Dynamic Deployment and Testing of Virtual On-board Units in 5G

Jorge Gallego-Madrid
Odin Solutions S.L.
Murcia, Spain
jgallego@odins.es

Ana Hermosilla
Odin Solutions S.L.
Murcia, Spain
ahermosilla@odins.es

Antonio F. Skarmeta
Odin Solutions S.L.
Murcia, Spain
skarmeta@odins.es

***Abstract*—5G networks are encountering virtualization technologies as the foundations of the softwarization of the infrastructure. The usage of these techniques in the Connected and Automated Mobility (CAM) vertical is the key to address mobility and computing issues. The next generation of CAM services are demanding continuous sensor-data gathering and processing, but current solutions lack of flexibility and computing capabilities in the On-Board Units (OBUs). Consequently, a dynamic intermediate stratum with adaptable networking resources and data processing offloading is required to cover the requirements imposed by the upcoming vehicular applications and users. Besides, due to the changing nature of these environments, dynamic testing and validation of the deployed services is necessary to assure their correct functioning. In this line, a solution that exploits the Multi-access Edge Computing (MEC) paradigm to instantiate virtual OBUs (vOBUs) to act as virtual counterparts of the physical ones is presented. By doing so, in-vehicle OBUs can be protected from the characteristic disconnections of vehicular networks using the vOBU as an intermediate communication layer. Besides, they can offload heavy computing processes to the edge. The solution is dynamically deployed as a Network Application (NetApp) in a real 5G testbed in the context of the 5GASP project, in which it is also possible to test and evaluate the functioning of the NetApp after the deployment.**

***Index Terms*—vOBU, MEC, CAM, 5G**

I. Introduction

The On-Board Units (OBUs) embedded in the vehicles are suffering an evolution during the last years, from being designed for a specific purpose and with dedicated hardware, to white box networking and computing nodes capable of doing multiple kinds of tasks and interconnecting all the actors involved in the Connected and Automated Mobility (CAM) vertical. These OBUs's main objective is to gather all the sensor-data collected by the vehicle and make it accessible to the infrastructure or other vehicles.

With the arrival of disruptive virtualization technologies such as Software Design Networking (SDN), Network Function Virtualization (NFV) and Multi-access Edge Computing (MEC), the flexibility of the network can be significantly improved [1], thus providing new opportunities for the design and development of innovative CAM services.

This work has been supported by Fundación Séneca—Agencia de Ciencia y Tecnología de la Región de Murcia—under the FPI Grant 21429/FPI/20, and co-funded by Odin Solutions S.L., Región de Murcia (Spain); by the Spanish Ministry of Science and Innovation under the DIN2019-010827 Industrial PhD Grant, and co-funded by Odin Solutions S.L.; and by the European Commission under the 5GASP (Grant No. 101016448) project.

One of the main beneficiaries of the advent of the aforementioned virtualization technologies are the 5G and beyond networks. This new cellular networks will capitalize the new capabilities of the infrastructure to offer state-of-the-art communication technologies and integrate the vehicular paradigm within its capabilities [2].

CAM services are increasing their requirements in terms of continuous sensor-data gathering and processing. However, current architectures and available solutions are based on fixed client-server models in which the OBUs collect the data and directly send the information to cloud servers. This ossified approach is outdated given the stated capabilities of 5G and beyond. Following the MEC paradigm, intermediate nodes with high caching and computing performance can be introduced as a middle layer between the OBUs and the cloud services. By doing so, processing tasks can be offloaded to the MEC, which can also be used as a cache to reduce the data retrieval time.

As 5G is increasing its maturity level, the development, testing and validation of the 5G services has become of paramount importance. Verticals demand very different requirements and diverse levels of support to transform their applications into real services that can be later marketed as products. Besides, the development of this kind of 5G services can be difficult due to the complex learning curve of cellular architectures and virtualization techniques. In this context, the H2020 5GASP project[1] (ICT-41-2020) started in 2021 with the goal of shortening the idea to market process through the establishment of a European testbed for SMEs to promote quick development and validation of innovative Network Applications (NetApps) using a 5G virtualized architecture. Connecting multiple existing physical infrastructures, 5GASP aims at offering a platform to automatically and dynamically deploy experiments and tests with Continuous Integration and Continuous Deployment (CI/CD) in a trusted environment [3].

