EXPERIMENTATION AND VALIDATION OPENNESS FOR LONGTERM EVOLUTION OF VERTICAL INDUSTRIES IN 5G ERA AND BEYOND

[H2020 - Grant Agreement No.101016608]

Deliverable D5.2

NetApps Validation and onboarding to Open Repository (intermediate)

Editor G. Makropoulos (NCSRD)

Contributors (NCSRD), (TID), (ATOS), (INTRA), (COS), (MAG), (IMM), (GMI), (INF), (CAF), (ININ), (ZORT), (UMA), (8BELLS), (FOG), (IQBT), (PAL), UML)

Version 1.0

Date August 31st, 2022

Distribution PUBLIC (PU)

DISCLAIMER

This document contains information, which is proprietary to the EVOLVED-5G ("Experimentation and Validation Openness for Longterm evolution of Vertical inDustries in 5G era and beyond) Consortium that is subject to the rights and obligations and to the terms and conditions applicable to the Grant Agreement number: 101016608. The action of the EVOLVED-5G Consortium is funded by the European Commission.

Neither this document nor the information contained herein shall be used, copied, duplicated, reproduced, modified or communicated by any means to any third party, in whole or in parts, except with prior written consent of the EVOLVED-5G Consortium. In such case, an acknowledgement of the authors of the document and all applicable portions of the copyright notice must be clearly referenced. In the event of infringement, the consortium reserves the right to take any legal action it deems appropriate.

This document reflects only the authors' view and does necessarily reflect the view of the European Commission. Neither the EVOLVED-5G Consortium as a whole, nor a certain party of the EVOLVED-5G Consortium warrant that the information contained in this document is suitable for use, nor that the use of the information is accurate or free from risk, and accepts no liability for loss or damage suffered by any person using this information.

The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

REVISION HISTORY

Revision	*Date*	*Responsible*	*Comment*
0.1	*June, 1, 2022*	*NCSRD*	*1ST Draft*
0.2	*June, 7, 2022*	*NCSRD*	*2ND Draft*
0.3	*June, 20, 2022*	*NCSRD*	*3rd Draft*
0.4	*July, 1, 2022*	*NCSRD, TID, SMEs,*	*4TH Draft*
0.5	*July, 15, 2022*	*NCSRD, TID, INTRA, FOG,*	*Finalisation of contributions*
0.8	*July, 20, 2022*	*NCSRD*	*Final edits*
0.9	*July, 27, 2022*	*NCSRD, TID, UMA*	*Final draft for internal review*
1.0	*August, 26, 2022*	*NCSRD, TID*	*Final version*

LIST OF AUTHORS

Partner ACRONYM	*Partner FULL NAME*	*Name & Surname*
NCSRD	*NATIONAL CENTER FOR SCIENTIFIC RESEARCH "DEMOKRITOS"*	*George Makropoulos* *Dimitrios Fragkos* *Anastasios Gogos* *Harilaos Koumaras*
TID	*TELEFONICA INVESTIGACIÓN Y DESARROLLO*	*Javier Garcia* *David Artunedo* *Alejandro Molina*
INTRA	*NETCOMPANY INTRASOFT SA*	*Angela Dimitriou*
IMM	*IMMERSION*	*Charles Bailly*
GMI	*GMI AERO*	*Marc-Olivier Sauer* *George Kanterakis*
INF	*INFOLYSIS P.C*	*Christos Sakkas* *Theoni Dounia* *George Theodoropoulos* *Antonios Varkas*
CAFA	*CAFA TECH OU*	*Märten Rannu*
ININ	*INTERNET INSTITUTE, COMMUNICATIONS SOLUTIONS AND CONSULTING LTD*	*Luka Korsic* *Jaka Cijan* *Rudolf Susnik*
ZORTENET	*ZORTENET P.C.*	*Akis Kourtis, Andreas Oikonomakis, George Xilouris*
UMA	*UNIVERSIDAD DE MÁLAGA*	*Bruno García* *Rafael López*
8BELLS	*EIGHT BELLS LTD*	*Ioannis Margaritis* *George Avdikos*
FOGUS	*FOGUS INNOVATIONS & SERVICES*	*Dimitris Tsolkas* *Anastasios-Stavros Charismiadis* *Nikos Passas*
IQBT	*INQBIT INNOVATIONS SRL*	*Stylianou Ioannis, Eleni Argyriou*
PAL	*PAL ROBOTICS SL*	*Thomas Peyrucain*
UML	*UNMANNED SYSTEMS LIMITED*	*Raul de Roa, Sergi Grau Moya, Bianca Bendris*

GLOSSARY

Abbreviations/Acronym	*Description*
3GPP	3rd Generation Partnership Project
5QI	5G Quality of Service Identifier
AWS	Amazon Web Services
API	Application Programming Interface
CAPIF	Common API Framework
CI/CD	Continuous Integration / Continuous Development
DC-KPI	Data Consistency KPI
DQ-KPI	Data Quantity KPI
E2E	End to End
ECR	Elastic Container Registry
ELCM	Experiment Life Cycle Manager
ETSI	European Telecommunications Standards Institute
FoF	Factory of the Future
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IaC	Infrastructure as Code
IAM	Identity and Access Management
IEM	Interaction of Employees and Machines
IIoT	Industrial Internet of Things
IP	Internet Protocol
KPI	Key Performance Indicator
KVI	Key Value Indicator
LOC-KPI	Location KPI
NEF	Network Exposure Function
NetApp	Network Application
NPN	Non-Public Networks
OAuth	Open Authorization
PLI	Production Line Infrastructure
QoS	Quality of Service
REST	REpresentational State Transfer
RBAC	Role-Based Access Control
SaaS	Software as a Service
SEC	Security Guarantees and risk Analysis
SIEM	Security Information and Event Management
SLA	Service Level Agreement
SLS	Service Level Specification
SME	Small Medium Enterprise
SUT	System Under Test
TCP	Transmission Control Protocol
UE	User Equipment
UI	User Interface

vApp	Vertical Application
VM	Virtual Machine
VNF	Virtual Network Function
VPN	Virtual Private Network

EXECUTIVE SUMMARY

EVOLVED-5G responds to the 5G PPP ICT-41-2020 5G innovations for verticals with third party services call, whose main goal is to deliver enhanced experimentation facilities on top of which third party experimenters (e.g., SMEs or any service provider and target vertical users) will have the opportunity to test their applications. The EVOLVED-5G project realises this vision by encouraging the creation of a NetApp ecosystem revolving around a 5G facility which will provide the tools and processes for the development, verification, validation, and certification of NetApps as well as their smooth running on top of actual 5G network infrastructures, and mechanisms for market releasing.

The primary objective of this deliverable is to present the use of the validation tools and open repository, developed in WP3 and more specifically in T3.2, steering the validation of the NetApps developed in WP4. In the same scope, details regarding the manual validation of a NetApp with the vApp, guaranteeing that the coupling of the former with the latter, is resulting to a proper service delivery, are also presented.

The work presented in this deliverable will guide the project towards later implementations in regard to the validation process, by systematically using the tools of the EVOLVED-5G validation framework to run automated tests and examine non-functional properties (performance, interoperability, security etc.) of the industry 4.0 NetApps.

