed next.

ENI Assisted system

The set of ENI-assisted functionality are essentially two:

Intent translation from verticals: this task helps the vertical service providers (a video content provider in this case) to operate the services without deep knowledge of the network internals. Besides the initial, high level, input towards the orchestration included in the JSON files, verticals may want to be included in the loop to have a finer control level without going too much into the details of the specific implementation, as specified by the other template listed here. As discussed also in the following, we provide a “knob” to the vertical that can be used to steer the autonomous operation in an understandable way.

Scaling automation: by forecasting the future load of the network, the ENI system can provide scaling suggestion to the assisted system (the slice management and orchestration one in this case). After the initial trial period, however, we detected that the timings related to the data forecasting were too fast to be coupled with the

[6] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs and Z. Smoreda, "Identifying Common Periodicities in Mobile Service Demands with Spectral Analysis," 2020 Mediterranean Communication and Computer Networking Conference (MedComNet), Arona, Italy, 2020, pp. 1-8, doi: 10.1109/MedComNet49392.2020.9191477.

orchestration system. Thus, we designed a two-tiers monitoring and forecasting system that 1) performs a long term forecasting, and 2) continuously re-evaluate the decision at short term, to correct the long term one only if needed. By doing this we achieve a scalable operation of the system, by avoiding too frequent re-orchestrations of the resources, which have an inherent cost.

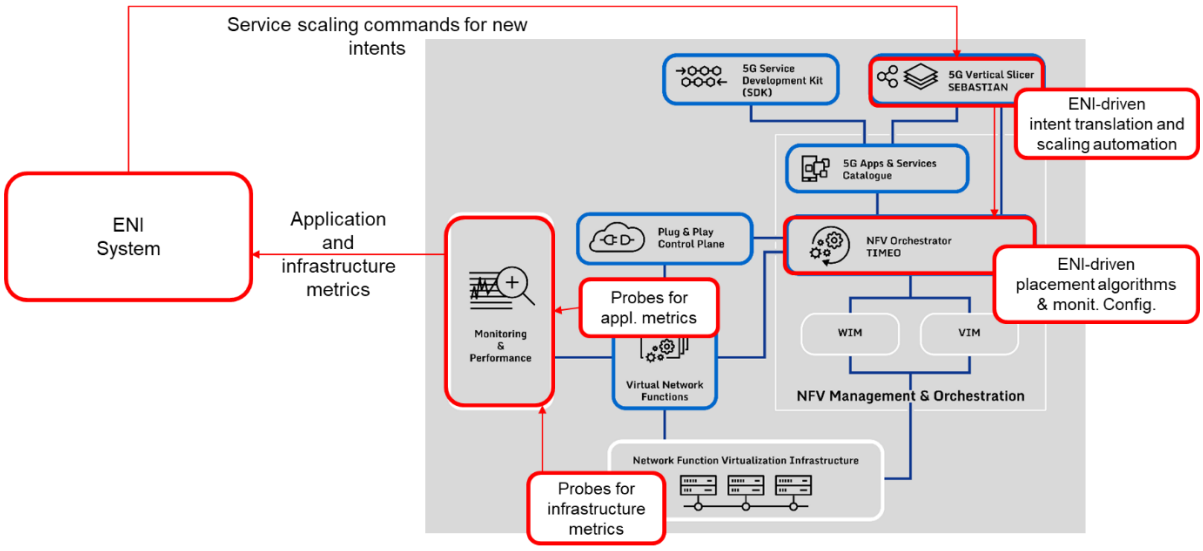

Figure 6: The overall monitoring framework in the PoC architecture

Network Automation Algorithm

For this PoC, we employ the algorithm described in [7], adapting it to the characteristics of this problem. The overall structure of the machine learning algorithm is depicted in the Figure 7 below.