In this work, we present the design, development, deployment and testing of one of the NetApps that are participating in the 5GASP project: the virtual OBU (vOBU), framed in the CAM vertical. In this way, we overview all the phases in the *vnification* of the service to be deployed in a real 5G testbed participating in the project. Besides, we also design and

[1]https://5gasp.eu

execute multiple tests following the 5GASP recommendations that validate the correct functioning of the vOBU NetApp once instantiated.

The remaining of the paper is organized as follows: Section II details the motivation of this work; Section III defines the vOBU NetApp and explains its architecture and functioning; Section IV elaborate on the design and development of the vOBU NetApp; Section V shows the deployment of the vOBU NetApp in a real 5G testbed; SectionVI includes the defined tests for the NetApp and their execution and, finally, Section VII concludes the paper and present the future work.

II. Motivation

Current approaches to gather data collected from vehicles directly retrieve it in the physical OBUs and send it directly to the cloud through the available connection in the vehicle. Vehicular communications typically suffer from disconnections due to the mobility characteristics present in this kind of environments. Although new wireless technologies are very promising, mobility-related issues are still to be completely overtaken. Besides, the next generation of vehicular services is increasing the requirements from the network infrastructure, demanding a continuous gathering of huge amounts of data and heavy computational tasks to properly process all this information. In this line, a need appears to address the challenging requirements claimed by the new breed of vehicular applications. To tackle these problems, an intermediate cache and processing layer arises as a promising solution, as proposed in [4]. On the other hand, the development of 5G services requires extensive knowledge of cellular architectures and virtualization technologies, which suppose a steep learning curve difficult to overcome.

In this way, the vOBU service is capable of virtualizing physical OBUs in the MEC with the aim of creating virtual counterparts that are always available, which was first demonstrated in the 5GINFIRE project [2]. The OBUs located in the vehicles are directly connected to the vOBUs to provide all the sensor data that will be required by the external services. Besides, the processing of this information can be performed in the MEC, saving resources from the OBU and increasing the performance of the system. Once processed, the data can be directly served from the vOBU, avoiding to reach the moving OBU, saving radio resources and decreasing the access time to the critical information. These vOBUs are dynamically managed by the virtualized infrastructure, which can flexible adapt service provisioning to the changing nature of vehicular scenarios. The vOBU can also benefit both from the communication technologies improvements made in the 5G New Radio (5G NR) [5] and from the dynamism of the 5G architecture enabled by SDN, NFV and MEC [6]. The development of the vOBU as a NetApp in the context of the 5GASP project will enable the integration of the solution in the real 5G world in an effortless way when compared with addressing this challenge with no support. Permitting its dynamic deployment and the automatic validation of the NetApp once deployed in the 5G network.

[2]https://5ginfire.eu/surrogate/

III. vOBU NetApp in the 5GASP project

The objective of the vOBU NetApp is to cover quasi-ubiquitous data-gathering and offload the processing of vehicular sensor data. To do so, it acts as an intermediate cache and processing layer between the physical OBU and the cloud, enabling the data cache to reduce the retrieval time of information and provide robustness against network disconnections.

The architecture of the NetApp can be seen in Fig. 1. The vOBU solution is composed by three main components, namely, the vOBU, the vOBU manager and the Data Aggregator. These are the three elements that conform the NetApp itself. Besides, the operation of the vOBU NetApp requires from the vOBU app running in the physical OBU and also a client that performs data requests to the vehicle.

The vOBU is the main component of the NetApp, it is instantiated in the MEC and multiple of them are ready to be used. They are directly linked in a 1-to-1 relationship with a physical OBU, acting as a virtual twin. The vOBU manager is the entity in charge of the control of the NetApp and located in the cloud. It receives the vOBU requests from the OBUs and it is in charge of the instantiation of the vOBUs on-demand. The Data aggregator is the entrypoint of the application in the cloud, it hosts databases that aggregate the information retrieved from the OBUs and vOBUS. Thus, acting as a cache in the higher layer of the architecture, the cloud.

