# Implementing a holistic approach to facilitate the onboarding, deployment and validation of NetApps


Kostis Trantzas, Christos Tranoris, Spyros Denazis
*University of Patras*
Patras, Greece
ktrantzas@ece.upatras.gr
tranoris@ece.upatras.gr
sdena@upatras.gr

Rafael Direito, Diogo Gomes
*Instituto de Telecomunicações*
*Universidade de Aveiro*
Aveiro, Portugal
rdireito@av.it.pt
dgomes@av.it.pt

Jorge Gallego-Madrid, Ana Hermosilla, Antonio Skarmeta
*Odin Solutions*
Murcia, Spain
jgallego@odins.es
ahermosilla@odins.es
skarmeta@odins.es



***Abstract*—The 5G system promises to deliver an environment for applications and markets that can easily capitalize on the new services and advanced interfaces that the 5G developments introduce. Recently, in the context of European research and innovation actions, the term NetApp was introduced. A NetApp, in the context of the 5G System, is defined as a set of services that provide certain functionalities to the verticals and their associated use cases. While there is no specific standard for describing the NetApp concept, the ETSI NFV model is adopted to describe the delivery and deployment of NetApps. This work presents an approach for automating the deployment and testing of NetApps in the context of the 5G system. The envisaged process focuses on innovations related to the operation of experiments and tests across several domains, providing software support tools for Continuous Integration and Continuous Deployment (CI/CD) of NetApps and their artifacts in a secure and trusted environment.**



***Keywords—5G, NetApps, CI/CD pipeline, multi-domain, automated deployment and validation***


## I. Introduction

The EC Innovation Action of Horizon 2020 5G-PPP Phase 3b ICT-41 projects coined the term *NetApp* to describe specific Virtual Network Function (VNF)-based solutions tailored to verticals that would like to utilize the 5G system. The challenge of defining NetApps today is carried out by 9 ICT-41 projects [1]. Specifically, they provide enhanced experimentation infrastructures on top of which third-party experimenters, Small and Medium-size Enterprises (SMEs), or any service provider and target vertical users have the opportunity to test applications in an integrated, open, cooperative, and fully featured network platform running across multiple domains, where needed, simultaneously tailored to specific vertical use cases (Enhanced Media, Augmenting Reality, PPDR, eHealth, Smart Cities, etc.). While there is no specific standard for describing the NetApp concept, the ETSI Network Function Virtualization (NFV) model is adopted to describe the delivery and deployment of NetApps.

In this way, as a continuation of [2], and contextualized in the H2020 5GASP Project (whose aim is to ease the 5G development process for SMEs), this paper presents a detailed methodology that facilitates the whole NetApp creation process, not only focusing on the development and deployment per se, but also on the inclusion of tests that automatically verify the accurate functioning of the NetApp within the 5G infrastructure. Additionally, this methodology also includes a novel onboarding process that automates all these procedures, therefore simplifying their executions. Thus, when a NetApp is onboarded, an automated process takes place in order to perform the required verifications. For that purpose, the NetApp is deployed in a restricted and controlled scenario, where several infrastructure, functional and developer-customized tests are also performed. To do so, specific artifacts are deployed, which are responsible for conducting the corresponding verification tasks and gathering the results. Those are then returned to the platform, therefore indicating the feasibility and correctness of the onboarded NetApp. All the above processes are executed automatically when the NetApp is onboarded, thus avoiding the need for performing manual operations.

In this paper, we first overview the work introduced by notable related research projects and prominent standardization bodies (Section II). Second, we present the current 5GASP framework that facilitates the envisaged CI/CD process (Section III). Following this, an explicit reference to each entity of the proposed unified model is attempted (Section IV, V, VI) alongside our implementation and actuation of the model (Section VII) that validates its credibility. Lastly, we end with a conclusion and future work discussion (Section VI).

## II. Related Work

### *A. EU Projects*

Several European projects tackled the validation of 5G NetApps, with 5GTango and 5G-VITAL being the two most relevant ones.