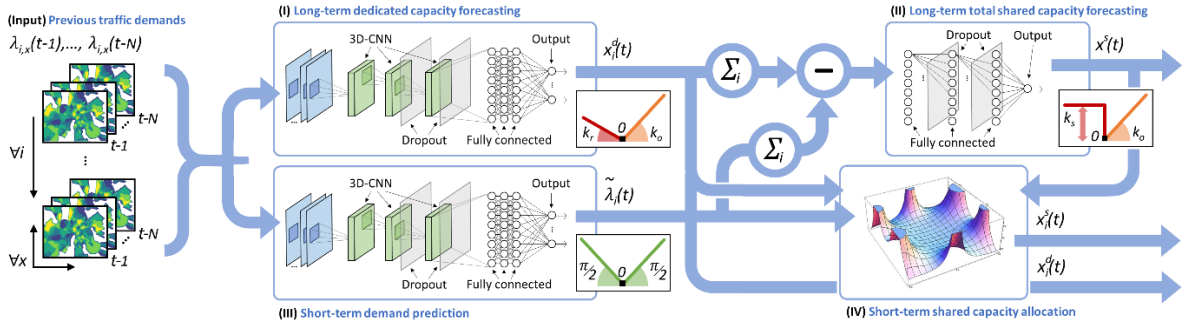

Figure 7: The architecture of the Machine Learning algorithm taken from [6]

The structure stems from the previous load of the system, gathered through the monitoring platform discussed before. Among the plethora of statistics that could be monitored with the Prometheus platform, we selected the CPU consumption, monitored at 5 seconds granularity. This is used as the input block that feeds the blocks I and III that provide the long term and short term capacity forecasting used afterwards. This information is merged into the blocks III and IV that perform the short term allocation of the resources using a customized loss function, that is needed to interpret the vertical needs with respect to resource allocation. As exemplified in Figure 8, we employ a customized function that includes the cost incurred by the vertical in terms of resource consumption.

[7] D. Bega, M. Gramaglia, M. Fiore, A. Banchs and X. Costa-Perez, "AZTEC: Anticipatory Capacity Allocation for Zero-Touch Network Slicing," IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, Toronto, ON, Canada, 2020, pp. 794-803, doi: 10.1109/INFOCOM41043.2020.9155299.

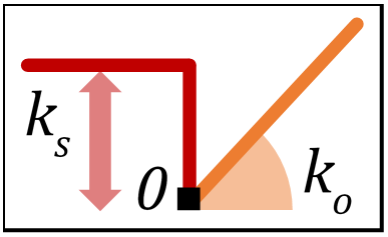

Figure 8: The cost function used by the PoC

The cost function is modelled by two variables K_S and K_O, which embody the cost for the SLA violations of the system and the cost of overprovisioning, respectively. That is, if the prediction of the system is below the real value (the perfect prediction is marked as 0 in Figure 8), we incur into a cost equal to K_s, while positive errors means that more resources are allocated in the system, which have a proportional cost increase in the system. We summarize these two variables into one $\alpha = K_S / K_O$, that catches the relation between the "aggressiveness" of the system that can be configured by the vertical: higher values of α will push the operation of the system towards "more expensive" prediction and scaling decisions, as depicted in the next Figure 9.