The working flow of the vOBU NetApp is detailed as follows. Once the NetApp is deployed and running in the 5G infrastructure, the vOBU app is launched in the physical OBU of a vehicle connected via 5G NR to the network. The OBU then will request a vOBU to the cloud, which will be handled by the vOBU manager and a vOBU will be instantiated and assigned to the applicant OBU. In this point, the NetApp is fully up and ready to operate. Whenever any client interested on vehicle data send a request, it will enter the system through the Data Aggregator, which will not have that value cached in the first access so it will forward the petition to the vOBU in the MEC. As we are in the initial state, the vOBU neither will have the information required and it will request the data to the vOBU app through the 5G connection. The values will be then retrieved from the OBU and sent to the vOBU, which will forward them to the Data Aggregator. In this state, the requested information can be sent to the client and the latest values will be also cached in all levels of the NetApp. From now on, every time the vOBU app detects an update of the cached values, it will forward them to the vOBU and to the Data Aggregator, which will maintain as much as data freshness as possible.

As it can be seen in the detailed operation of the NetApp, once the service is up and running it will maintain the required data cached in all levels. The first accesses will be the most expensive ones in terms of latency and data-retrieval time, however, it will quickly improve when the data-caching starts

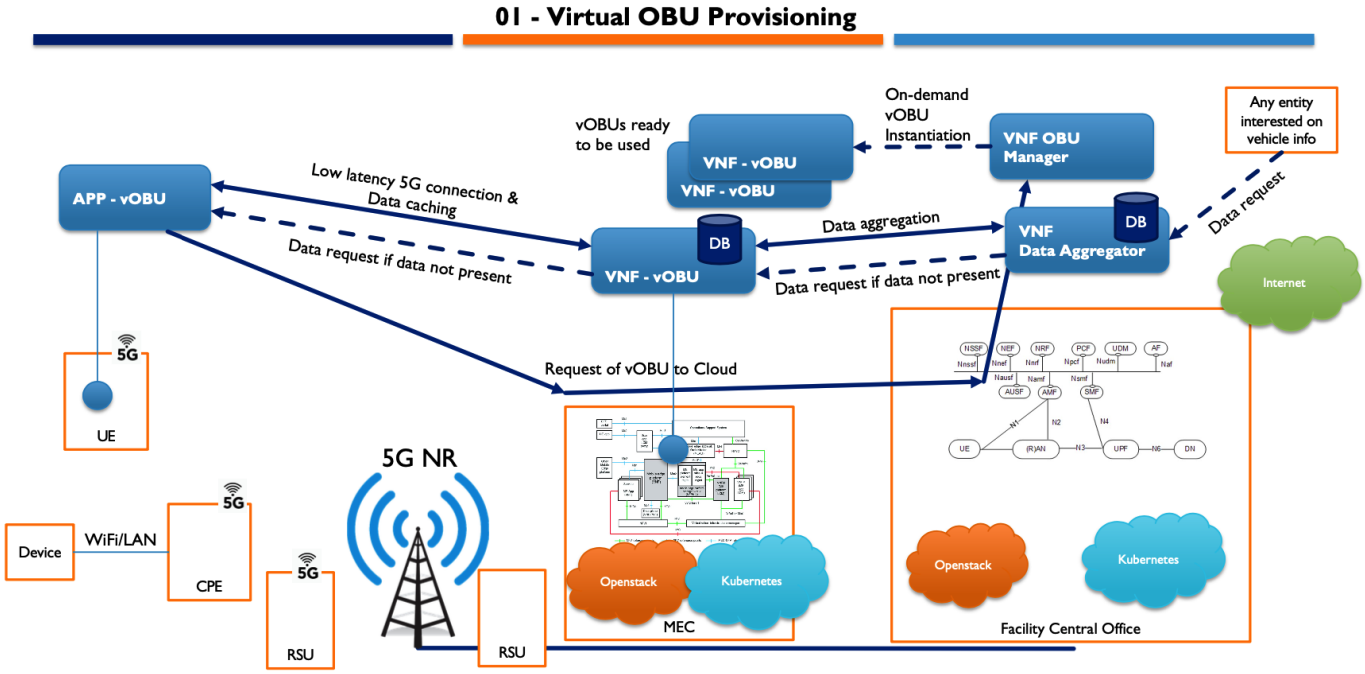

Fig. 1. vOBU NetApp architecture.

to work. Besides, the vOBU app can offload CPU intensive tasks to the vOBU to be performed in the MEC, which has more powerful hardware resources. All this operation is automatically performed and orchestrated by the 5G virtualized infrastructure and it operates in a transparent way so, from the point of view of the user, it is directly accessing the physical OBU every time it requires vehicle data.