TABLE OF CONTENTS

1 Introduction 1
1.1 Purpose Of The Document 1
1.2 Structure Of The Document 1
1.3 Target Audience 1
2 Validation Environment 3
2.1 Validation Environment Reference Architecture 3
2.2 CI/CD Support For Validation Process 4
2.3 Open Repository 4
2.3.1 Amazon Web Services Elastic Container Registry 5
2.3.2 JFrog 5
2.4 Validation Platforms 6
2.4.1 Openshift 6
2.4.2 Kubernetes in Athens Platform 7
2.4.3 Kubernetes in Malaga Platform 8
2.5 5G Core Network Functions 9
2.5.1 3GPP NEF APIs 9
2.5.2 CAPIF Core Function 9
3 Manual Validation Tests 11
3.1 Overview 11
3.1.1 Test case templates 11
3.1.2 Test case reports 31
4 Automated Validation Tests 65
4.1 Overview 65
4.2 Tools For Automation 65
4.2.1 Jenkins 65
4.2.2 Robot Framework 66
4.2.3 ELCM 67
4.2.4 OpenTAP 67
4.3 Validation Pipeline 68
4.3.1 Validation Steps 68
4.3.2 Open-repository onboarding 70
5 Vulnerability And Code Analysis 72
5.1 Container (NetApp) Vulnerability Scan (Trivy) 72
5.2 Open Ports Testing (Nmap, Telnet and Netstat) 72
5.3 Syntax Analysis For Source Code (Sonarqube) 73
5.4 Open-Source License Scanning (Debricked) 73

6 Conclusion 75
7 References 76
8 Annexes 78
8.1 Manual Test Case Template 78
8.2 Quality Assurance-Test Case Report 79

LIST OF FIGURES

Figure 1 Validation Environment architecture 3
Figure 2 CI/CD flow 4
Figure 3 AWS ECR for EVOLVED-5G Validation process 5
Figure 4 JFrog storage platform 5
Figure 5 Openshift look and feel 7
Figure 6 Openshift deployment topology 7
Figure 7 Kubernetes architecture in the Athens platform 8
Figure 8 Kubernetes architecture in the Málaga platform 8
Figure 9 Validation process of the NetApps towards the NEF APIs consumption 9
Figure 10 Validation process of the NetApps towards the CAPIF APIs consumption 10
Figure 11 Overview of the predefined IMM scenario on the NEF emulator graphical interface 13
Figure 12 NEF and NetApp running on same computer 17
Figure 13 vApp running on distant computer 18
Figure 14 vApp Anita computer network connected with NEF and NetApp 18
Figure 15 Test environment with pre-conditions enabled for ININ NetApp 21
Figure 16 Definition of the experiment in the ELCM 23
Figure 17 Screenshots of the IMM manual test execution 33
Figure 18 NEF Emulator set-up for the INF validation test 34
Figure 19 Dummy database table for factory cells 35
Figure 20 Dummy database table for area machines 35
Figure 21 Example of the chatbot test 35
Figure 22 Results of post requests during the validation test 36
Figure 23 Access Evolved5G screen from menu 38
Figure 24 Select UE external ID 38
Figure 25 Retrieve UE Latitude and Longitude 38
Figure 26 Compare with coordinates on NEF Emulator 39
Figure 27 Retrieve the exact position of UE from another system 39
Figure 28 Obtain the LOC-KPI (passed) 39
Figure 29 Test case scenario in the NEF Emulator Map view 41
Figure 30 Test case scenario in the NEF Emulator Map view 41
Figure 31 UMA-CSIC NetApp prerequisites and components running during a test 43
Figure 32 UMA-CSIC NetApp validation running on the ELCM 45
Figure 33 Provision “external id” and “qos profile” for selected UE 47
Figure 34 Subscription to Monitoring event and AsSessionwithQoS API 47
Figure 35 Initiation of the gateway/UE services via IoT Management 47
Figure 36 QoS Guaranteed and QoS_NOT_Guaranteed notifications 48
Figure 37 vApp user creates a subscription and the NetApp receives the subscription request 50
Figure 38 NEF emulator is sending messages to NetApp 50
Figure 39 vApp enforces a policy about a specified UE. 51
Figure 40 Panel displaying the nornal logs 51
Figure 41 Panel displaying the detected anomaly logs 52
Figure 42 Predefined scenario prepared in NEF Emulator 53
Figure 43 Notification retrieval from NEF Emulator by the NetApp 54
Figure 44 Deployed scenario in NEF Emulator 54
Figure 45 Vertical Application portal 55
Figure 46 Subscription for a single callback 56

Figure 47 Callbacks returning to NetApp from multi-time monitoring subscription 57
Figure 48 Location update on Vertical Application after receiving a Monitoring Event from NEF Emulator 57
Figure 49 IQB Unit Tests Execution Results 59
Figure 50 NEF Emulator set-up 60
Figure 51 Simulated Tiago robot positioned at the Cell ID 2 location matching the UE 3 Cell ID connection 61
Figure 52 LocalizationNetApp publishing the current Cell_ID of UE 3 61
Figure 53 Initial state of the NEF Emulator with a valid connection with the robot 62
Figure 54 TIAGo matching movement of the haptic device when QoS guaranteed 64
Figure 55 TIAGo not moving when QoS not guaranteed 64
Figure 56 Jenkins pipeline flow 1 65
Figure 57 Jenkins pipeline flow 2 66
Figure 58 Jenkins's pipeline definition. 66
Figure 59 UI of OpenTAP 67
Figure 60 Stages of the validation pipeline in Jenkins 70
Figure 61 NetApp Validation Process 70
Figure 62 Validated image onboarded to Open-Repository 71
Figure 63 Trivy vulnerabilities report 72
Figure 64 Netstat output 73
Figure 65 SonarQube report example 73
Figure 66 Debricked licenses report 74

LIST OF TABLES

Table 1 Test case for the Remote Assistance in AR NetApp 12
Table 2 Test case for the Chatbot assistant NetApp 14
Table 3 Test case for the Digital/Physical twin NetApp 15
Table 4 Test case for the Occupational safety analysis NetApp 19
Table 5 Test case for the Industrial grade 5G connectivity NetApp 20
Table 6 Test case for the Anomaly detection NetApp 21
Table 7 Test case for the Smart irrigation 5G Agriculture NetApp 23
Table 8 Test case for the Traffic Management NetApp 25
Table 9 Test case for the 5G enabled SIEM NetApp 26
Table 10 Test case for the ID Management and Access Control NetApp 27
Table 11 Test case for the Localization NetApp 28
Table 12 Test case for the Teleoperation NetApp 30
Table 13 Test report for IMM NetApp -General information 31
Table 14 Test report for IMM NetApp Pre-Conditions 31
Table 15 report for IMM NetApp -Test Case Sequence 32
Table 16 Test report for INF NetApp-General information 33
Table 17 Test report for INF NetApp Pre-Conditions 34
Table 18 Test report or INF NetApp -Test Case Sequence 36
Table 19 Test report for GMI NetApp-General information 37
Table 20 Test report for GMI NetApp Pre-Conditions 37
Table 21 Test report or GMI NetApp -Test Case Sequence 37
Table 22 Test report CAF NetApp-General information 40
Table 23 Test report CAF NetApp Pre-Conditions 40
Table 24 Test report CAF NetApp-Test Case Sequence 40
Table 25 Test report UMA NetApp -General information 42
Table 26 Test report UMA NetApp Pre-Conditions 42
Table 27 Test report UMA NetApp -Test Case Sequence 43
Table 28 Test report ININ NetApp- General information 45
Table 29 Test report ININ NetApp Pre-conditions 45
Table 30 Test report ININ NetApp- Test case Sequence 46
Table 31 Test report ZORT NetApp -General Description 48
Table 32 Test report ZORT NetApp Pre-conditions 48
Table 33 Test report ZORT NetApp -Test case sequence 49
Table 34 Test report 8Bells NetApp -General Description 52
Table 35 Test report 8Bells NetApp Pre-conditions 52
Table 36 Test report 8Bells NetApp -Test case sequence 53
Table 37 Test report FOG NetApp -General Description 55
Table 38 Test report FOG NetApp Pre-conditions 55
Table 39 Test report FOG NetApp-Test case sequence 55
Table 40 Test report IQB NetApp-General information 57
Table 41 Test report IQB NetApp Pre-Conditions 58
Table 42 Test report IQB NetApp-Test Case Sequence 58
Table 43 Test report UML/PAL NetApp-General information 59
Table 44 Test report UML/PAL NetApp Pre-Conditions 59
Table 45 Test report UML/PAL NetApp -Test Case Sequence 60
Table 46 Test report PAL NetApp General information 61

