

From Network Functions to NetApps: The 5GASP Methodology

Jorge Gallego-Madrid¹, Ramon Sanchez-Iborra^{2,*} and Antonio Skarmeta¹

¹University of Murcia, Murcia, 30100, Spain

²University Center of Defense, General Air Force Academy, San Javier, 30720, Spain

*Corresponding Author: Ramon Sanchez-Iborra. Email: ramon.sanchez@tud.upct.es

Received: 13 July 2021; Accepted: 14 August 2021

Abstract: As the 5G ecosystem continues its consolidation, the testing and validation of the innovations achieved by integrators and verticals service providers is of preponderant importance. In this line, 5GASP is a European H2020-funded project that aims at easing the idea-to-market process through the creation of an European testbed that is fully automated and self-service, in order to foster rapid development and testing of new and innovative 5G Network Applications (NetApps). The main objective of this paper is to present the 5GASP's unified methodology to design, develop and onboard NetApps within the scope of different vertical services, letting them use specific 5G facilities. Besides, we examine the whole 5GASP process in a tutorial fashion by adopting a specific use case focusing on the integration of a virtual On-Board Unit (vOBU) service that permits offloading processing from the attached vehicle and serving data-access requests. As demonstrated, the presented workflow permits the agile, rigorous, and safe development, testing and certification of NetApps, which will enable valuable in-network services for 5G and beyond infrastructures.

Keywords: 5G; NetApp; 5GASP; vOBU

1 Introduction

5G physical infrastructures are reaching a notable level of maturity. Given their potential to support a plethora of heterogeneous services for a massive number of users, their efficient and dynamic management is a crucial aspect [1]. To this end, the evolution of the Network Function Virtualization (NFV) paradigm is enabling an in-network flexibility never seen before [2]. However, the integration of network functions developed by third-parties within operative 5G infrastructures must follow a rigorous testing and evaluation process for security reasons [3]. This is the principal aim of the European H2020-funded project 5GASP (<https://5g-ppp.eu/5gasp/>) that, to this end, is currently building an open and inter-domain 5G NFV-based ecosystem of experimental facilities. This wide environment of 5G facilities integrates existing infrastructures deployed and tested in previous H2020 projects, e.g., 5GVINNI (<https://www.5g-vinni.eu/>), 5GinFIRE (<https://5ginfire.eu/>), or SLICENET (<https://slicenet.eu/>), among others. In addition, the 5GASP infrastructure will also lay down the foundations for instantiating fully softwarized architectures of vertical industries with the objective of



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

enabling the testing and validation of novel network functions, or Network Applications (NetApps) according to 5GASP terminology. Once a NetApp is approved and certified, it is put at disposal of the community through a NetApp Marketplace.

The principal aim of this paper is to present the initial proposal of the unified methodology approach to design, develop, and onboard NetApps within the scope of the 5GASP project [4]. Depending on the NetApp requirements, this process can be performed in vertical-specific 5G facilities by following standard interfaces in NFV and 5G solutions [5]. At the same time, the interoperability across facilities is also supported to cope with multi-domain scenarios. The intention of the proposed methodology is to define how 5GASP solutions will be defined, modeled, designed, and implemented to conform a NetApp to be automated, integrated, and tested in the envisioned Continuous Integration and Continuous Delivery (CI/CD) process. This methodology is defined according to the requirements of the ETSI standard of OSM/SOL006 (YANG model). The generic process definition helps to enhance NetApps from their design phase and proposes assessment procedures and Key Performance Indicators (KPIs) of interest on the targeted vertical. Besides, this methodology also provides guidelines to NetApp developers to help the service creation time KPI by suggesting optimizations and best practices. 5GASP has strong relation with state-of-the-art initiatives from other projects and standardization bodies [6].

Therefore, in first place, a contextual overview of the 5GASP project is given to examine its aims and objectives. Thereafter, in order to provide a general vision of the steps that a developer or an experimenter should take to prepare and onboard a designed NetApp, we introduce the design, development, testing, and validation methodology that will be followed in the 5GASP platform. To this end, the lifecycle of this procedure is presented and integrated within the global 5GASP platform. Next, the NetApp onboarding approach is thoroughly presented, describing how the NetApps are managed in the 5GASP platform and the categories in which they can be divided. Also, it is discussed how the NetApps are defined with a combination of descriptors that represent: The Network Service (NS) itself, the Virtualized Network Functions (VNFs) of which it is composed, the network slice in which it will be located once deployed, and the testing and validation requirements that the NetApp has to pass to be validated and certified by 5GASP and to be available in the NetApp Marketplace for the community. Finally, a specific use case is presented in a tutorial fashion in order to show the whole 5GASP process. The adopted service permits the deployment of a virtual On-Board Unit (vOBU) in edge-computing premises, aiming at offloading processing from the physical vehicle's OBU as well as serving data-access requests for reducing latencies and increasing reliability. The rest of the document is organized as follows. Section 2 presents the main objectives of the 5GASP project. Section 3 describes the development stages of a NetApp. The NetApp onboarding procedure is comprehensively presented in Section 4. A specific NetApp integration use case is presented in a tutorial fashion in Section 5. The paper is concluded in Section 6, which also presents the next steps that will be taken within the context of the 5GASP project.

2 The 5GASP Project: Objectives and Validation Indicators

2.1 5GASP Aims

The principal goal of the 5GASP project is the implementation of an open platform to automate the development, validation, and certification of innovative 5G and beyond NetApps. For this to be done, a number of specific objectives have been defined. Firstly, the project aims at the acceleration of the development, testing, and certification of 5G Network Applications (NetApps), through the creation of a common platform, Development and Operations (DevOps) tools and a certification

roadmap (Objective 1). To reduce the time-to-market for novel NetApps from third party developers, an integrated and automated DevOps methodology is the most adequate procedure. Furthermore, 5GASP will address the challenges of validation, verification and certification of NetApps, so that operators are aware of their behavior before deploying them on their production network. To achieve all the above, 5GASP introduces novel procedures and a novel fully automated toolchain that caters for the production of any NetApp without human interaction. Besides, the roadmap of the project also comprises a community portal where design and development guidelines, technical tutorials and articles will be posted online in order to ease the introduction to 5G. By doing so, the required knowledge about the 5G infrastructure and services for developing this kind of NetApps will be reduced. Therefore, non 5G-familiar technological SMEs will find the slope to enter the 5G market less pronounced.