5GTango introduced a Validation and Verification (V&V) platform, which offered advanced mechanisms to validate VNFs and Network Services (NSs). The main components of this platform are: (i) the V&V Gateway, which exposes an API to interact with this service, (ii) the Test Invoker, responsible for test configuration and scheduling, (iii) the Test Repository, which stores the Testing Descriptors, (iv) the Package Validator, for the validation of the submitted packages (descriptors, for instance), (v) the Testing Engine, that is responsible for deploying the tests, (vi) the Test Results Repository, that stores the outputs of the testing phase, and (vii) the Test Analysis Component, responsible for the results processing and visualization [3].

The verification and validation process is composed of three stages: (i) structural validation of the descriptor, (ii) functional testing, and (iii) performance evaluation.

During the structural validation of the descriptors, four tests are performed: (i) syntax testing, where the descriptor is validated against the 5GTANGO schemas, (ii) integrity testing, which checks if all tags of the descriptor have the correct values, (iii) topology testing, where the network topology of the NS is validated, and (iv) custom rules testing, that validates the descriptor against custom rules defined by the developer, and provided via a YAML file.

If the descriptors are structurally correct, the second testing phase may commence. Along with the onboarding of the VNFs and NSs, the developer also onboards some test plans. These are YAML files with the definition of the tests that will be performed. The second phase starts with the deployment of the VNFs and NSs in an emulated environment. Then, the test plans are executed.

After the test execution, the third and final phase occurs, where several results and metrics are gathered and processed. Following, the outputs of the V&V process are displayed via a graphical interface. These results could be a simple binary "pass" or "fail" or more complex if the tests aimed to collect metrics [4].

VITAL-5G project employs a different approach, as it divides the testing and validation methodology into two folds: the first one focusing on the technology validation itself (steps, procedures, and framework) to test and evaluate the functionalities and performance; and the second one focused on the business validation methodology, to exploit the business potential of the platform attracting third-party experimenters [5]. Regarding technology validation, the idea is to offer *Testing as a Service* (TaaS) for automated execution and evaluation, providing it through the VITAL-5G portal. There, the experimenters can define the required experiments, the validation parameters, as well as the specific test cases of interest, and then obtain the validation results and report. One of the most relevant and powerful tools of the VITAL-5G Testing and Validation procedure is the creation of a Testcase template that provides the tests' necessary context and information. This template is a document that defines all the required information to conduct the experiments, register the results, and monitor the KPIs, e.g. their components and configuration, measurements, key use-case requirements, etc.

Concerning the VITAL-5G's End-to-End (E2E) Testing and Validation procedure, it is composed of the following steps: (i) Experiment Design, which includes the detailed definition of the experiments to be executed via the fulfillment of the Testcase template; (ii) Experiment Execution, where the information required is collected by the monitoring and metric/data collection tools according to the measurement points; (iii) Results Evaluation and Service Validation, where the performance is evaluated and the service validated, according to the target KPIs; and (iv) Experiment Finalization, providing feedback to the experimenters across the portal and the results report.

Finally, regarding business validation, VITAL-5G uses Lean Startup (LS) methodology (a methodology based on testing with real customers) to collect business-related feedback from the experimenters. The pursued feedback can be: (i) Use Case Business Related Feedback, depending on which use case the actor belongs to; (ii) Platform User Feedback, depending on the role of the platform's actor; and (iii) Third-Party Experimenter Feedback, as VITAL-5G platform is offered to third-parties. To obtain feedback from the above actors, they use Business Validation Questionnaires, which are created based on the analysis of the previous user groups.

### B. Standards Development Organizations

Tele Management Forum (TM Forum) is a global industry confederation actively working on evolving current Operations/Business Support Systems (OSS/BSS). Its sought solutions involve the integration with existing standards-driven architectural frameworks and the seamless consumption by verticals. In these terms, TM Forum's major contribution is the introduction of the Open Digital Architecture (ODA) [6], which presents slice management architectures and use cases derived from various catalysts, along with a set of Open APIs that facilitate the latter. Also, it incorporates Network as a Service (NaaS) Component Suite [7], which covers the needed operations to provide the required functionality across interworking operational domains.