Table 47 Test report PAL NetApp Pre conditions..... 62
Table 48 Test report PAL NetApp-Test case sequence 62

1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

One of the main objectives of EVOLVED-5G project and WP5 more specifically, is to define and provide validation tests to all the NetApps. Thus, the current document *"NetApps Validation and onboarding to Open Repository"* provides details on the use of the validation tools and open repository with the aim to validate the NetApps developed in WP4. Moreover, the document provides details towards the manual validation of a NetApp with a vApp, guaranteeing that the coupled NetApp-vApp can properly deliver the enhanced features in the provisioned service under different conditions (i.e., the vApp makes proper use of all the capabilities exposed by the NetApp).

1.2 STRUCTURE OF THE DOCUMENT

The core part of the document is divided into the following sections:

- **Section 2: Validation Environment** focuses on the Validation Framework that is being utilised for the implementation of the tests that the NetApps should undergo. This framework includes the Continuous Integration / Continuous Development (CI/CD) tools, the Kubernetes clusters and 5G Network functions that will be installed in the EVOLVED-5G platforms (Athens-Malaga).
- **Section 3: Manual Validation Tests** aggregates the manual validation test cases of the NetApps that have been performed as the initial step towards the final validation process, as well as the test case reports for each test case and each NetApp.
- **Sections 4: Automated Validation Tests** describes the role of the of CI/CD automation server tools and the steps comprising the validation process of the NetApps.
- **Section 5: Vulnerability And Code Analysis** summarizes the reports that have been obtained after using these tools during the validation phase.

1.3 TARGET AUDIENCE

The release of the deliverable is public, intending to expose the overall EVOLVED-5G ecosystem and NetApps Lifecycle design to a wide variety of research individuals and communities.

From specific to broader, different target audiences for D5.2 are identified as detailed below:

- **Project Consortium:** To validate that all objectives and proposed technological advancements have been analysed and to ensure that, through the proposed NetApp's lifecycle phases and the various environments, further work can be concretely derived. Furthermore, the deliverable sets to establish a common understanding among the consortium with regards to:
  - The validation framework used within the EVOLVED-5G platforms for the management and orchestration of the resources and the procedures in the testbeds.
  - The validation process of the NetApps is realised by the several tools that are geared towards the automation of the process.
- **Industry 4.0 and FoF (Factories of the Future) vertical groups:** To crystallise a common understanding of technologies, and design principles that underline the development of the

NetApps, and to understand the utilisation of the network APIs exposed by the 5G Infrastructure. A non-exhaustive list of Industry 4.0-related groups is as follows:

- Manufacturing industries (including both large and Small Medium Enterprise (SMEs)) and IIoT (Industrial Internet of Things) technology providers.
- European, national, and regional manufacturing initiatives, including funding programs, 5G-related research projects, public bodies and policy makers.
- Technology transfer organizations and market-uptake experts, researchers, and individuals.
- Standardisation bodies and Open-Source Communities.
- Industry 4.0 professionals and researchers with technical knowledge and expertise, who have an industrial professional background and work on industry 4.0-related areas.
- Industry 4.0 Investors and business angels.

- **Telecom Service Providers:** to engage with verticals and also to simplify the way 5G services can be offered to a potential customer or 3rd party service provider.
- **Other vertical industries and groups:** To seek impact on other 5G-enabled vertical industries and groups in the long run. Indeed, all the architectural components of the facility are designed to secure interoperability beyond vendor specific implementation and across multiple domains. The same categorization as the above but beyond Industry 4.0 can be of application.
- **The scientific audience, general public and the funding EC Organisation:** To document the work performed and justify the effort reported for the relevant activities. The scientific audience can also get an insight of the validation process of the NetApps developed by the project.

2 VALIDATION ENVIRONMENT

2.1 VALIDATION ENVIRONMENT REFERENCE ARCHITECTURE

The Validation Framework is used within the EVOLVED-5G platforms for the management and orchestration of the resources in the testbed. It can be used both for the implementation of common validation tests that all NetApps must pass during their lifecycle, as well as for customized tests that the SMEs and the vApp developers may be interested in, such as the procurement of specialized KPIs or the validation of certain features. As depicted in Figure 1 below, the Validation Framework is separated in several components, namely:

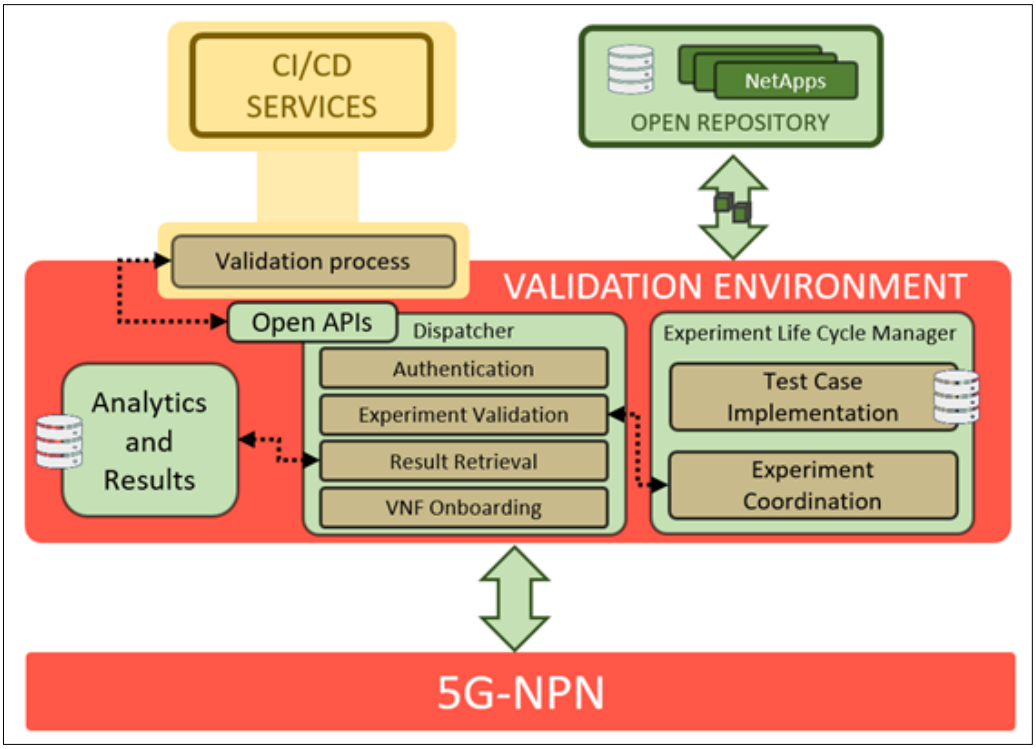


Figure 1 Validation Environment architecture

- The Dispatcher, which acts as the front-end of the testbed, and handles authentication and validation of requests, user management and access to different resources. Additionally, it also offers the possibility of onboarding Virtual Network Functions (VNFs) to the platforms.
- The Experiment Life Cycle Manager (ELCM), which coordinates the execution of experiments while managing the correct usage of resources. The ELCM also offers the possibility of specifying and storing test-case definitions, which are used for the implementation of specific experiments that are composed by tasks interfacing with the different components of the infrastructure.
- The Analytics Module and Results storage provide a long-term solution for storing the raw results generated by the experiments, along with capabilities for performing statistical analysis, graphical representations as well as study the generated measurements in-depth.