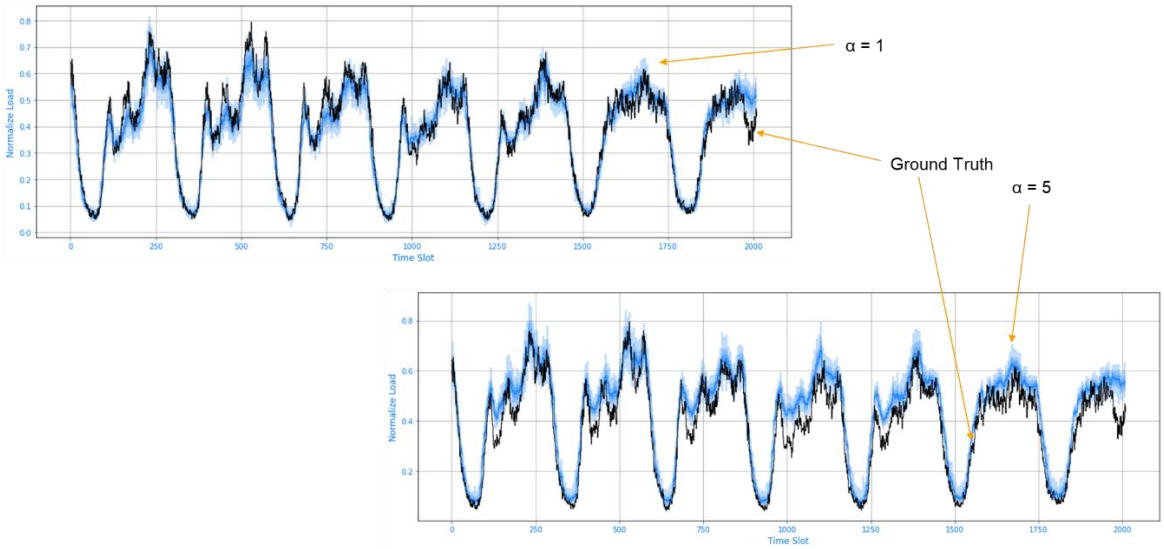

Figure 9: Load Prediction with different values of α

From the figure, we can see that higher α values achieve a more resilient resource assignment (scaling is performed on thresholds) at the price of a higher resource provisioning.

In order to achieve a statistically significant variability for the short term prediction reconsideration of previously taken actions we also employ a measure for uncertainty, depicted by the shades of blue in Figure 9. The overall neural network structure is available as Open Source at [8].

2.4 PoC Goals Status Report

The PoC defined three main goals that have been achieved according to the following list.

- PoC Project Goal #1: Translation between intent-based Vertical Service definition and resource-based descriptor of the end-to-end 5G network slice.

  Goal Status: Met, the intent of the Vertical is captured through the configuration parameter α of the deep learning function, which is exposed via an interface to the vertical.

- PoC Project Goal #2: Enhanced strategies for sharing and composition of network slices.

  Goal Status: Met, the slice is dynamically and automatically re-composed with multiple CDN caches.

[8] https://github.com/wnluc3m/AZTEC

- PoC Project Goal #3: Automation of scaling and migration procedures for self-re-optimization of the global set of network slices.

Goal Status: Met, fully validated with scaling techniques implemented on the VNFs composing the network slice, automatically scaled on the basis of the real-time performances and migrating the traffic on the most suitable CDN cache.

2.5 PoC Feedback Received from Third Parties (Optional)

We did not receive any particular feedback from third parties.

3 ENI PoC Technical Report (Optional)

3.1 General

The work performed during this PoC served to identify some potential gaps in the ENI standardization and, in general, the work performed by researchers in the fields. We summarize them next. Note that we removed all the subsections that were not fully relevant

3.3 Gaps identified in ENI standardization

We identified some gaps in the ENI standardization related to the two main technical innovation of the PoC, as discussed next:

Integration of verticals in the architecture: based on the experience of our work in European Projects, we found a very high interests from verticals in being involved in the network operation in a mild way. That is, there should be a closed loop between network operator and the party that is providing the service, avoiding the traditional interactions from OSS/BSS that are hardly acceptable in nowadays service dynamics but, at the same time, service providers do not need to go into the details of e.g., how much memory assign to a VM, the IP addressing or other low level aspects. Hence, vertical shall be in the loop with some high level parameters, such as the "knob" we offer in this PoC.

Dealing with different timescales: Depending on the resources that have to be assigned different timescales may be included in the decision. Taking decisions in a system has a cost, and if they are taken too frequently there may be an indirect cost or, simply, they cannot be taken because of the physical limitations of the system. Thus, interfaces or procedures that take into account this aspect shall be defined, such as re-considerations at shorter timescales of the previously taken decision, as we perform in the PoC.

Definition of standard interfaces towards the assisted system: The closed-loop automation of actions to be executed on the assisted system on the basis of the suggestions provided by the ENI system should be enabled through standard interfaces. In this PoC we integrated the ENI system with a network slice management and MANO framework based on two different NFV Orchestrators, each of them exposing its own APIs. The assisted system exposed a single, unified interface towards the ENI system, dealing with the adaptation to the NFV Orchestrators through NFVO-specific drivers.

3.4 PoC Suggested Action Items

Aspects related to security, privacy and dependability of the vertical integration in the architecture shall be considered by this ISG and other standard definitions activity.