5GASP will provide state-of-the-art testbeds where applications for relevant verticals can be tested and validated in a cost-effective way (Objective 2). This is another key aim of the project, which is building an open and inter-domain 5G NFV-based ecosystem of experimental facilities on top of existing 5G infrastructures developed in previous projects such as those mentioned above. Thanks to this heterogeneous set of testbeds, 5GASP will address inter-domain use-cases, and security and trust aspects associated with NetApp DevOps (Objective 3). In 5GASP these issues are pragmatically addressed from the viewpoint of the companies developing and deploying their NetApps, requiring safety for their Intellectual Property Rights (IPR) and from the viewpoint of the network operator which must trust on the correct and limited behavior of the NetApp in its network. With this ecosystem of 5G testbeds, the 5GASP framework aims at becoming a reference in the design and development of NetApps and a fundamental actor in the industrial market processes for the SMEs. Another important aspect tackled by the project is the automation of the testing and validation process, lowering costs associated with testing and certification of NetApps in telecommunication environments (Objective 4). In 5GASP, the focus is on providing the blueprints for all the stakeholders in the ecosystem, in accordance with their capabilities, to be able to setup automated testing of NetApps. In this line, it is crucial to provide all the community with state-of-the-art tools for test deployment, test automation, continuous integration and monitoring of testbeds, mainly through Open-Source Software (OSS) tools (Objective 5). Therefore, 5GASP plans the development of software tools that can assist the NetApp developers in the integration of their 5G applications into a Continuous Integrations framework, targeted at their specific vertical, consisting mostly out of OSS tools and producing reports that can be used to start certification processes.

Finally, 5GASP aims at creating a business model around a marketplace of NetApps, by which all stakeholders can share revenue (Objective 6). The purpose of this portal is to promote the developers and their respective NetApps as well as the platforms and operators that support them. Furthermore, this portal has the potential to be marketed in a way similar to stores for mobile apps, such as Apple's store, Google Play, and Huawei store. Besides, the portal will include information of tests/validations the NetApps have successfully passed, as well as testbeds and network operators that have deployed successfully the NetApp. In this way, even the SMEs that are not familiar with the development and deployment of 5G-related services and NetApps will be able to acquire them by means of the marketplace. Thus, being able to obtain and deploy already proved and certified solutions that can be used to address their requirements. The whole NetApp development-to-onboarding process is presented in the following sections.

Until now (first year of the project), 5GASP has centered its efforts in the definition of the general architecture and in the design of the CI/CD pipelines and methodologies. By doing so, strong conceptual and theoretical foundations have been established to ease the development of the

technical work. In this way, the framework has been envisioned and prepared to adopt state-of-the-art virtualization technologies such as SDN, VNF, and MEC. While SDN and VNF technologies will provide high network programmability and flexibility, the adoption of a MEC-based infrastructure permits to bring computing resources close to end-users [2]. These technologies will be implemented using open-source solutions in each one of the test sites, each one with its own capabilities, and they will be available, in a transparent way, through the 5GASP portal. Once the architectural background is consolidated, the project will move its focus to the NetApp development, as well as the setting up of the first proof of concept of the framework. This will be done with the aim of having the first prototype version of the portal. At that point, it is expected that the portal will be online, with some NetApps available in the market-place and with the possibility of onboard and test new ones in any of the facilities. Besides, 5GASP plans to participate in standardization activities related to C-ITS (ISO TC 204, CEN TC 278, ETSI TC ITS), extended vehicle (ISO TC 22), and ETSI's MANO and MEC.

2.2 5GASP Key Performance Indicators (KPI)

In order to validate the achievement of the aforementioned objectives, a series of KPIs have been identified:

- Objective 1:
 - Time to automatically perform a CI/CD process of NetApp service (creation time in minutes).
 - Number of open source contributions in flexible licensing scheme for easy adoption by third parties.
 - Number of NetApps, Network Functions, and Network Services registered at the marketplace.
- Objective 2:
 - Number of new Vertical use cases accessible from testbeds.
 - Zero-code line change requirements for validating NetApps.
- Objective 3:
 - Inter-domain NetApps deployed over multiple testbeds.
 - Security audit of testbeds and NetApps.
- Objective 4:
 - 95% adoption of automated testing and validation by all NetApps deployed over Vertical Use-Cases.
 - Automated testing and validation of a NetApp in minutes.
- Objective 5:
 - Number of NetApps testing suites using an Open Source License.
 - Time to automatically perform a CI/CD process to for a new NetApp: < 15 min.
- Objective 6:
 - Number of NetApps developed by 5GASP's Consortium SMEs registered in the NetAppsStore: ≥ 7 .
 - Number of third-party SMEs/operators/service providers registered on the NetApps-Store: ≥ 10 .

3 5GASP NetApp Lifecycle

In the following, the NetApp design, development, testing, and validation methodology that is followed in the 5GASP project is presented. The objective of defining such methodology is motivated

by the need of bringing the NetApp developers and the 5G ecosystem together. In this way, the methodology will reduce the time and costs of making new 5G NetApps or adapting the existing ones to these new technologies. In order to ease the development process of NetApps, a well-defined lifecycle has been defined within the project methodology. This also permits the production of high-quality NetApps that meets the expected customer requirements at the time of optimizing development time and cost. As can be seen in Fig. 1, it consists of 4 phases which are described as follows.

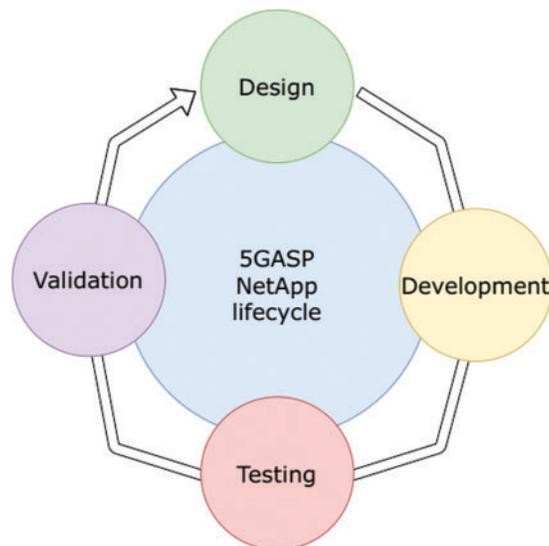


Figure 1: 5GASP NetApp lifecycle

3.1 Design

Certain considerations should be taken into account by a NetApp developer when designing a network function to be onboarded in the 5GASP environment. This is necessary in order to ensure that the 5GASP framework is able to adequately deploy and evaluate the different NetApp components. The design phase is divided in three different steps. Firstly, the main components of the NetApp should be defined in terms of services and interworking service components, in other words, VNFs in the 5G terminology. This is done by using Network Service Descriptors (NSDs). Therefore, each NetApp should be split into the different VNFs composing it, and the relation among them is described by using NSDs. In this first step, the programmer should also define the cardinality between VNFs and NSDs given that an NSD can consist of more than one VNFs. Each NSD shall thoroughly describe the NS under consideration by taking into account some of the following aspects, among others:

- Number of VNFs composing the NS.
- Type of packaging employed for the network function. For example, if the service should be packed as a Virtual Machine (VM) or containerized.
- Maximum tolerated latency in milliseconds.
- Need for connectivity with the Internet.
- Hardware resources required to deploy the service, e.g., CPU cores, RAM, storage requirements, etc.

- Services' delivery model, e.g., if the function is packaged as a VM image or in a docker container and its location.
- Ingress and egress bandwidth requirements of the service.