Finally, the vOBU NetApp is being developed within the context of the 5GASP project. As aforementioned, the project aims at providing a European platform built on top of existing physical infrastructures. To do so, it defines a NetApp as a triplet composed by the descriptors of the application itself, the network slice requirements and the testing procedures that validates its functioning [7]. In this case, the vOBU NetApp is being mainly developed and tested in one of the 5G networks participating in the project, the GAIA-5G[3] testbed, located in the University of Murcia (Spain). It counts with a 5G SA network and it uses as Virtual Infrastructure Manager (VIM) Openstack[4] , OSM version 10[5] as orchestrator and Robot Framework[6] and Jenkins[7] as CI/CD tools.

IV. Design and Development

In this section, we present the design and development processes that has been followed to create the vOBU NetApp under the umbrella of the methodology elaborated by the 5GASP project [8].

[3]https://ants.inf.um.es/en/gaialab
[4]https://www.openstack.org/
[5]https://osm.etsi.org/
[6]https://robotframework.org/
[7]https://www.jenkins.io/

The NetApp will be created to work with OSM version 10, a management and orchestrator tool developed by the European Telecommunication Standards Institute (ETSI) and the chosen orchestrator in the 5GASP project and in the GAIA-5G testbed, the one that will be used for the deployment. In this way, the first step is to analyze the components of the service and define the Virtual Network Functions (VNFs) in which the NetApp will be divided and the Network Service (NS) that will comprehend the whole NetApp. This process is straightforward, as the vOBU has three main components that will be directly considered as three VNFs directed by a singe NS. Further important aspects of the design are the packaging of the VNFs, that will be deployed as Virtual Machines (VMs), the hardware resources required by each VNF and the necessities of Internet connectivity. With all this defined, the development of the VNFs Descriptors (VNFDs) and NS Descriptor (NSD) can be started. As an example, Listing 1 shows the vOBU component VNFD, where the information about that entity is detailed to be instantiated by OSM. In the descriptor, multiple network interfaces are defined together with the hardware characteristics of the hosting VM.

```
vnfd:
  description: vOBU entity
  id: vOBU_vnfd
  product-name: vOBU_vnfd
  provider: DIIC
  version: '1.0'
  df:
  - id: default-df
    instantiation-level:
    - id: default-instantiation-level
      vdu-level:
      - number-of-instances: 1
        vdu-id: vOBU-vnf-VM
```

```
    vdu-profile:
    - id: vOBU-vnf-VM
      min-number-of-instances: 1
      max-number-of-instances: 4
  ext-cpd:
  - id: vobu-vnf-eth0-ext
    int-cpd:
      cpd: eth0-int
      vdu-id: vOBU-vnf-VM
  - id: vobu-vnf-eth1-ext
    int-cpd:
      cpd: eth1-int
      vdu-id: vOBU-vnf-VM
  - id: vobu-vnf-eth2-ext
    int-cpd:
      cpd: eth2-int
      vdu-id: vOBU-vnf-VM
  mgmt-cp: vobu-vnf-eth0-ext
  sw-image-desc:
  - id: vOBU
    image: vOBU
    name: vOBU
  vdu:
  - cloud-init-file: cloud-init.cfg
    description: vOBU-vnf-VM
    id: vOBU-vnf-VM
    name: vOBU-vnf-VM
    sw-image-desc: vOBU
    int-cpd:
    - id: eth0-int
      virtual-network-interface-requirement:
      - name: eth0
        virtual-interface:
          bandwidth: 0
          type: VIRTIO
    - id: eth1-int
      virtual-network-interface-requirement:
      - name: eth1
        virtual-interface:
          bandwidth: 0
          type: VIRTIO
    - id: eth2-int
      virtual-network-interface-requirement:
      - name: eth2
        virtual-interface:
          bandwidth: 0
          type: VIRTIO
    virtual-compute-desc: vOBU-vnf-VM-compute
    virtual-storage-desc:
    - vOBU-vnf-VM-storage
  virtual-compute-desc:
  - id: vOBU-vnf-VM-compute
    virtual-cpu:
      num-virtual-cpu: 6
    virtual-memory:
      size: 16.0
  virtual-storage-desc:
  - id: vOBU-vnf-VM-storage
    size-of-storage: 50
```