GSMA is a trade body representing the interests of mobile operators worldwide. Its work mainly focuses on translating vertical industry requirements into network characteristics. On this notion, GSMA introduced the Generic Slice Template (GST) in an attempt to narrow down the gap between network service customers and vendors. First, it offers a universally accepted model among service providers that simultaneously enables the customers to express their desired service requirements. Second, similar network slice templates (NESTs) are shared between all service vendors and operators leading to worldwide adoption, which eventually promotes the seamless interoperability between restricted private networks with wider public networks.

## III. The 5GASP Framework

The main goal of the project, as described in the Introduction, is supported by an onboarding and experiment management portal that serves as a single-entry point for relevant actors. Its task is to supply a user-friendly User Interface (UI) that would facilitate the seamless onboarding, design and close to zero-touch provision of both NetApps and their respective testing suites. The portal solution is part of 5GASP NetApp Onboarding and Deployment Services (NODS), which also integrates a Service and Network Orchestrator that coordinates the deployments on the underlying facilities, and an interaction point with the 5GASP CI/CD Manager, charged with the execution of the CI/CD pipeline. The 5GASP NODS is based on the open-source project Openslice[1].

The 5GASP project comprises six independent testbeds that, in the scope of the project, are offered as a seemingly unified testbed. To achieve this, we rely on NetOr [8], a network orchestrator that provides the instantiation and orchestration of Wireguard [2] tunnels between different facilities, thus interconnecting them. NetOr onboards a VNF, comprised of a Wireguard peer, in each facility's NFV Orchestrator (NFVO), which will then be employed to assemble a mesh network between the facilities. This mesh enables transparent communication between different testbeds, making it possible for the 5GASP project to support inter-domain scenarios. Although, each testbed can also be offered as a standalone since all the 5GASP testbeds are comprised of an NFVO, a Test Execution Engine (TEE), and

[1] openslice.io

[2] www.wireguard.com

a Local Test Repository (LTR), which are the key components necessary for the orchestration, deployment, and validation of a NetApp.

To onboard, i.e., to upload, the artifacts that compose a NetApp, a triplet model is adopted within the project's scope, as introduced in [2]. In short, this triplet is formed by: (i) the NFV artifacts that define the NetApp itself, i.e., NSD(s) and VNFDs; (ii) the NEST descriptor with network requirements and characteristics that specify the network slice to be deployed; and (iii) the Test Descriptor that defines the set of tests to be executed to ensure the correct functioning of the NetApp. These tests comprise two different types: (i) infrastructure tests available in the facilities to evaluate the proper operation of the NetApp in the infrastructure; and (ii) developer-defined tests, whose aim is to validate the operation and functioning of the NetApp itself. Specifically, the latter can be defined as custom scripts that are executed in the 5GASP platform, or as custom test VNFs, which are deployed together with the NetApp. Eventually, the overall triplet is required to deploy a NetApp.

Having all the artifacts onboarded to NODS, the developers can now trigger the deployment and validation of their NetApps. This process initially involves selecting the testbed, or testbeds, where the NetApps shall be deployed. A complete list of the capabilities of each testbed will be offered to guide the developers through this process. After the testbed has been chosen, the NODS will interact with that testbed's NFVO to deploy the NetApp, and after being instantiated, it can be validated eventually by the 5GASP Validation Service. This process heavily relies on CI/CD Agents deployed throughout the testbeds, which will perform the validation tasks required by the CI/CD Manager, responsible for coordinating the validation of a NetApp. After these tasks are executed, the Agents will collect the validation outcomes and forward them to the CI/CD Manager, which will post them back in NODS, rendering them available for the NetApp developers.

That said, the following sections elaborate on how every entity of the triplet is being conceived and outlined in the score of 5GASP.