Besides, during this first design phase, the dependencies with other NetApps should be also identified. This is notably important in the case of implementing inter-working NetApps, which collaborate to meet the requirements of a given vertical use case. The second step of the design phase addresses the detailed definition of the requirements of the NetApp from the perspective of the 5G infrastructure. Concretely, the programmer should indicate how many network slices are needed as well as their characteristics. This is done by means of a Network Slice Type (NEST) document that, as explained in Section 4.1.2, is a filled Generic Network Slice Template (GST) as introduced by the GSM Alliance (GSMA). Finally, the third stage of this phase is intended to define the set of tests that must be carried out by the 5GASP platform in order to validate the NetApp under evaluation. This permits the avoidance of errors during its instantiation or operation on the targeted vertical use case. The tests should be described according to the test descriptors presented in Section 4.1.2.

3.2 Development

Once the design of the NetApp is performed, the developer can start to develop the different descriptors that conform the NetApp as a whole. To do this, the methodology considers a division of the development phase into three steps: (i) the VNF Descriptors VNFDs/NSDs, (ii) the GST/NEST, and (iii) the test scripts and test VNFs descriptors. The development insights of the NetApp functionality itself, i.e., the code of the application, are out of the scope of this methodology definition.

The first descriptor to take into account is the VNFDs/NSDs that compose the NetApp. These descriptors must follow the standards that will be supported by the NFV Orchestrators (NFVOs) offered by each one of the test sites, e.g., OSM or ONAP. This information is available beforehand to the experimenters in order to avoid problems with the descriptor structure and keywords. Depending on the type of NFVO to be used, the format of the descriptors varies. Moreover, developers have to take into account the version of the NFVO, as there may be differences between them. For example, the format of a descriptor for OSM Release NINE is not the same that for OSM Release EIGHT (or previous releases). To prevent the developer from having to be aware of these details, 5GASP offers the specific type and version of the NFVO available on testbeds, as well as examples for them. Furthermore, some examples of descriptors are available to simplify the developer's job, easing the task avoiding their creation from scratch. It could be also possible to offer some "ready-to-fulfil" descriptors, which could be customized by developers (for example, the number of hardware resources, or network interfaces). Also, a list of network and computing resources available on each facility is offered to developers, where they can easily select the resources they want to use in their NetApps.

Secondly, when preparing the GST/NEST descriptor for the NetApp, it is important to take into account the capabilities of the testbeds. A list of predefined NESTs for each testbed is offered to the developers when onboarding the NetApp, therefore the network slice assigned to the NetApp in the deployment is one of the predefined ones for each testbed. Further on in the project, the experimenter will be also able to select the desired capabilities for the network slice, and a series of available NESTs from the different testbeds will be offered, leaving the decision of which one will be used to the developer. Finally, the 5GASP framework will also accept the NESTs defined by the experimenter, choosing a best-effort option if the requirements demanded by the NetApp cannot be fulfilled by any testbed. As a consequence, it is important for the developer to know the network resources that the

NetApp will demand once deployed, in order to achieve an optimal operation. Also aiming at passing all the test and validation procedures to obtain the certification by the 5GASP platform.

Finally, regarding the development of the testing descriptors, depending on the type of tests the approach may change. In first place, multiple infrastructure-related tests are pre-provided by the different test facility sites. Thus, the developer can avoid implementing this kind of tests, as they will be available to select during the onboarding process, and they will have the correspondent KPIs associated, in order to be included in the NetApp descriptors. Moving on to the custom test scripts, the code itself is under the responsibility of the developer, and he/she needs to define an output to establish the success or failure of the test. Regarding the custom test VNFs, the development considerations are similar to the ones presented and discussed above in relation to the NetApp VNFDs/NSDs. It is important to mention that from the point of view of these test VNFs, the NetApp must be considered as a black box with some inputs and some outputs, which are the ones used to validate the test. In this way, these VNFs commonly embed specific applications with a certain configuration prepared to validate a concrete aspect of the deployed NetApp. For example, a testing VNF could be a traffic generator together with a certain data packet trace, in order to evaluate the behavior and response of the NetApp when that specific traffic is received and processed.

It is expected the creation of a practice community, where developers can share knowledge about 5G NetApps. The socialization of NetApp developers is of outmost importance in the success of the ecosystem. Through a tight support community, built upon trust and information sharing, the NetApp development process will be enhanced by the adoption of novel development technologies and tools that may be adapted to the subsequent 5GASP NetApp validation process.

3.3 Testing

This is a crucial and mandatory stage of the lifecycle that permits to inspect and thoroughly examine a developed NetApp with the aim of ensuring that it does not present errors of security defects. To this end, the NetApp components are executed by means of automated or manual tools, which evaluate them from different perspectives. In this line, besides potential errors or gaps, the basic NetApps requirements defined in previous steps are analyzed. Thus, the 5GASP NetApp testing phases are the following:

- 1 NetApp and test VNFs deployment in test environment, which includes the scheduling and planning of the tests as well as the actual deployment of NetApps and test VNFs.
- 2 Test execution when the appropriate test procedures are run under controlled circumstances.
- 3 Monitoring and analysis, which thanks to a developed monitor infrastructure, NetApps and test VNFs are analyzed by means of the evaluation of pre-defined metrics and extraction of KPIs.

The specific test procedures are being currently developed in the project and are out of the scope of this work.

3.4 Validation

Given the variety of 5G and beyond infrastructures [7], it is of prominent importance the validation and verification of NetApps in different execution environments. This permits infrastructure operators to be sure that the acquired and deployed NetApps will behave as expected in production platforms [8]. During this stage, the NetApp under validation is evaluated at the end of its development

process with the aim of determining whether it satisfies its functional requirements. To this end, several KPIs can be employed:

- 1 Functional Suitability (functional completeness, functional correctness, functional appropriateness).
- 2 Performance efficiency (time-behavior, resource utilization, capacity).
- 3 Compatibility (co-existence, interoperability).
- 4 Reliability (maturity, availability, fault tolerance, recoverability).
- 5 Security (confidentiality, integrity, non-repudiation, accountability, authenticity).
- 6 Maintainability (reusability, analyzability, modifiability, testability).
- 7 Portability (adaptability, instability, replaceability).

Given the importance of the validation process, the ETSI Industry Specification Group (ISG) in NFV has developed and released a series of documents (<https://www.etsi.org/newsroom/blogs/entry/public-availability-of-etsi-nfv-isg-drafts>) framing the pre-deployment validation process, considering interoperability and portability issues. 5GASP is making use of these guidelines for developing adequate procedures that will permit to comprehensively evaluate the aforementioned KPIs.