These functionalities are exposed in the front-end by the Open Application Programming Interface (Open-APIs) implemented by the Dispatcher, which is a collection of Representational

State Transfer (REST) APIs that are used by the CI/CD Services in order to initiate the execution of the validation process. Once the request is received, the Validation Framework assumes the control of the validation, which is implemented in the platform as a result of the collaboration between the testbed owners (Malaga and Athens) and the SME performing the validation of the NetApp.

The ELCM, as experiment coordinator, provides a set of pre-defined tasks that can be used for interfacing with the different components of the testbed as well as with any equipment required for a particular experiment that may be provided by the SMEs. Using these tasks, it is possible to implement the sequence of actions defined in the validation Test Cases, producing either fully automated experiments or semi-automated experiments where part of the validation is performed automatically while other steps depend on the interactions of humans with the equipment or Systems Under Test (SUT).

Once the Validation is completed, the generated measurements are used for the creation of a report that includes both standardized information as well as any Key Performance Indicators (KPIs) agreed with each SME during the consultation phase that takes place prior to the implementation of the validation test cases.

2.2 CI/CD Support For Validation Process

In a similar approach as in the certification process [1], CI/CD will be used as a method for delivering NetApps in EVOLVED-5G. Indeed, it is undeniable that CI/CD practices bring to the EVOLVED-5G project countless advantages as reducing errors in deployments, delivering software faster and making easier rollbacks processes.

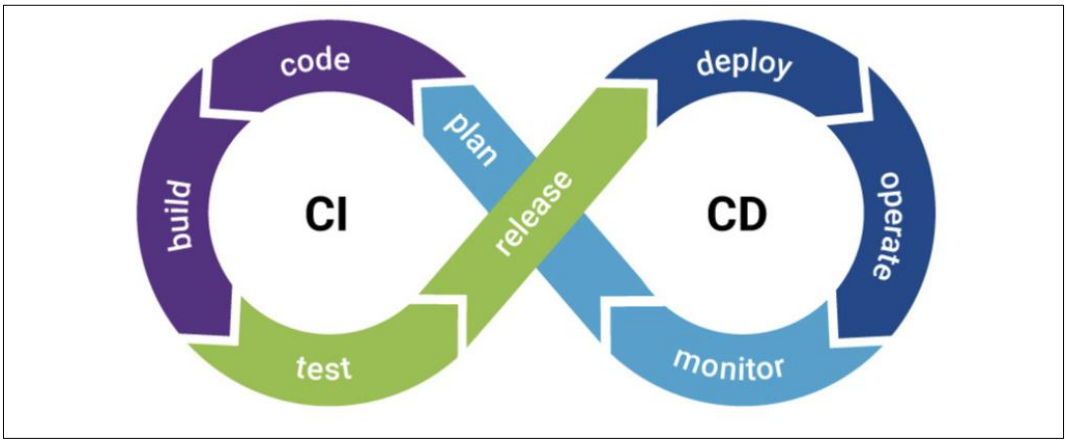


Figure 2 CI/CD flow

All the CI/CD tools and processes that the project utilises have been explained in the D3.2 [1] but the main idea of this deliverable is to explain how these tools are involved in the onboarding NetApp process. The main CI/CD tools are: Open Repository to store artifacts, Jenkins for automation, Terraform [2] to manage the Infrastructure as Code (IaC) and Robot Framework to build automated tests.

2.3 Open Repository

Validation Process will grab NetApp Source Code from GitHub repository. Additionally, it is necessary to indicate the 5G Platform environment selected to run the validation tests. This environment can either be Athens or Málaga platforms, depending on the features that the

NetApp needs to validate, as well as the underlying supporting 5G features of each platform. With these inputs, it is possible to launch the Validation Pipeline that will automatically execute all validation tests to complete EVOLVED-5G's validation process. After the successful validation of the NetApps, the container images of the NetApps and all validation reports generated will be uploaded to the JFrog Artifactory.

2.3.1 Amazon Web Services Elastic Container Registry

AWS ECR [3] is an AWS solution to easily store and share container images through the public cloud. Therefore, this public container registry makes easy to pull and push images from any platform in the project, in this case, Malaga and Athens platforms. Figure 3 AWS ECR for EVOLVED-5G Validation process depicts the container image of a successfully validated NetApp that has been pushed to the AWS ECR validation registry upon the completion of the process.

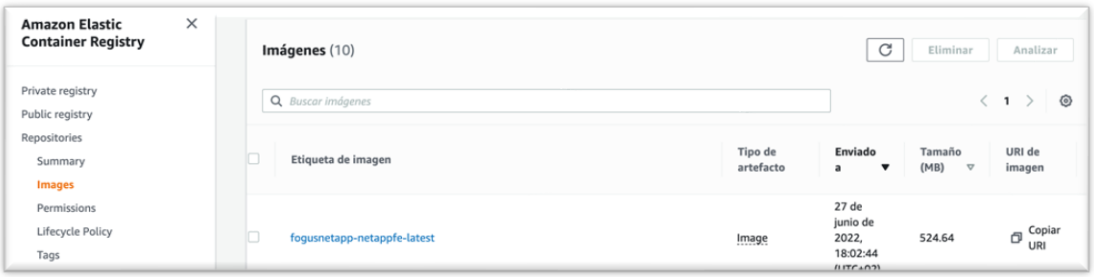


Figure 3 AWS ECR for EVOLVED-5G Validation process

2.3.2 JFrog

JFrog Artifactory [4] is a repository manager that supports all available software types. This feature makes JFrog Artifactory a complete storage solution, since it is possible to store more types of files (images, reports, artifacts, etc). For example, in Figure 4, how all the reports generated after each validation stage are stored in JFrog Artifactory are depicted

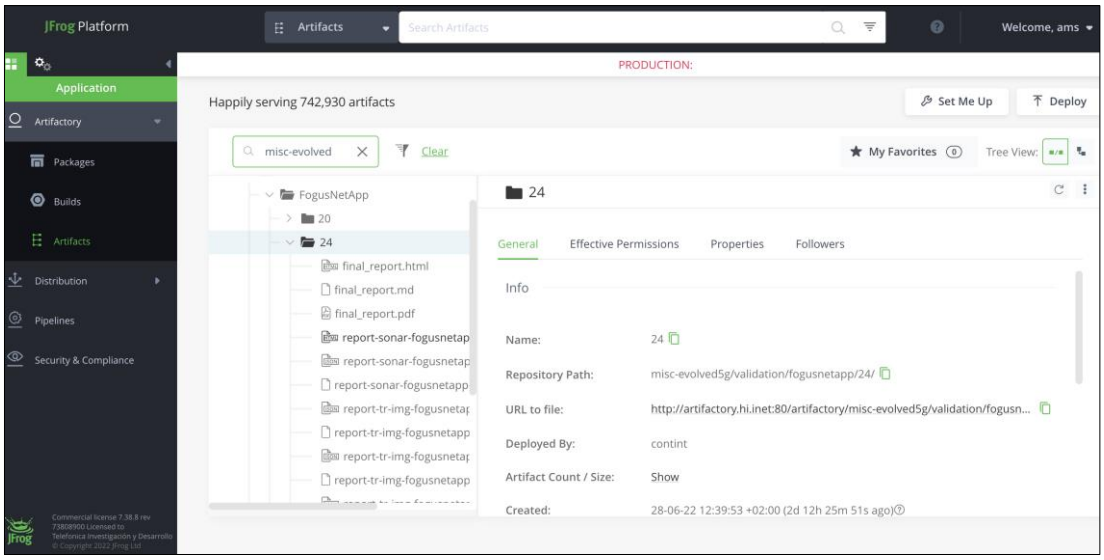


Figure 4 JFrog storage platform

2.4 VALIDATION PLATFORMS

In order to Validate the NetApps and execute validation tests, whether they are automatic or manual, the NetApps need to be deployed in containerized infrastructure, such as Kubernetes and executed in order to test the interaction with NEF and CAPIF.