## IV. NFV Artifact Onboarding

As part of the overall portal solution, a dedicated NFV artifact management User Interface (UI) is employed that allows its actors to onboard and manage the NetApp packages, i.e. VNF/NSD artifacts. The provided package descriptors undergo a pre-flight validation process to guarantee the sanity of their content, depending on their declared versioning. Currently, an endeavor to integrate supplementary external validation services is made, with the scope to support even more complex validation processes. An outline of these enriched processes would be syntax checking, packaging format confirmation, file and reference consistency, among others. Subsequently, the validated packages are assembled in corresponding catalogs.

Nonetheless, to support the proposed unified model adoption approach, the onboarded artifacts are mapped towards TMF's Resource Facing Service Specifications (RFSSs) expressing the resource aspects of the NetApps with their corresponding requirements. This approach simultaneously delineates the NFV deployment characteristics, as well as any additional user-oriented information expressed with a standardized set of characteristics. Also, this common representation enables multiple NetApp stitching into a single NetApp bundle, but also guarantees compliance with the rest of the onboarding entities, as they are also expressed under the same standard.

## V. Network Slice Selection

Each testbed offers a set of network slices that acts as the foundation for the NetApps. During the onboarding of the NetApp, we distinguish two options regarding the selecting of a hosting network slice: (i) the developer specifies the accommodating testbed manually through a list of explicitly described network capabilities, or (ii) the 5GASP platform automatically appoints the NetApp to a target testbed, based on more abstract requirements input from the developer.

That said, to achieve the desired automation of the latter choice, the hosting network slice requirements need to be defined likewise in a uniform and standardized way. Hence, these requirements are fully aligned with GST properties. Each designated testbed provides the range of network requirements it supports in form of NESTs, an example of which is depicted in Table 1.

Table 1. Network capabilities expressed in NEST

| ***Example NEST*** | |
|---|---|
| Area of service | GR |
| Area of service: Region specification | Patras |
| Downlink maximum throughput per UE | 10000 kbps |
| Uplink maximum throughput per UE | 100000 kbps |
| Isolation level | Virtual resources isolation |
| Slice quality of service parameters: 3GPP 5QI | 69 |
| Supported device velocity | Pedestrian |
| User data access | Termination in the private network |

## VI. Testing Definition

As mentioned in Sec. III, Test Descriptors can involve two different kinds of tests: tests included as scripts (code that will be executed elsewhere) and Test VNFs. The latter are VNFs whose sole purpose is to perform specific tests on the NetApp, not offering any other functionality. To create and configure those VNF Tests, a Test Descriptor template is proposed (in YAML format). This descriptor is composed of two sections: (i) *test_info*, which includes general information about the test, such as its relations with the NetApp, its NS, and the testbed where it will be deployed; and (ii) *test_phases*, which defines the necessary steps to deploy, execute and validate the test. An example of such file is shown below (Fig. 1).

In the *setup* phase, two sections are shown: *deployments*, where the deployment of the required custom Test VNFs is indicated, and *testcases*. The latter includes some predefined tests, i.e., infrastructure-related tests, and custom tests created by the developer to test specific functionality of the NetApp.