4 NetApp Onboarding Process

In this section, we provide additional insights about the NetAPP onboarding process, concretely the focus in on the NetApp management, the types of NetApps, and the descriptors that precisely define them. As explained previously, NetApps are understood as a set of virtual functions, which facilitates an automated and repeatable deployment and testing cycle. In order to follow standardized procedures, NetApps should be designed and developed following models proposed by well-known standardization bodies. For that reason, in 5GASP the ETSI's SOL005/OSM (YANG) and SOL001/ONAP (TOSCA) models have been adopted. Consequently, adaptations and enhancements could be made on those models to enable particular deployment and testing scenarios. Furthermore, the 5GASP platform provides developers with a single entry-point by means of a portal to enable a straightforward procedure towards the onboarding process. This portal allows any developer to onboard the designed NetApp, specify the accommodating testbed to host it, and describe the tests that should be triggered once the NetApp is deployed. This "triplet" is bundled together in a single entity creating a unified abstraction for all sites, thus assuring that the onboarding, activation and testing can be properly performed on any 5GASP facility. Also, with the aim of achieving a full interoperability with any NFV/3GPP-compliant 5G system, the 5GASP's approach for each entity of the "triplet" is to be defined under widely embraced models in the industry, i.e., GSMA's GST/NEST, TMF's Service Specification and Service Order, as further detailed in Section 4.1.2. Finally, as mentioned above, NetApps can be implemented as ordinary VM-based approaches or as the more contemporary container-based one. Both strategies are supported by the 5GASP platform and present each pro and cons as explained in the following.

4.1 Netapp Categories

Although the NFV precepts are completely agnostic about how VNFs are instantiated, in practice the most common method so far is deploying them as VMs. Nevertheless, in recent years the instantiation of VNFs as containers has become more popular, since its deployment, scaling, etc., is considerably faster and requires lower resources. For certain purposes, it may be necessary to use one or the other depending on the characteristics and needs of the NetApp, or even depending on the

facility where it is going to be deployed. The 5GASP platform support both possibilities transparently to the NetApp developer.

NetApps as VMs

During the last years, deploying VNFs as VMs is the most widely used way to instantiate them. A VM-based VNF is a virtual machine with hardware resources, network interfaces, and essentially, an image, usually *qemu*-based, that implements the desired functionality. Therefore, one of the most important points when deploying a VM-based VNF is to have the image that conforms the VM, since it is the one that will contain the functionality offered by that VNF. To do this, normally a previously configured image is available with basic functionality (for example, the necessary packages installed), ready to receive the specific configuration required (IPs, targets, configuration files, etc.) after its instantiation. Thus, it is obtained a generic-enough image to adapt to the characteristics of the scenario, but specific-enough for not having to do all the required configuration after the instantiation process, which reduces the time needed until the VNF is ready. To automate the image generation, other related projects such as 5GTANGO (<https://www.5gtango.eu/>) have developed methodologies to create VM-based images which consists of selecting a pre-existing docker container (or uploading one) that implements the required functionality. Once identified, it generates a VM with a Vanilla OS, e.g., a freshly installed Ubuntu distribution, and installs the container on it. Then, a new VM image is generated based on that VM, so an image with the required functionality is ready to be instantiated. Another typical option is to use a base image and configure it when deployed using different methods, e.g., Ansible. In this case, a *yaml* file includes the packages and functions to be installed, defining with a template and scripting system, the configuration files to be deployed on the VM to properly configure the VNF. Another possibility to inject configuration is using Day-0 and Day-1 configuration (from OSM) that uses Juj Charmes to configure instances [9].

NetApps as Containers

The other approach to build cloud-native 5G NetApps is by means of Containerized Network Functions (CNFs). The principal aim of CNFs is to reduce the weight of VMs, as CNFs consist of a series of micro-services that can be flexibly instantiated in different targeted systems. This approach also provides low-latency and ultra-reliability guarantees, among other advantages. The motivation for the development of a CNF-based VNF infrastructure is that monolithic network functions implemented as VMs require a long time to be deployed. Thus, making VMs much less scalable than CNFs, especially considering specialized container-management tools such as Kubernetes, which also provides monitoring tools for quick and smart CNFs handling. Following this approach, VNFs can be rapidly replaced or moved to different points within the network infrastructure attending to current needs. Thereby, the reduced footprint of microservices and their fast instantiation and launching times bring a range of advantages in terms of high performance to deal with the requirements of advanced 5G services. This is especially critical in the case of ultra reliable low latency communications (URLLC) applications [10].

4.2 Netapp Descriptors

As mentioned in previous sections the approach followed by 5GASP is to create a “meta-package” which includes the network slice requirements (NEST information), the NSDs that conforms the NetApp (with their respective VNFDs), and the tests the user wants to perform over them to validate the NetApp and obtain the 5GASP certification. These packages can be completely included in the meta-package, or they can be references to already uploaded descriptors available in the repositories.

VNFD/NSD Packages

Besides the descriptor, the packages can include other relevant information, for example, the image to be used, cloud-init files, some after-deployment configuration such as charms in OSM packages, or scripts following the TOSCA schema, etc. These files must be included in the package following the established format as defined by the orchestrator. As aforementioned, the idea is to work with a meta-package that is compatible with both types of descriptors. The meta-package should indicate which kind of descriptor each one is, e.g., OSM and which release, and afterwards, based on this meta-package, the descriptors for the specific orchestrator would be generated.

Depending on the type of orchestrator required (OSM or ONAP), the format of the descriptor varies, likewise in accordance with the used version of the orchestrator, given that there is not a common format between OSM Rel. EIGHT and OSM Rel. NINE, as the first one uses ETSI SOL005 and the second one ETSI SOL006. In this way, a proper verification must be performed in order to ensure the correctness of the package. Package examples are available in the orchestrator's repositories as well as in the 5GASP repository to be used as a template. Thus, developers do not need to create their packages from scratch, as they have a guideline to perform it.

GST/NEST

Network slicing is one of the key enablers of the development of 5G technologies. 3GPP defines it as a dedicated logical network that provides specific network capabilities and network characteristics. These slices are usually bonded to a Service Level Agreement (SLA) agreed between a Network Slice Customer (NSC) and a Network Slice Provider (NSP). A network slice can be instantiated across multiple domains of the network and it is composed by a set of dedicated or shared resources. If dedicated resources are used, the slice can be isolated from other network slices.

The Generic Network Slice Template (GST) is a set of attributes that can characterize a type of network slice or service, it is generic and it is not tied to any specific network deployment [11]. This template was introduced by the GSMA with the aim of introducing guidelines for verticals on how to address their service requirements and to facilitate the establishment of SLAs between operators and business customers. The Network Slice Type (NEST) is a GST filled with values. This set of attributes with certain values comply with a given collection of requirements derived from a use case defined by a network slice customer. The NEST is used as an input for the preparation of a network slice instantiation performed by the network slice provider. Furthermore, multiple network slice instances can be created out of the same NEST, and existing instances can be also reused. The network slice preparation process is illustrated in Fig. 2. The NSC provides the requirements of its particular use case to the NSP. Then, the latter generates a NEST by mapping these service requirements into the attributes of the GST. GST/NEST is the selected template to be used as the network slice descriptor in the NetApps definition. Each designated testing facility of the 5GASP project provides information about the network requirements they support by the means of NESTs. Therefore, the template is perfectly aligned with this GSMA standard. By doing so, two different approaches can be followed by NetApps developers. The first one is to define and fulfill the NEST associated with its NetApp and including the network requirements the NetApp will demand. The second one is to select a pre-defined NEST which is associated with a certain 5GASP facility. While the latter is the best starting point and do not raise any issues, the former has to be handled with care, as the facility test sites need to support and implement the network requirements defined by the NESTs. Thus, in the case that none of the sites can fulfill the required network resources as a whole, a best-effort policy is followed and the testbed with the most similar network characteristics is selected. By doing so, the design and development phase of the NetApp is simplified from the point of view of the experimenter.