EVOLVED-5G has identified three different container platform instances, each of them with different purposes:

- Two Kubernetes clusters will be associated with the two 5G platforms, Málaga and Athens. This way, the NetApps will be deployed in container infrastructure integrated with the 5G platforms.
- The third container infrastructure instance is available in the CI/CD environment. This is independent from 5G platforms and is being used for testing purposes and the integration of EVOLVED-5G SDK.

Jenkins pipelines will take as an input parameter the container infrastructure selected to execute the pipeline. For this purpose, VPN connections have been established between Telefónica´s CI/CD environment and both Málaga and Athens 5G Platforms.

In the following subsections we describe the container platforms infrastructure used in each instance.

2.4.1 Openshift

Openshift by RedHat [5] is the container management platform that has been selected to deploy and validate NetApps in the CI/CD environment, offering built-in monitoring, enterprise-grade security, centralized policy management, and self-service provisioning. This is the environment used for EVOLVED-5G SDK commands to test building, deploying and destroying NetApps during development and verification tests.

2.4.1.1 Openshift GUI

Openshift offers a default feature-rich graphical interface that allows you to build, deploy, scale, monitor and implement almost every Kubernetes task from a web browser, both as an administrator and developer. This makes Openshift management platform simple and easy to use, and timesaving for every developer considering the required short learning curve in contrast to the User Interface (UI) dashboard in Kubernetes, which is not implemented by default. On the contrary, in the case of Kubernetes a developer has to deploy and access it by running different commands manually, requiring from the developer to learn additional commands before being able to use them, dealing with a longer learning curve. An example of the Red Hat UI can be seen in the Figure 5 and Figure 6.

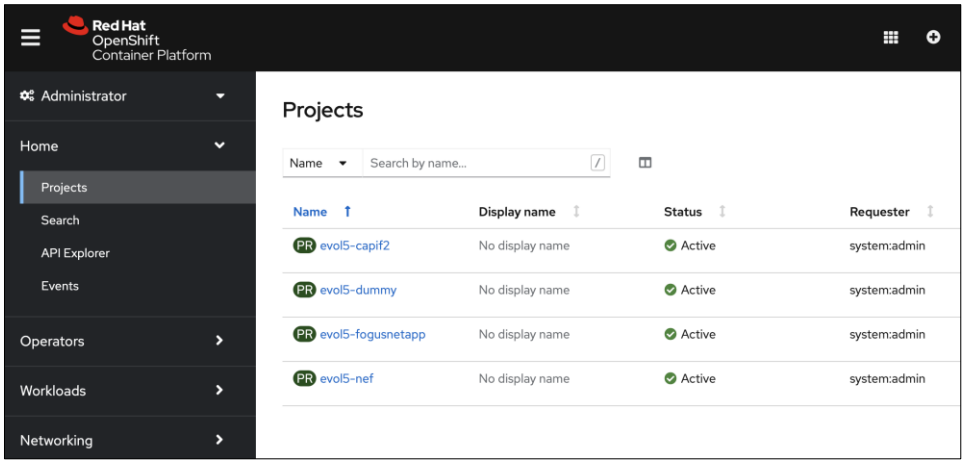


Figure 5 Openshift look and feel

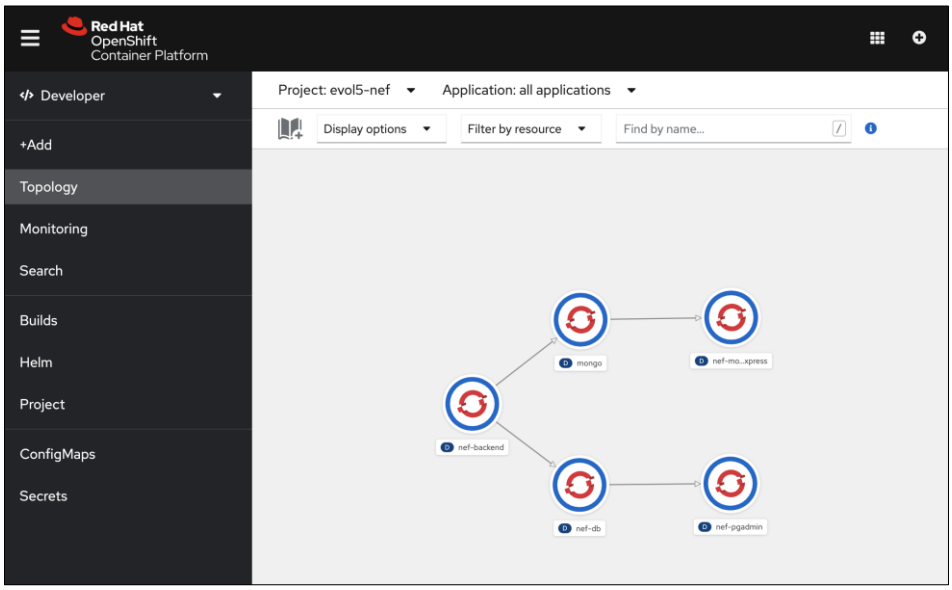


Figure 6 Openshift deployment topology

2.4.1.2 Openshift Security Features

To prevent potential assaults, it is necessary to adequately secure each team member's access and permission to the containerization platform when working on a team of developers with various permissions on an application. In order to prevent the problem of account compromise, OpenShift by default provides role-based access control (RBAC), which helps to make sure that each developer in EVOLVED-5G platforms only receives permission to the required features. Other security rules, such as Identity and Access Management (IAM) and Open Authorization (OAuth), are all automatically created when a project is created using OpenShift. This saves time and makes more efficient the set-up of an application environment. Even though Kubernetes now enables Kubernetes RBAC, still everything has to be set-up by the developer manually on that side, making Openshift a much more efficient solution, facilitating the engagement and the use of the EVOLVED-5G ecosystem by SMEs and third-party developers.

2.4.2 Kubernetes in Athens Platform

The Deployment of Kubernetes (K8s) cluster in Athens Platform consists of 3 virtual machines (VMs), 1 master node and 2 working nodes each one with the following characteristics, 2x vCPUs, 4GB RAM. Access to the nodes can be achieved via Virtual Private Network (VPN) connectivity

and the overall setup of the cluster is based on the K8s v1.23.3. Moreover, it includes Calico plugin [6], geared to provide a flexibility towards the connectivity of the network through specific configurations and policies. The high-level architecture of the deployment can be seen in Figure 7 below.