Currently, apart from the Test Descriptors that are predefined and publicly available to developers, one may leverage the dedicated test designing UI to supply its test procedures to 5GASP's CI/CD pipeline. Specific runtime parameters are populated during deployment and expressed in the TMF Service Test model.

```yaml
# Test general information and relations with NetApp, NS and testbed
test_info:
  test_id: 1
  netapp_id: vOBU
  network_service_id: vOBU_ns
  testbed_id: gaia5G
  description: Infrastructure tests for vOBU NetApp
  startTime: startTime
  endTime: endTime

# Test definition in three phases
test_phases:
  # Test setup: deploy the required VNFs and define test cases
  setup:
    deployments:
      # Preparation the deployment of custom test VNFs
      - deployment_id: 1
        name: vOBU_traffic_generator_vnf_deployment
        descriptor: vOBU_traffic_generator_vnfd
        id: none # deployment information filled by NODS
      - deployment_id: 2
        name: vOBU_service_consumer_vnf_deployment
        descriptor: vOBU_service_consumer_vnfd
        id: none
      - deployment_id: 3
        name: vOBU_vnf_testing
        id: none
        descriptor: vOBU_testing_nsd
    # Definition of the test cases
    testcases:
      # Predefined tests
      - testcase_id: 1
        type: predefined
        scope: infrastructure
        name: infrastructure_kpi
        kpi: deployment_time
        parameters:
          - key: host1_ip
            value: 10.0.0.1
          - key: host2_ip
            value: 10.0.0.2
      # Custom test vnfs
      - testcase_id: 5
        type: custom_vnfs
        scope: operational
        name: generator_consumer_vnfs
        vnfs_deploy_id: [1,2]
  # Execution of the testcases
  execution:
    - batch_id: 1
      scope: infrastructure
      executions:
        - execution_id: 1
          name: infrastructure_tests
          testcase_ids: [1]
          instances: 1
          start_delay: 0
    - batch_id: 2
      scope: operational
      executions:
      - execution_id: 1
        name: custom_vnfs
        testcase_ids: [2]
        instances: 1
        start_delay: 60
  # Validation of the test cases outputs
  validation:
    - validation_id: 1
      execution_id: 5
      file: "vnf_log.txt"
```

Fig. 1. Test Descriptor template

## VII. 5GASP Prototype Implementation

As previously stated, to deploy and validate a NetApp, the developer has to submit a bundle composed of (i) a Network Slice Template, (ii) the NetApp artifacts, and (iii) the testing artifacts. Thus, we can divide the deployment and validation process into three main stages: (i) the Network Slice deployment, (ii) the NetApp deployment, and (iii) the testing and validation of the NetApp. These are highly coupled and executed sequentially. Thus, one stage cannot be executed if the previous fails. The first two stages are focused on having the NetApp deployed in 5GASP testbeds. The NODS portal interacts with the testbed's Network Slice Management Function (NSMF) to orchestrate a Network Slice based on the provided template. This slice will host the NetApp, deployed by the testbed's NFVO after a request from NODS. Then, having the NetApp deployed, the NODS will trigger a new validation process, starting this pipeline's third and last stage.

Throughout this section, we will further elaborate on each of the phases that comprise the described pipeline.

### A. *Network slice deployment*

As the orchestration scheme determines, the hosting Network Slice is deployed first. Specifically, the Service Orchestrator that resides within NODS transforms the provided high-level requirements into actual network requirements, based on the selected NEST. Eventually, certain NFV artifacts that facilitate the extracted network requirements are employed and commissioned for orchestration to the respective testbed's NFVO.

The nominal scenario would be for the designated testbed to provide the overall 5G Core functionality, but depending on NEST's properties, such as requested latency, additional deployment schemes may also be considered. For instance, a low latency preference may translate to a discrete User Place Function (UPF) placement outside the testbed's deployed 5G Core (Edge scenarios). In that case, the discrete UPF instance registers itself into the 5G Core's Session Management Function (SMF) and renders its data path available for traffic, thus enabling NetApp's deployment.

### B. *NetApp deployment*

The NetApp can be deployed only after the instantiation of 5G Core to its full functionality. The latter is assumed after specific 5G Core properties are exposed, e.g., Network Exposure Function (NEF), UPF's data path., laying the foundation for the NetApp.

Currently, the NetApp deploys in a manually selected testbed. Our envisaged goal is that the deployment selection should be performed automatically. As formerly said, NetApp's resource aspects are loosely described, despite being represented through a set of standardized characteristics. Therefore, the developer's requirements can also be expressed in NESTs, or even elected from a pool of supported NESTs that will eventually translate to specific hosting Network Slices. Either way, once the deployment process instantiates, NODS expects specific deployment parameters (see Section V) to be populated, e.g. VNF IP addresses, to redirect the flow to the next component, namely the CI/CD Manager.