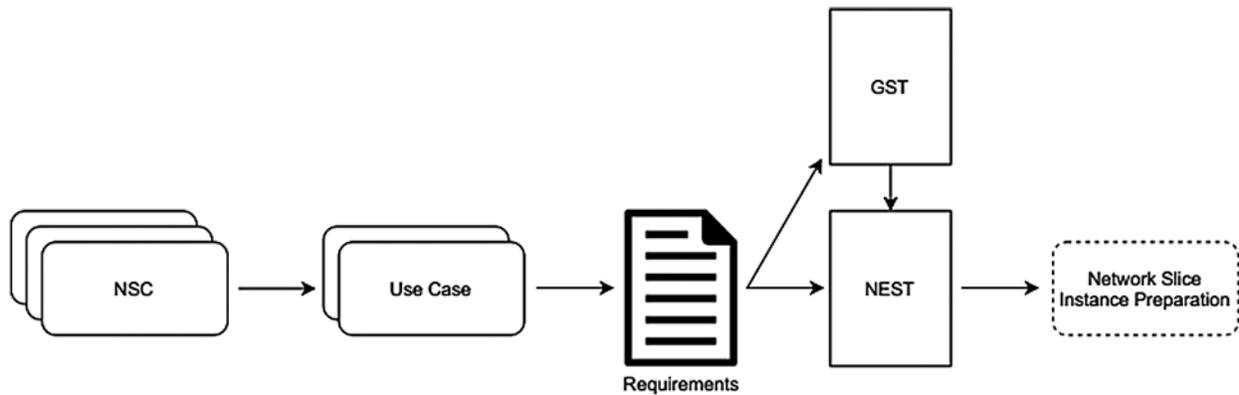


Figure 2: Network slice preparation process

Test Descriptor

To simplify and provide high generalization capability to the testing phase, 5GASP aims to design its own test descriptor. In 5GASP, there are two different types of tests: the tests included as scripts and test VNFs. A test VNF is a VNF whose purpose is to perform some test and that it is not involved in the functionality of the NetApp itself. In this sense, the difference between them is that the test VNF is a separate VNF that is requested to execute the tests, whereas the scripts can be executed inside the NetApp VNFs (the ones we want to analyze) or in a different place. It is also remarkable the possibility of using simple infrastructure tests (available in each test facility site), such as *iperf*, *ping*, or similarly, in order to check the basic connectivity and operation of the VNF(s) under evaluation.

Moreover, the platform includes a test repository, with predefined and simple tests to be used, and the possibility of creating custom tests using the test descriptor. In this way, the main characteristic of the proposed test descriptor is its ability to be divided into different steps or phases, namely, “Setup”, “Execution”, and “Validation”. These different steps are described below:

- **Setup:** This phase includes the definition of the scripts and the VNFs that will be used to perform the test, and also the deployment of the test VNFs. Regarding scripts, they could be included in the package as executable files, or the commands could be directly included inside the test descriptor.
- **Execution:** This phase includes the launching of the scripts (both VNF and bare scripts, in the preferred order or simultaneously), and later the collection of metrics obtained from the executions.
- **Validation:** Finally, this phase includes the analysis of the obtained data, the computing of the KPIs using the obtained metrics as input, and lastly the validation of both KPIs and tests.

These phases are still under study and consideration and they may suffer some changes during the following stages of the project, as additional adjustments that better fit the needs of the involved use cases may be found.

4.3 Onboarding Workflow

The onboarding procedure in the 5GASP system involves the uploading to the 5GASP portal of the NetApp, by means of the triplet of descriptors described in previous sections. The onboarding is performed by the NetApp developer itself, or by a NetApp experimenter, and the interaction is done against the unique 5GASP portal. The process is totally interactive with the user, requiring more actions in the first versions of the platform and evolving towards a more automatic procedure during its further development.

The initial onboarding procedure is depicted in [Fig. 3](#). It shows the different steps that conform the onboarding in the initial stages of the 5GASP project, in which the NetApp descriptors are uploaded one by one. By doing so, the developer is able to interact with the 5GASP platform to configure different NetApps and testing different parameters in each step. The process starts with the upload of the NetApp VNFDs/NSDs, which are stored in a catalogue. Then, based on the submitted NSDs and their respective network requirements, the developer should select the test site that fulfill the latter. In advanced 5GASP versions, the platform will automatically compare the network requirements with the capabilities offered by the multiple facility test sites, filtering out the unfitting ones or even automatically selecting the ones that match the criteria of the NetApp. The list of test sites is shown to the developer as a list of NESTs that represents each facility, each one defines the host slice that will be reserved if the site is finally selected. The developer chooses one of them and this information is stored together with the NetApp VNFDs/NSDs. Alternatively, the developer can select among a list of predefined NESTs, offered by the 5GASP portal, the ones best suited to the NetApp. Based on this selection, the 5GASP portal offers to the developer a list of KPIs that can be measured in the NetApps deployed in the corresponding facility test site. Then, the platform offers a list of custom tests available in the test repository, which the developer may select to test the NetApp. These custom tests can be predefined in each facility. Besides, in more advanced project phases, the developer of the NetApp, apart from the default testbed tests, will be able to upload its own test suites (in form of custom test scripts or test VNFs) that could extend the available test cases. At this point, the NetApp is perfectly defined in the 5GASP portal, and it is composed by the three descriptors: (i) the NSDs/VNFDs, (ii) the NEST, and (iii) the tests descriptors. Now, the descriptors will be sent to the NetApp transformation service, which will automatically enhance the descriptors, looking for bad practices and trying to correct them. Once this is done, the triplet of descriptors can be bundled as a service order and pass a static validation of the descriptors, to check the validity of them. If succeeded, the onboarding procedure is complete. Finally, the 5GASP platform can trigger the CI/CD pipeline interacting with the CI/CD service manager by sending to it the required information for the deployment and for the testing of the NetApp.