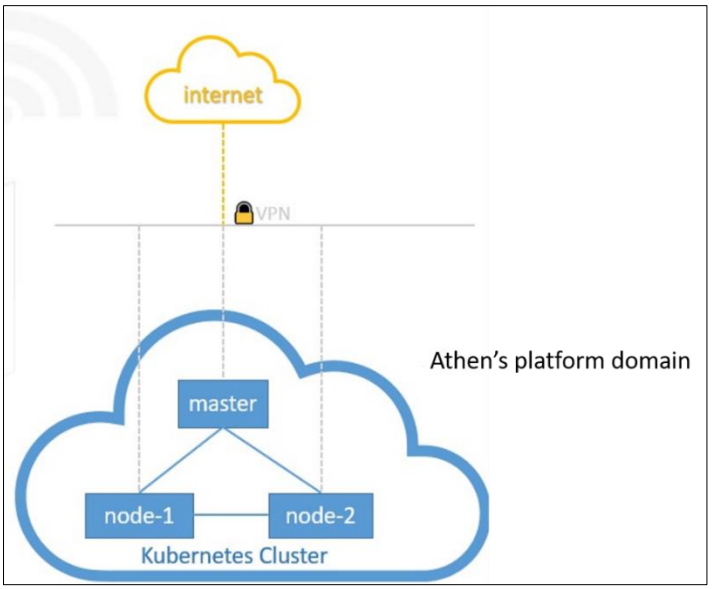


Figure 7 Kubernetes architecture in the Athens platform

2.4.3 Kubernetes in Malaga Platform

The Kubernetes deployment in the Málaga platform is based on a multi-master architecture configured for high availability. The deployment is composed by three master nodes and three worker nodes, with an additional node dedicated to storage, as can be seen in Figure 8.

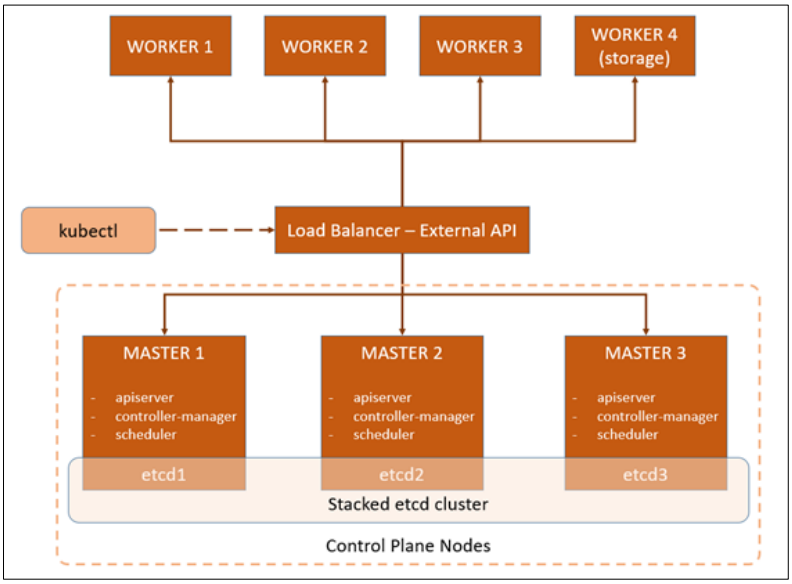


Figure 8 Kubernetes architecture in the Málaga platform

Access to the nodes is controlled by a load balancer in order to guarantee a correct distribution of the work. The management of the deployments and resources is performed with kubectl [7]. For security reasons, access to the required functionality is controlled by using dedicated namespaces and a RBAC scheme, which provides isolation and prevents possible issues from affecting every deployment per user.

The container runtime used is containerd [8]. For networking, the Container Network Interface configured is Calico, combined with MetalLB [9] for load balancing. In addition, the KubeVirt [10] module enables VMs deployment on top of the containerized infrastructure.

2.5 5G Core Network Functions

2.5.1 3GPP NEF APIs

Network Exposure Function (NEF) emulator exposes Northbound APIs following the 3GPP TS 29.522 [11] and TS 29.122 [12] specifications. It is one of the core components of the project that enables NetApps to be developed and to carry out the use cases defined from the verticals. After the successful development of the NetApps it is essential to ensure that communication between NetApps and the NEF Emulator is acceptable. Therefore, the interaction between NetApps and the NEF APIs is an important and mandatory part of the validation process. NEF Emulator has a pivotal role on both manual and automated tests as described in Sections 3 and 4, respectively.

The validation process between NEF Emulator and NetApps is twofold. As can be seen from Figure 9, the NetApps are validated for the proper consumption of the NEF APIs exposed by the emulator (step 1), and afterwards whether they can successfully receive the callback notifications (step 2). At the time of writing this deliverable, the current version of the NEF emulator supports the MonitoringEvent and AsSessionWithQoS APIs. The details of the architectural concerns, implementation aspects and the technologies used to develop the NEF emulator can be found on D3.1 [13]. In addition, a thorough description of NEF APIs, including the two APIs that are currently supported can be found on D4.1 [14].

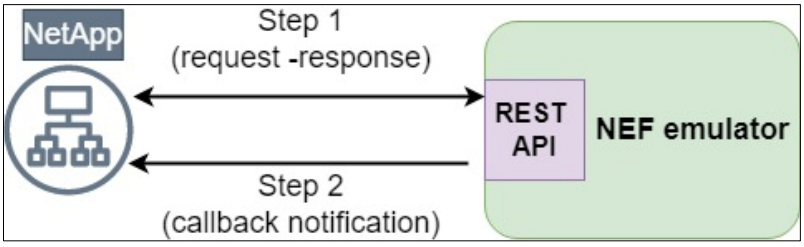


Figure 9 Validation process of the NetApps towards the NEF APIs consumption

In order to facilitate the usage of the NEF Emulator as part of a NetApp validation and to simplify the implementation of the validation test case in the EVOLVED-5G platforms, certain backend functionality has been integrated in the ELCM. In particular, a new Task, that makes use of the *ue_movement* endpoints, has been developed and can be included as part of any NetApp validation process for initiating and stopping the movement of individual User Equipment (UEs) during the execution of the validation test case.

2.5.2 CAPIF Core Function

The 3GPP Common Application Programming Interface Framework (CAPIF) has been standardized to guarantee a unique reference model for API-based service provisioning in 3GPP systems. EVOLVED-5G has selected CAPIF as the API Framework to enable the interaction between the NetApps (vertical industries domain) and the 5G Core exposed APIs (mobile networks domain). As it has been explained in the D3.2 [1], CAPIF deployment follows the related 3GPP and European Telecommunications Standards Institute (ETSI) standards, and focused on the open implementation of the major part of CAPIF, i.e., the CAPIF Core Function.

During validation phase, NetApp will test connectivity and performance with the different implemented modules (API services) of the CAPIF Core Function. More precisely, the functionality that will be tested are the following (Figure 10):

- Authenticating the API invoker based on the identity and other information required for authentication of the API invoker.
- Supporting events from the CAPIF authentication with the API invoker;
- Publishing, storing and supporting the discovery of service APIs information.
- Onboarding a new API invoker and offboarding an API invoker.

Basically, NetApps are validated for the proper consumption of the CAPIF APIs exposed and the reception of callback notifications from CAPIF (e.g., events).