### C. *CI/CD pipeline triggering*

The CI/CD Pipeline is triggered by the NODS with the submission of a TMF 653 Service Test payload to the CI/CD

Manager, as presented in Fig. 2. The CI/CD Manager is the central entity that coordinates all the validation jobs, creating and distributing validation jobs to the CI/CD Agents deployed in the testbeds. To boost the interoperability of the CI/CD Manager, it implements the TMF 653 standard, which is the base of all interactions with the NODS.

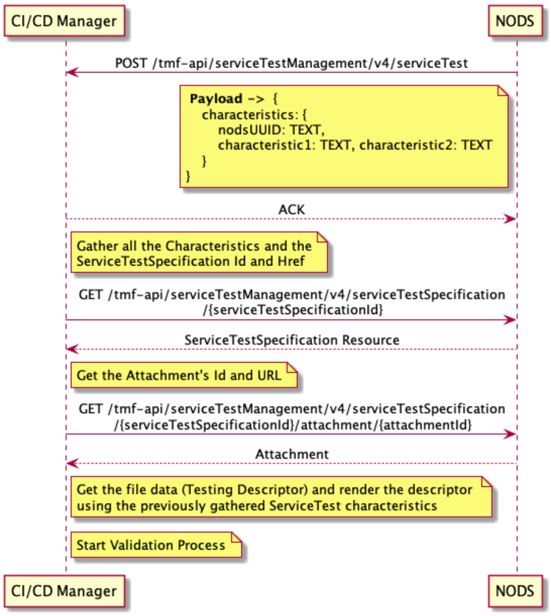



Fig. 2. Interaction between NODS and the CI/CD Manager

The Service Test payload, submitted by the NODS, comprises all the information needed to start a new validation job. Some information is directly specified in the initial payload, while other is referenced through URLs, from where the information can be gathered. For instance, the initial payload is composed of a collection of test characteristics that solely by themselves are not sufficient to perform a validation task. These characteristics are defined by the NODS and comprise information such as the IPs of a NetApp's VNFs. Without the onboarded Testing Descriptor, it is impossible to make any sense of the submitted characteristics. Thus, the initial payload also includes an URL from which the Testing Descriptor can be obtained. The CI/CD Manager uses this information to obtain the Testing Descriptor, which will then be rendered using the Service Test characteristics. Only after this process, the CI/CD Manager will be able to generate a validation pipeline configuration, which guides the entire testing process.

It is also worth mentioning that the Service Test Payload may also include URLs to gather developer-defined tests if the NetApp developer onboarded its own tests. If so, the CI/CD Manager will gather these tests and make them available to the CI/CD Agents so that they can eventually execute them.

After rendering the Testing Descriptor and gathering the developer-defined tests, the CI/CD Manager will assemble a validation pipeline. This process relies on the concept of Pipeline as Code since the CI/CD Manager dynamically generates a pipeline configuration file, which will then be submitted to the CI/CD Agents responsible for performing the validation tasks. The CI/CD Agents, enabled by Jenkins, offer a straightforward API to submit pipeline configuration files, which heavily simplifies the creation and execution of dynamic validation pipelines.

Once a new validation pipeline is onboarded to a CI/CD Agent, the latter will set up an untainted testing environment and gather all the required tests. As previously stated, two types of tests can be conducted. To gather the pre-defined tests, the CI/CD Agents rely on LTRs available on all testbeds. The LTRs are FTP Servers that store the pre-defined tests that can be performed on the testbed where they exist. On the other hand, the developer-defined tests will be gathered directly from the CI/CD Manager, which exposes them through a specific API endpoint. All these tests are obtained and locally stored in the newly created testing environment, from where they will be executed.

During the tests' execution, the CI/CD Agents will assemble their outputs and results. This task is straightforward when using Robot Framework[3] defined tests, which is the case in 5GASP. When executing Robot Framework tests, Robot transparently creates log and report files, using HyperText Markup Language (HTML). These files enable a simple understanding of the outputs of a test and make it possible for the NetApp developers to explore the results of their tests in detail.