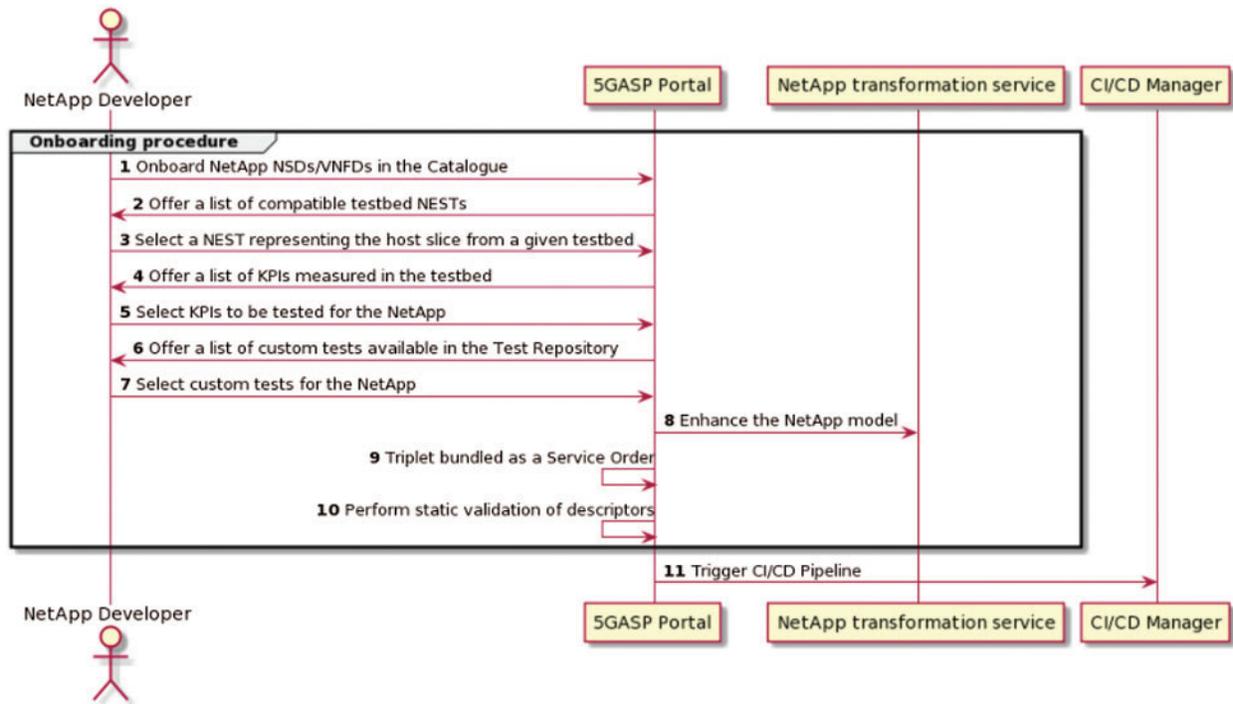


Figure 3: NetApp onboarding and connection to CI/CD service

5 Use Case

To illustrate the designed methodology presented previously, in this section we provide an example of the application of this methodology to the design, development, and onboarding of one of the NetApps that are exploited within the 5GASP project: The vOBU. This NetApp proposes the virtualization of vehicle physical OBUs with the aim of creating a Mobile Edge Computing (MEC) layer to offload heavy computational tasks from the vehicle and to serve data-access requests [12]. By doing so, it provides the system with robustness against potential disconnections periods from the vehicle, it saves radio resources on the link, and improves data processing performance.

5.1 VOBU Netapp Design

The first step to design the NetApp is to analyze its functioning and architecture. This is because the programmer must clearly differentiate the multiple network functions and services that conform the application. In this way, in Fig. 4 the architecture of the vOBU NetApp can be seen. By analyzing its architecture, we can conclude that the NetApp is formed by three different VNFs, namely, the vOBU itself, the data aggregator, and the OBU manager, which will be aggregated in one single NS.

Once the number of NSs and VNFs is decided, the information the NSDs should include has to be identified by taking into account the considerations detailed in Section 3.1.1. The following items are examples of the kind of requirements that have to be defined in the design phase:

- Packaging: The VNFs will be deployed as VMs, ready to be used in OSM.
- Internet connectivity needed: Yes.
- Hardware resources required: 2 CPUs, 1 GB RAM, 10 GB storage.
- Placement latency: 500 ms.

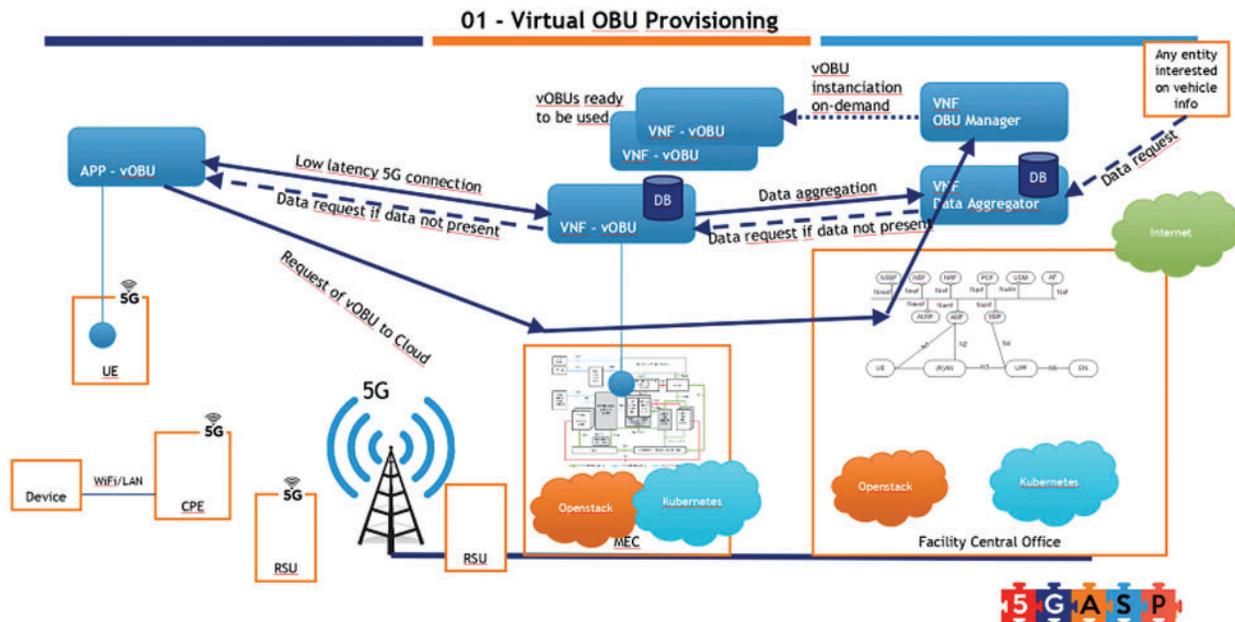


Figure 4: Virtual On-Board Unit (vOBU) NetApp

The last step regarding the design of the NS and VNFs is to consider the dependencies with other NetApps. In this case, this NetApp does not have any dependency with others, however, in more complex vertical use cases, the vOBU NetApp may need to cooperate with other NetApps, which should be considered when designing the use case.

In the design of the network requirements of the NetApps, the developer has to define how many network slices are needed and the characteristics of them. In this case, the vOBU will simply need one network slice. Next, it has to be decided the requirements that will include the NEST template that define the network slice on which the NetApp will be hosted. Among multiple parameters, below we present some of the most common ones:

- Area of service and region specification: Murcia (Spain).
- Isolation level: Virtual resources.
- V2X communication mode: Yes, with New Radio.
- Slice quality 5GPP 5QI: 9.
- Maximum packet loss: 1%.
- Supported UE velocity: Vehicular.