Listing 1. VNF descriptor of the vOBU

Next, the network requirements of the Network Slice (NetSli) that will host the NetApp in the 5G network need to be defined. In this case, the 5GASP project has adopted the Generic Slice Template (GST) [9] defined by the GSMA[8]. Following this definition, filling of the GST gives as a result the NEST, which contains the requirements of the NetApp in terms of network characteristics. Table I shows the NEST designed and developed for the vOBU NetApp. It contains general details such as the area of service and the QoS identifier of the NetSli, but also performance requirements from the NetApp such as the maximum packet loss ratio permitted and the mobility support as this NetApp is under the CAM paradigm.

vOBU NetApp NEST	
Area of service	SP
Area of service: Region Specification	Murcia
3GPP QoS Identifier (5QI)	9
Maximum Packet Loss Ratio	1%
Supported device velocity	3: Vehicular: 10km/h to 120 km/h

TABLE I
VOBU NETAPP NEST

Finally, the testing of the vOBU NetApp will be made from two perspectives: the first one is from the infrastructure point of view, to evaluate the performance of the underlying network and assure that it satisfies the demanded requirements; and from the NetApp-specific operation viewpoint, validating that the vOBU NetApp is correctly functioning. More details about the performed tests will be explained in Section VI.

V. ONBOARDING AND DEPLOYMENT

The onboarding and deployment process of the vOBU NetApp goes through OSM and Openstack. OSM is in charge of the onboarding, it receives the VNFDs and the NSD of the NetApp to communicate then to Openstack and to solicitate the instantiation of the VMs with the characteristics indicated in the descriptors. Once the VMs are deployed, Openstack sends deployment-related information to OSM such as the assigned IPs and the NetApp is ready to operate. In Fig. 2, the different entities of the vOBU NetApp can be seen as VMs deployed in the GAIA-5G testbed's Openstack, connected to the dedicated networks available for 5GASP and compliant with the NetSli requirements. Note that three different vOBUs are instantiated in this concrete case.

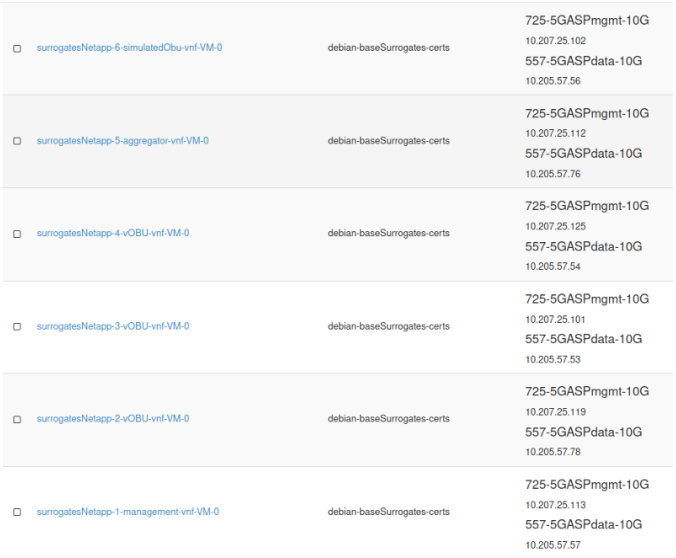

Fig. 2. vOBU NetApp instantiated in Openstack.

[8]https://www.gsma.com

VI. Testing and validation

Finally, once the vOBU NetApp is deployed and functioning, the testing and validation phase can start. The tests have been made using Robot Framework directives, employing Python language for the test development and Robot language for the validation of the test execution results. In this case, four infrastructure tests and two NetApp-specific tests have been defined and developed. The first ones validate the deployment time of the NetApp components, the connectivity among the different entities, the Packet Loss Ratio (PLR) among the components, and the latency among the different endpoints of the vOBU NetApp. On the other hand, the NetApp-specific tests are tailored to the operation of the vOBU. The first one checks that the APIs of all the components are up and correctly respond to the requests. The second one validates both the status of the databases embedded in the Aggregator and the vOBU and the correct functioning of the application flows by performing data requests to the NetApp. This queries are directed to specific values pre-inserted in the databases, which validate that the client can obtain a value present in the first layer, the Aggregator, or in the second one, the vOBU. The results log of an unsuccessful execution of this last test can be seen in Fig. 3. As explained before, it can be seen how the test has tried to access this two values but it has just found the one in the aggregator, as the one located in the vOBU was removed on purpose.