When all tests have been performed, the CI/CD Agent will create a test report composed of all the logs and reports of the executed tests. Then it will forward this report to the CI/CD Manager, that, alongside NODS, is responsible for making it available to the NetApp developers.

Fig. 3 presents the interactions between the CI/CD Manager and a CI/CD Agent during the execution of a validation pipeline.

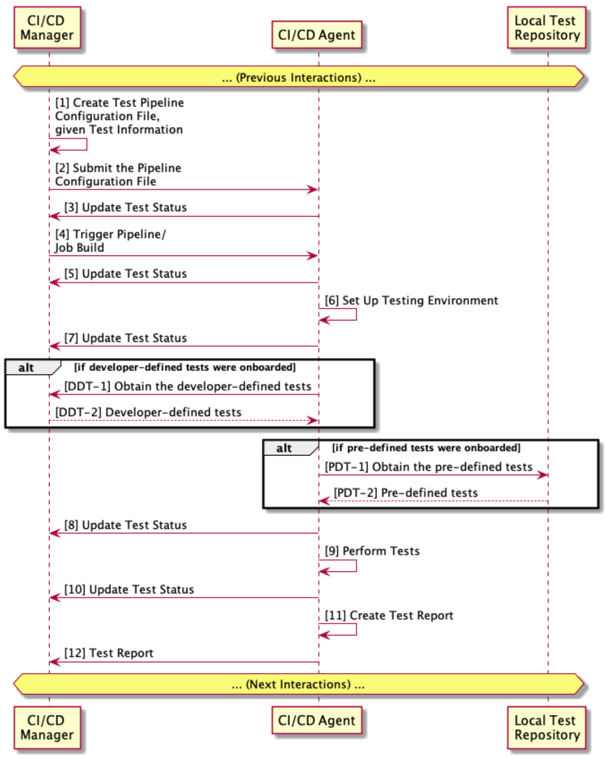



Fig. 3. Interaction between the CI/CD Manager and a CI/CD Agent

---

[3] https://www.robotframework.org/

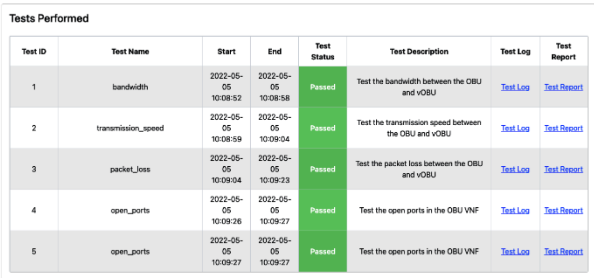
### D. *CI/CD results*

Such as previously stated, after the execution of the tests, the CI/CD Agents will bundle all test outcomes and results into a test report. This report is composed of HTML reports and log files that provide information regarding the tests' execution process and offer significant feedback on why a test may have failed. Besides these files, by performing tests implemented through the Robot Framework, another report file can be gathered. This file provides additional information regarding the tests, which is offered in Extensible Markup Language (XML). The CI/CD Manager will use this file to further elaborate on the test outcomes and results.

After the test report is submitted to the CI/CD Manager, this component will create a visual representation of the validation phase outcomes. This visualization will be provided by the Test Results Visualization Dashboard (TRVD), and it will encompass the following information: (i) base test information, such as the testbed where the tests were performed and the timestamp of the test's execution, (ii) information related to the results of each of the phases of the validation process (testing environment set up, tests gathering, tests execution, etc.), and (iii) the results of the executed tests, offered as HTML TRVD embedded web pages.

Then, it is required to provide all this information to the end NetApp developers. To do so, the CI/CD Manager generates a URL for the aforementioned visualization and patches the initial TMF 653 Service Test with this URL. This way, the NODS will be able to present this information to the NetApp developers, redirecting them to the TRVD to consult the results of the validation of their NetApp.

Fig. 4 presents a portion of the TRVD's Graphical User Interface (GUI), from which it is possible to observe how effortless it is for the NetApp developers to get the Robot-generated report and log files.