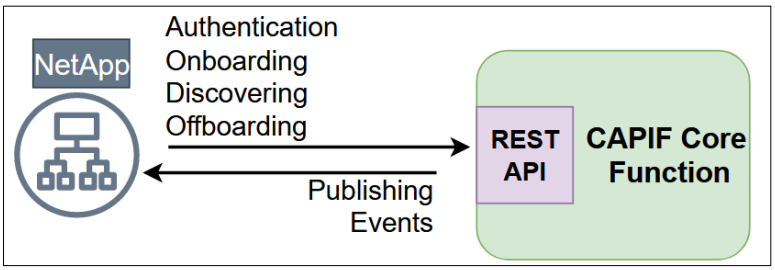


Figure 10 Validation process of the NetApps towards the CAPIF APIs consumption

CAPIF Core Function has been developed during first period of the project, but NetApps yet need to integrate CAPIF APIs to make use of it for Validation and Certification phases.

3 MANUAL VALIDATION TESTS

3.1 OVERVIEW

This section describes the concept of the manual validation tests as the initial step in the framework of the NetApps' validation process. The manual validation tests have been defined with scope to assess the functionality enhancement of the vApp by coupling it with its companion NetApp and the NEF emulator. Thus, the manual validation tests are built on top of use-case driven test-cases/scenarios geared to collect qualitative data/Key Value Indicators (KVIs).

Given the high-level functionality assessment nature of the manual validation tests, they are being categorised based on the four pillars in the EVOLVED-5G context: Interaction of Employees and Machines (IEM), Efficiency in FoF Operations (FoF), Security Guarantees and risk Analysis (SEC), as well as Production Line Infrastructure (PLI).

3.1.1 Test case templates

The test case template used in EVOLVED-5G project is the one that has been adopted by 5GENESIS experimentation methodology [15], however, in order to meet the requirements of EVOLVED-5G there has been an adaptation by modifying some of the existing fields, adjusting their current content, but also adding new fields to reflect better the NetApp and VApp specificalities.

In particular, the EVOLVED-5G test case template is formed by the following descriptions:

- **Scenario**: Description of the motivation and the scope of the test in a qualitative level, through a reference scenario in a real (industrial) environment.
- **Testing Infrastructure**: Any requirement that needs to be in place before execution of this test case. A list of specific pre-conditions that need to be met by the system under test, or the set of software and hardware components involved along with their configuration.
- **Target KVI**: Here goes the definition of the target KVI. Each test case targets only one KVI (main KVI). However, secondary measurements from complementary KVIs can be added as well. The definition of the main KVI specializes the related target metric (the name of the related target metric is declared in the first row of the template).
- **Test case sequence**: This section specializes the measurement process (methodology) of the metric for the selected underlay system. More precisely the steps to be followed for performing the measurements are specified as well as the iterations required for proper assessment of the KVI and the monitoring frequency are declared in this section.

3.1.1.1 Test cases of Interaction of Employees and Machines (IEM) pillar

3.1.1.1.1 Remote Assistance in AR NetApp Test Case

The manual validation test that has been performed for the specific NetApp is related to the detection and notification of Quality of Service (QoS) change for both the worker and the remote helper with respective notification of the vApp by the companion NetApp.

Table 1 Test case for the Remote Assistance in AR NetApp

EVOLVED-5G Test Case Template	IMM_1	Notification of QoS change for both the worker and the remote helper	KVI Metric: Qualitative, latency
Scenario (storyline)	The scenario considers the use of AR headsets as vApp that are used by workers in the factory in order to receive remote assistance by remote experts. Thus, remote assistance scenario involves two kinds of users: the worker within the factory and the remote expert outside of it (for instance, in another building). The coupling of the vApp (AR headset) with the Notification of QoS change NetApps aims at informing both users when the required QoS cannot be guaranteed, independently of which side the issue occurred, in order to reassure that communication faults and erroneous assistance is avoided.		
Testing Infrastructure (Pre-conditions)	The SUT is the local deployment of the NEF emulator, NetApp and vertical application for two remote users: 1 playing the role of the worker, the other one the role of the expert. Required: • NEF Emulator and NetApp are locally installed on a computer. • vApp installed on AR headsets (Hololens 2), which must be connected to the same network than the computer. A predefined scenario has to be created within the NEF emulator. A first set of cells with 2 UEs (1 for the worker, 1 “perturbator” representing another user working in and out of the cell to trigger QoS monitoring events), as can be seen in Figure 11. A second set of cells on the same principle (1 expert + 1 “perturbator”) Procedure: 1. Create the predefined scenario in NEF Emulator with cells and UEs 2. Start emulator, NetApp and vApp instances and establish end-to-end communication between them Start the remote assistance call on vApps. The two users should hear and see each other.		

Target KVI	The targeted KPIs/KVIs resulting from the test case is 1) the notification of each user when the QoS guarantee changes ("qualitative" KPI) and 2) If possible, the end-to-end latency related to these notifications. End-to-end latency between the issue detection (QoS change because of perturbator event) and: 1. The moment the local user (the one in the same cell than the perturbator) receives the corresponding notification 2. The moment the other user also receives the notification Target values: should be less than 60ms (end-to-end). The starting point is always the detection of QoS change (emulator side). If end points cannot be on the end-user side (vApps), then they can be on the NetApp side (local or remote NetApp receives notification of the event from the emulator).
Test Case Sequence	1. Users start the remote assistance (vApp) 2. The NetApp creates the endpoint for QoS monitoring with the corresponding user id, then request the appropriate subscription to the NEF emulator. 3. For now, the QoS should be guaranteed on both sides 4. Activate perturbator 1 (on worker's side) on the emulator map. 5. When perturbator 1 enter the worker's cell, check that both the worker and the remote expert are notified that the QoS cannot be guaranteed. Measure end-to-end latency. 6. When perturbator 1 leaves the worker's cell, check that both the worker and the remote expert are notified that the QoS can be guaranteed again. Measure end-to-end latency. 7. Repeat steps 5 and 6 for x5 iterations 8. Stop perturbator 1 on the emulator map. Repeat steps 4 to 7 on the expert's side with perturbator 2

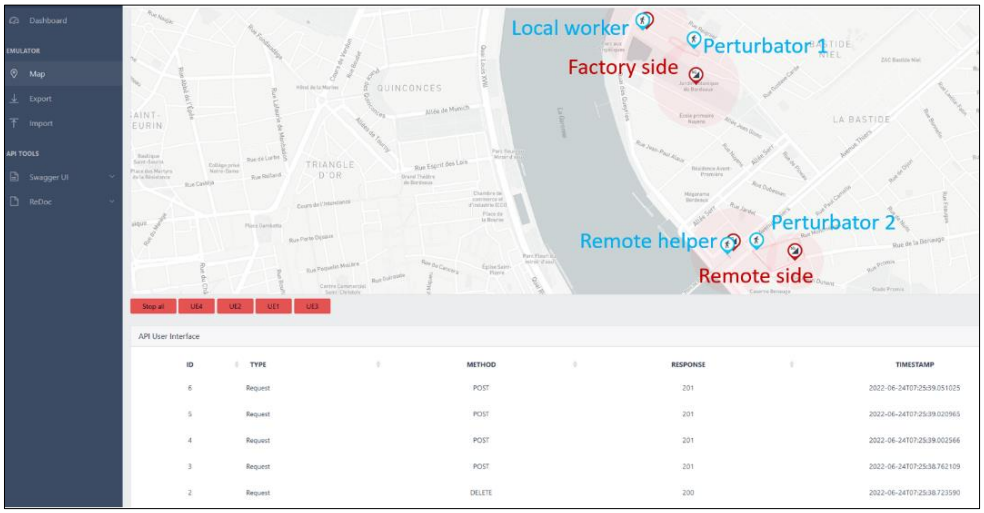


Figure 11 Overview of the predefined IMM scenario on the NEF emulator graphical interface

3.1.1.1.2 Chatbot assistant NetApp Test Case

According to the development status of the Chatbot Assistance NetApp, the manual validation test that has been performed is related to the returning list of machines that appears on the chatbot screen. This list depends on the factory cell in which the worker is located, thus this qualitative test checks whether this procedure is performed correctly, ensuring the workers' safety and the smooth function of the environment by testing the correct retrieval of machines with respect to each worker's location and authorization. Table 2 provides the test case for the Chatbot assistant NetApp validation test.