Finally, the last step of the design phase is the definition of the set of tests that must be performed in the 5GASP platform against the vOBU NetApp in order to ensure a valid functioning and, if successful, resulting in the certification of the NetApp. As the testing procedure and its insights are currently under design and development, here we will focus on simple tests, mainly focused in the infrastructure,

which will be used to validate the KPIs of the vOBU NetApp. To this end, a series of infrastructure tests will be defined to evaluate the metrics from which the KPIs are computed. These KPIs are (i) initial deployment time, (ii) transaction speed of the messages, (iii) Packet Loss Ratio (PLR) of the

messages exchanged between the OBU and the vOBU, and (iv) vOBU service downtime. These KPIs are further detailed in [Tab. 1](#).

Table 1: vOBU NetApp KPIs

KPI name	Metric indicator	KPI value	KPI unit
Initial time	Time needed to deploy the entity (vOBU) for first time	<5	minutes
Transaction speed	Each message sent from a OBU needs to be redirected to the vOBU	<500	ms
Packet loss ratio	Packets loss above packets sent	<1	%
Service downtime	Ratio of time the vOBU is not up and running	<1	%

5.2 VOBUNetApp Development

Once the design phase has finished, the development of the NetApp can start. It consists in developing a number of descriptors to reflect the information gathered in the previous step.

The first one is the NSD, which defines the vOBU network service; it can be seen in Code 1. As aforementioned, it includes the three VNFs that compose the network service and the network to which they are attached to. This network is already present in the descriptor as it is known that the 5GASP's Murcia facility offers it. However, as previously discussed, in case that the developer is unaware of the available networks of the test sites, and in the first stages of the project, a list of the networks ready to be used in each facility will be presented to the experimenters beforehand.

Code 1: vOBU NetAPP NSD

```

nsd: nsd-catalog:
  nsd:
    -constituent-vnfd:
      -member-vnf-index:'1'
      vnfd-id-ref: vOBU_mgmt_vnfd
      -member-vnf-index:'2'
      vnfd-id-ref: vOBU_vobu_vnfd
      -member-vnf-index:'3'
      vnfd-id-ref: vOBU_agg_vnfd
    description: vOBU NS prepared for OSM version 8.
    id: vOBU_nsd
    name: vOBU_nsd
    short-name: vOBU_nsd
    version: '1.0'
  vld:

```

(Continued)

Code 1: Continued

```

-id: red800
mgmt-network: true
name: Red800BigNAT
short-name: Red800BigNAT
type: ELAN
vim-network-name: Red800BigNAT
vnfd-connection-point-ref:
-member-vnf-index-ref:'1'
vnfd-connection-point-ref: eth0
vnfd-id-ref: vOBU_mgmt_vnfd
-member-vnf-index-ref:'2'
vnfd-connection-point-ref: eth0
vnfd-id-ref: vOBU_vobu_vnfd
-member-vnf-index-ref:'3'
vnfd-connection-point-ref: eth0
vnfd-id-ref: vOBU_agg_vnfd

```

Next, we present in Code 2 one of the VNFs that compose the NetApp, namely, the one corresponding to the vOBU entity itself. Here it is important to highlight some of the fields in relation to the design phase such as the interface that will be connected to the network defined in the NSD, the image used to host the VNF, and the resources required for that image. Here, the situation is similar to the network that is defined in the NSD, but with regard to the operative system image and the flavors. In this VNF, the image and VM-flavor defined are known in advance, as they are present in the Murcia test site, although a list of offered images and flavors in each facility will be provided beforehand as well. Another aspect to note in these descriptors is that they are designed and developed following the guidelines of OSM to be compatible with OSM Rel. EIGHT, as the Murcia test site in which the NetApp is intended to be hosted counts with an instance of OSM8. In the same line, the supported OSM releases in each testbed will be exposed to the experimenters, as well as any other available NFVO.

Now that the NSD and the VNFs are defined, we can move on to the NEST. This is a straightforward step, as the heavy work has been performed in the design phase, analyzing the network requirements of the NetApp. Therefore, the procedure is to map them to the NEST template. In [Tab. 2](#), the NEST template filled with the values discussed in the previous section is presented. The values are defined according to the guidelines of the GSM's GST (see Section 4.1.2). As in the previous case, a network slice able to fulfill these requirements (or almost identical) will be available in Murcia. In other case, the NEST would be selected from the list of predefined ones offered by the 5GASP portal.

Table 2: vOBU NetApp NEST

vOBU NetApp NEST	
Area of service	Spain
Area of service: Region specification	Murcia
V2X communication mode	Yes, New radio (NR)

(Continued)

Table 2: Continued

vOBU NetApp NEST	
Slice quality of service parameters: 3GPP 5QI	9
Maximum packet loss rate	1%
Supported device velocity	3: Vehicular: 10 km/h to 120 km/h

Finally, the last stage of the development is the implementation of the tests. As presented before, 5GASP considers three types of tests: Pre-provided infrastructure tests, custom test scripts, and custom test VNFs. Due to the early stages of the project, at the moment there are only some infrastructure tests developed specifically for certain NetApps, thus they cannot be considered as general pre-provided infrastructure tests. These currently available infrastructure tests have been made ready to be used with Jenkins and its Robot Framework¹. The latter is in charge of the test itself, which has been developed using Python, and the former's responsibility is to manage the testing pipeline and trigger the execution of the test. In this way, three different tests have been developed with the aim of evaluating three of the KPIs defined for the vOBU NetApp: The deploy time, the transaction speed, and the PLR. In the following, we present as an example one of them; specifically, the one that tests the transaction speed between the OBU and the vOBU. In first place, Code 3 shows the Robot test definition, which defines the test script in charge of obtaining the metric value and the condition to validate the test.

Code 2: vOBU NetApp's vOBU VNFD

```
vnfd-catalog:
  vnfd>:
    -connection-point:
      -name: eth0
      type: VPORT
    description: vOBU entity prepared for OSM version 8.
    id: vobu_vnfd
    mgmt-interface:
      cp: eth0
    name: vobu_vnfd
    short-name: vobu_vnfd
    vdu:
      -count: 1
      description: vobu_vnfd-VM
      id: vobu_vnfd-VM
      image: debian-base-vOBU-certs
      interface:
        -external-connection-point-ref: eth0
        name: eth0
        type: EXTERNAL
      virtual-interface:
```

(Continued)

¹<https://plugins.jenkins.io/robot/>

Code 2: Continued

```

bandwidth:'0'
type: PARAVIRT
vpci: 0000:00:0a.0
name: vobu_vnfd-VM
vm-flavor:
memory-mb:'1024'
storage-gb:'10'
vcpu-count:'2'
vendor: vOBU
version:'1.0'

```

Code 3: vOBU NetApp test definition in Robot

```

*** Settings ***
Library          TransactionSpeed.py
*** Test Cases ***
Testing the transaction speed between OBU and vOBU
    ${milliseconds}=    Transaction Speed
    Should Be Equal    ${milliseconds}    Less than 500

```

Secondly, Code 4 presents the Python script in charge of obtaining the average transaction time between the OBU and the vOBU. It is a simple idea that uses the ping command to obtain the values.