Tests Performed

| Test ID | Test Name | Start | End | Test Status | Test Description | Test Log | Test Report |
|---|---|---|---|---|---|---|---|
| 1 | bandwidth | 2022-05-05 10:08:52 | 2022-05-05 10:08:58 | Passed | Test the bandwidth between the OBU and vOBU | Test Log | Test Report |
| 2 | transmission_speed | 2022-05-05 10:08:59 | 2022-05-05 10:09:04 | Passed | Test the transmission speed between the OBU and vOBU | Test Log | Test Report |
| 3 | packet_loss | 2022-05-05 10:09:04 | 2022-05-05 10:09:23 | Passed | Test the packet loss between the OBU and vOBU | Test Log | Test Report |
| 4 | open_ports | 2022-05-05 10:09:26 | 2022-05-05 10:09:27 | Passed | Test the open ports in the OBU VNF | Test Log | Test Report |
| 5 | open_ports | 2022-05-05 10:09:27 | 2022-05-05 10:09:27 | Passed | Test the open ports in the OBU VNF | Test Log | Test Report |

Fig. 4. Portion of the TRVD's GUI [9]

## VIII. CONCLUSION

The design, development, deployment, and verification of a NetApp is not a simple process, especially if those NetApps are deployed in heterogeneous and multidomain scenarios like the one proposed by 5GASP. In this way, the creation of an automated process responsible for deploying, performing the relevant validation tasks, gathering its results, and deciding if the NetApp complies with the requirements improves and eases the process significantly. Moreover, the offering of the 5GASP framework as a conjunction of different 5G facilities distributed across several countries with a single point of entry and automated connectivity management is also a contribution, as it provides flexibility and simplicity in the design and deployment of the components that conform a NetApp.

However, the study does not conclude here, as future work involves further developments that are equally relevant and noteworthy, for example, the monitoring of the test execution to show details of the conducting in real-time. Another considered enhancement is the addition of the ability to automatically select the hosting testbed (that is, the place on which the NetApp will be deployed) based on the NetApp Network Slice's characteristics and requirements. Yet, even in the absence of these planned enhancements, the proposed solution at present already provides a fully automated CI/CD process that can be used by SMEs and relevant actors to validate NetApps as close to the production environment as possible.


## ACKNOWLEDGMENT

This work is supported by the H2020 European Project 5GASP (grant agreement No. 101016448); by Fundación Séneca — Agencia de Ciencia y Tecnología de la Región de Murcia — under the FPI Grant 21429/FPI/20, and by the Spanish Ministry of Science and Innovation through the Industrial PhD grant DIN2019-010827.


## REFERENCES


[1] 5GPPP, "Phase 3.6: 5G Innovations and beyond 5G," [Online]. Available: https://5g-ppp.eu/5g-ppp-phase-3-6-projects/. [Accessed June 2022].

[2] K. Trantzas *et al*, "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.

[3] P. Twamley *et al*, "5GTANGO: An Approach for Testing NFV Deployments," in *2018 European Conference on Networks and Communications (EuCNC)*, 2018.

[4] M. Peuster et al, "Introducing Automated Verification and Validation for Virtualized Network Functions and Services," IEEE Communications Magazine, vol. 57, no. 5, pp. 96-102, 2019.

[5] VITAL-5G, "D1.4: Testing and validation methodology," [Online]. Available: https://www.vital5g.eu/wp-content/uploads/2022/01/VITAL5G-D1.4-Testing-and-validation-methodology_Final.pdf. [Accessed June 2022].

[6] T. Forum, "GB999 – ODA Production Implementation Guidelines," 2020.

[7] T. Forum, "TMF909 API Suite Specification for NaaS v3.0.1," 2020.

[8] J. Fonseca et al, "Dynamic Interdomain Network Slicing for Verticals in the 5Growth Project," in 2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2021.

[9] R. Direito, D. Gomes and R. L. Aguiar, "Towards a Fully Automated System for Testing and Validating NetApps," in *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*, 2022.