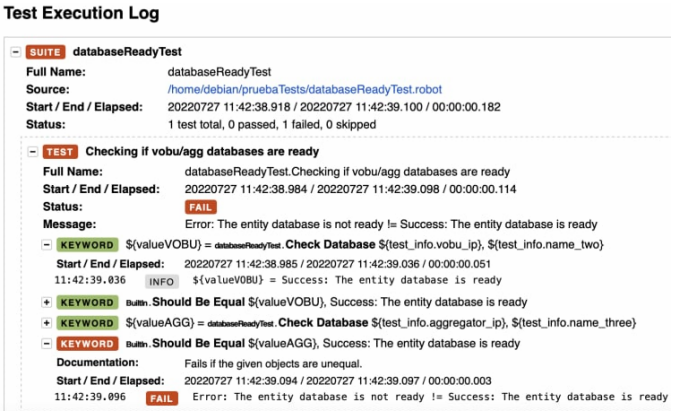

Fig. 3. vOBU NetApp database access test.

VII. Conclusions

Virtualization technologies are fostering the softwarization of network infrastructures, enabling the introduction of flexible and dynamic architectures. An example of this is 5G, in which virtualization is a fundamental pillar to address the stringent requirements demanded by the new services and applications. Besides, the next generation of CAM services are claiming the adoption of technologies that can handle huge amounts of sensor-data and computing resources. Present approaches lack of flexibility enough to fulfill these requirements.

In this way, in the context of the 5GASP project, we present the dynamic deployment and validation of the vOBU NetApp in a real 5G testbed. This proposal address both the needed flexibility by the network infrastructure and the CAM environment necessity for a solution capable of handle the next generation of vehicular services. As future work, the vOBU NetApp remains in continuous development, improving its performance and extending its capabilities. Besides, within the umbrella of 5GASP, a completely automated procedure from the onboarding to the testing and validation, going through the deployment, is being defined and the vOBU will be one of the first NetApps to be showcased in the developed framework.

References

[1] N. Kumar, S. Mittal, V. Garg, and N. Kumar, "Deep reinforcement learning-based traffic light scheduling framework for sdn-enabled smart transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2411–2421, 2022.

[2] B. Cao, Z. Sun, J. Zhang, and Y. Gu, "Resource allocation in 5g iov architecture based on sdn and fog-cloud computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3832–3840, 2021.

[3] K. Trantzas, C. Tranoris, S. Denazis, R. Direito, D. Gomes, J. Gallego-Madrid, A. Hermosilla, and A. Skarmeta, "An automated ci/cd process for testing and deployment of network applications over 5g infrastructure," in *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, 2021, pp. 156–161.

[4] S. Din, A. Paul, A. Ahmad, S. H. Ahmed, G. Jeon, and D. B. Rawat, "Hierarchical architecture for 5g based software-defined intelligent transportation system," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018, pp. 462–467.

[5] A. Filali, Z. Mlika, S. Cherkaoui, and A. Kobbane, "Dynamic sdn-based radio access network slicing with deep reinforcement learning for urllc and embb services," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2174–2187, 2022.

[6] X. Xu, Q. Huang, H. Zhu, S. Sharma, X. Zhang, L. Qi, and M. Z. A. Bhuiyan, "Secure service offloading for internet of vehicles in sdn-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.

[7] 5GASP. (2021) D3.1 5gasp experimentation services, middleware and multi-domain facilities continuous integration. [Online]. Available: https://www.5gasp.eu/assets/documents/deliverables

[8] A. S. Jorge Gallego-Madrid, Ramon Sanchez-Iborra, "From network functions to netapps: The 5gasp methodology," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 4115–4134, 2022. [Online]. Available: http://www.techscience.com/cmc/v71n2/45796

[9] GSMA. (2021) Generic network slice template, version, 6.0 25 november 2021. [Online]. Available: https://www.gsma.com/newsroom/wp-content/uploads//NG.116-v6.0.pdf