Code 4: vOBU NetApp test script

```

import paramiko, re
host = `1.1.1.1`
username = `jenkins-testing`
password = `password`
def transmission_speed ():
    client = paramiko.SSHClient ()
    client.set_missing_host_key_policy (paramiko.AutoAddPolicy())
try:
    client.connect (hostname=host, username=username,
password = password)
except:
    print (`[!] Cannot connect to the SSH Server`)
    Exit ()
stdin, stdout, stderr = client.exec_command (`ping-c 5 1.1.1.2`)
pingResult = stdout.read ().decode ()
regex = re.compile (r`(.*) = (.*)\/(.*)\/(.*)\/(.*) ms`)
result = regex.search (pingResult)
print (pingResult)

```

(Continued)

Code 4: Continued

```
if result:
    avgTimeTransmission = float (result.group(3))
    print (`AVG:", avgTimeTransmission)
    if avgTimeTransmission < 500:
        return `Less than 500 milliseconds"
    else:
        return `More than 500 milliseconds"
else:
    return `Not found"
if __name__ == `__main__":
    transmission_speed ()
```

In the future, these tests will be generalized to be offered as pre-defined tests in the facility test sites. Furthermore, more complex tests will be developed in the form of custom test scripts and custom test VNFs.

5.3 vOBU Netapp Onboarding

With the development completed, it is time to onboard the NetApp in the 5GASP platform. As explained previously, the NetApp consists of the three descriptors presented above. To enable the onboarding of them, 5GASP portal offers an interactive procedure for their uploading. Once done, the triplet is converted into a Service Deployment Order and the CI/CD pipeline is triggered. In this way, the procedure is as follows:

1. Uploading of the NSDs and VNFDs to the portal.
2. Uploading of the NEST representing the required network slice. Another possibility is that the NEST can be selected from a predefined list.
3. Selection of the NetApp KPIs to be tested in the infrastructure.
4. Uploading of the tests. Another possibility is that the tests can be selected from a predefined list.

With this guided process, the NetApp is finally onboarded on the 5GASP platform and ready to be checked and certified. If this process is successfully accomplished, the NetApp could be offered in the NetAppMarket to be safely used in 5G infrastructures. This process is currently under development and more details will be given in future works or project's documentation.

6 Conclusion

5GASP vision is to foster the use of NetApps within the 5G ecosystem by introducing a well-defined approach to design and develop this kind of in-network applications. This proposed methodology enables the automated and reproducible validation of NetApps across multiple facility test sites, including inter-domain scenarios. By doing so, the aim is to become an operational platform with strong industrial backup with the potential to be a reference ecosystem for validation and deployment of 5G experiments. In this work, the initial methodology for the design, development, testing, and validation of NetApps has been presented. For that purpose, the placement of the onboarding methodology within the 5GASP general architecture and workflows are described together with an initial design and dissection of the onboarding steps to be taken to submit a NetApp to the 5GASP

portal. In addition, an initial description of the design and development of innovative NetApps is provided by emphasizing the information that a novel 5G developer should take into account when working with this type of applications. Finally, the whole NetApp development-to-onboarding process has been shown in a tutorial fashion by means of a real use case. This example permits to better understand the 5GASP methodology and provides insights about the NetApp development process to interested audience. As future work, further iterations and enhancements to the methodology will permit the automation of the procedures, reducing the intervention and required knowledge from the 5G developers. Besides, we plan to onboard other functional NetApps under different vertical use cases in order to explore the flexibility and adaptability of the 5GASP platform.

Funding Statement: 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 Odín Solutions S.L., Región de Murcia (Spain); and by the European Commission under the 5GASP project (Gran No. 101016448).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] C. Benzaid and T. Taleb, “AI-Driven zero touch network and service management in 5G and beyond: Challenges and research directions,” *IEEE Network*, vol. 34, no. 2, pp. 186–194, 2020.
- [2] P. P. Ray and N. Kumar, “SDN/NFV architectures for edge-cloud oriented IoT: A systematic review,” *Computer Communications*, vol. 169, pp. 129–153, 2021.
- [3] R. Rosa and C. Rothenberg, “Experiences in IETF-bMWG: Towards a methodology for VNF benchmarking automation,” in *Anais do VII Workshop Pré-IETF (WPIETF2020)*, Porto Alegre, RS, Brazil, pp. 43–56.
- [4] J. Gallego-Madrid, R. Sanchez-Iborra and A. Skarmeta, “Security and trust in the integration of network functions within the 5G architecture: The 5GASP project,” in *MobiSec 2021: The 5th Int. Symp. on Mobile Internet Security*, Jeju Island, South Korea, 2021, In press.
- [5] C. Tranoris and S. Denazis, “A workflow for onboarding verticals on 5G/NFV experimental network facility,” in *IEEE Int. Conf. on Communications Workshops (ICC Workshops)*, Dublin, Ireland, pp. 1–5, 2020.
- [6] 5GASP Project, “D2.1: Architecture, model entities specification and design,” 2021. [Online]. Available: [https://5gasp.eu/assets/documents/deliverables/D2.1 Architecture, Model Entities Specification and Design.pdf](https://5gasp.eu/assets/documents/deliverables/D2.1%20Architecture,%20Model%20Entities%20Specification%20and%20Design.pdf).
- [7] A. A. Barakabitze, A. Ahmad, R. Mijumbi and A. Hines, “5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges,” *Computer Networks*, vol. 167, pp. 106984, 2020.
- [8] M. Zhao, F. Le Gall, P. Cousin, R. Vilalta, R. Muñoz *et al.*, “Verification and validation framework for 5G network services and apps,” in *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Berlin, Germany, pp. 321–326, 2017.
- [9] ETSI OSM, “VNF onboarding guidelines,” 2019. <https://osm.etsi.org/docs/vnf-onboarding-guidelines/> (accessed Jun. 29, 2021).
- [10] R. Ali, Y. Bin Zikria, A. K. Bashir, S. Garg, and H. S. Kim, “URLLC for 5G and beyond: Requirements, enabling incumbent technologies and network intelligence,” *IEEE Access*, vol. 9, pp. 67064–67095, 2021.
- [11] GSMA, “Generic network slice template,” 2020. [Online]. Available: <https://www.gsma.com/newsroom/wp-content/uploads/NG.116-v4.0-2.pdf>.
- [12] J. Santa, P. J. Fernández, J. Ortiz, R. Sanchez-Iborra and A. F. Skarmeta, “SURROGATES: Virtual OBUs to foster 5G vehicular services,” *Electronics*, vol. 8, no. 2, pp. 117, 2019.