Table 2 Test case for the Chatbot assistant NetApp

EVOLVED-5G Test Case Template	**INF_1**	**Returning list according to worker cell**	**KPI Metric: Machine list**
Scenario (storyline)	The scenario considers the use of a chatbot vApp by workers in order to receive information (e.g., user manuals, technical sheets, maintenance sheets etc) of the machines that are located in close proximity of them. As different worker authorization levels exist and also the factory is considered to be sectioned in cells, there is a different list of machines that will appear on each worker's chatbot according to his location and his authorization level. The coupling of the Chatbot vApp with the companion Chatbot assistant NetApp provides information to the vApp on the location of the worker, as well as its authorization level in the specific sector in order the vApp to properly adapt the loading of the technical documentation and access rights to them based on the authorisation level of the worker. As a reference to a real industrial environment, with this test, we want to ensure the workers' safety and the smooth function of the environment by testing the correct retrieval of machines with respect to each worker's location and authorization.		
Testing Infrastructure (Pre-conditions)	To measure the validity of the machine lists retrieved by workers, a set of machine lists, corresponding to cell ids, needs to be preconfigured in the NetApp's database. Also, a predefined scenario with cells and UEs (i.e., represent workers) has to be created in the NEF Emulator. Lists of machines should be assigned to all cells in the scenario and also stored in NetApp's database. The acceptable list of machines will be validated in the vertical app side after the NetApp returns the list to the vertical app. The validation process should be performed every time the worker handovers to a different cell and will end by the time the worker will have moved through all the cells. In order for the test to be properly performed, the NEF Emulator-NetApp must be locally installed. More specifically, the following preconditions are also necessary: 1. End-to-end communication between the vApp - NetApp - NEF Emulator 2. Create the predefined scenario in NEF Emulator with cells and UEs representing workers		

	Last but not least, as mentioned before, the machine assignment information in different cells inside the factory must be stored in the Database.
Target KVI	The SUT is the local deployment of the NEF emulator, NetApp and vertical application. The targeted KVI stemming from the test case is the receipt of the expected machines' list related to the correct cell as defined by the factory administrator. In order to perform the necessary measurement, the chatbot screen will be used as a reference point. No specific target values are required. The success of the test will be defined by the coherence with the designed scenario stored in the database.
Test Case Sequence	To successfully perform the described test the following actions should be performed sequentially: 1. User types "work" in the Chatbot App. This command is transmitted to the Web Chatbot Backend. 2. The Web Chatbot Backend calls the endpoint of the NetApp with the appropriate worker id. 3. The NetApp calls the endpoint of NEF emulator and retrieves the area (cell id) of the worker. 4. The NetApp accesses the database and retrieves the machines under the area of the worker according to the worker id. 5. The NetApp returns the list to the vApp. 6. The vApp displays the list on the screen If the list is what expected the validation process continues. Repeat steps 1 to 6 for N iterations, where N is the number of the cells configured in the predefined scenario.

3.1.1.1.3 Digital / Physical Twin NetApp

The Digital/physical twin NetApp focuses on validating the measurement of the quality of data transfer that the vApp received from the underlying network. Table 3 provides the respective Test case definition that used for performing the manual validation process of the GMI hot bonder vertical app in conjunction with the respective NetApp.

Table 3 Test case for the Digital/Physical twin NetApp

EVOLVED-5G Test Case Template	**GMI_1**	**Measurement of Quality of Data Transfer**	**KPI Metric: Data & Location KPIs**
Scenario (storyline)	The scenario of the Digital/Physical twin involves a real hot bonder machine (vApp #1) located in the industrial space along with a replica hot bonder (vApp #2) at a remote location, where an expert and certified group of technicians is located and supervises the bonding process remotely.		

	A critical parameter for the reliable creation of the digital – physical twin is the **Quality of Data Transferred**, both in terms of quantity (i.e., percentage of measurements received by the Replica hot bonder) and consistency (i.e., consistency of data received to data transmitted). Therefore, the coupling of the vApps with the Digital/Physical twin NetApp is used in order to enhance the capability of the hot bonders to gain awareness of the quantity and consistency of the received data and therefore adapting the bonding process accordingly (e.g., pausing the process in case of quality degradation of slowing down the bonding process etc.) In a second step, the knowledge of the geographical location is interesting for the concept of the repair twin, which could be used also for measuring the accuracy of the provided information.
Testing Infrastructure (Pre-conditions)	To calculate the Quality of Data Transferred KPIs, a full typical composite curing cycle will be executed by the Primary hot bonder and required data will be transmitted through 5G to the Replica hot bonder. Curing cycle data will be recorded both at the Primary hot bonder (i.e., just before being transmitted) and at the Replica hot bonder (just before used for controlling of the Physical – Digital Twin curing cycle). Inclusion of power or 5G network interruption may be also included, so as to calculate KPIs in various conditions. For LOC-KPI, it will be a matter of obtaining the information from another positioning system, for example GPS, and comparing the values obtained. Pre-conditions: 1. NEF Emulator-NetApp are installed on the same computer 2. Connecting Anita to the Internet via Wi-Fi or 5G 3. End-to-end communication between the vApp - NetApp - NEF Emulator The last step is the creation of the predefined scenario in the ANITA hot bonder
Target KVI	**“Data Quantity KPI” (DQ-KPI)**: Counting of total number of values (Time, Temperature, Vacuum etc.) that have been prepared for transmission by the Primary Hot Bonder and of the total number of values received by the Replica hot bonder. By dividing these two values, the percentage of data quantity actually transferred will be calculated. **Target value**=100% **“Data Consistency KPI” (DC-KPI)**: The actual values received by the Replica Hot Bonder (Time, Temperature, Vacuum etc.) will be compared one-by-one with the corresponding values transmitted by the Primary hot bonder, to review consistency. Received values with <1% deviation compared to transmitted values will be considered PASS while values with >1% deviation FAIL. DC-KPI will be calculated as the percentage of PASS over PASS + FAIL values. **Target value**=100% **Location KPI" (LOC-KPI):** The value obtained from the NEF Emulator will be compared to a value obtained with another positioning system like GPS or GALILEO. A coordinate distance calculation tool will then be used to measure

	the distance between the two values. If the difference is less than 0.5 km, the test will be conclusive. Example of tool: https://boulter.com/gps/distance/ **Target value**=PASS
Test Case Sequence	1) From the primary Anita, generate the standard data package, as it would be obtained under real repair conditions. 2) Calculate its size (will be useful for measuring the DQ-KPI) 3) Calculate the checksum (will be useful for measuring the DC-KPI) 4) Send the data packet to the storage location 5) From another computer, which will simulate the secondary Anita, retrieve the data packet from the storage location. 6) Measure the data size and checksum and calculate DQ-KPI and DC-KPI 7) Get the location from the NEF Emulator (Cell_ID and cell coordinates) 8) Get the position from another system 9) Measure the distance between the two values to obtain the LOC-KPI

The following figures describe the pre-configurations that are crucial in order to perform the manual validation process for the Digital/Physical twin NetApp.

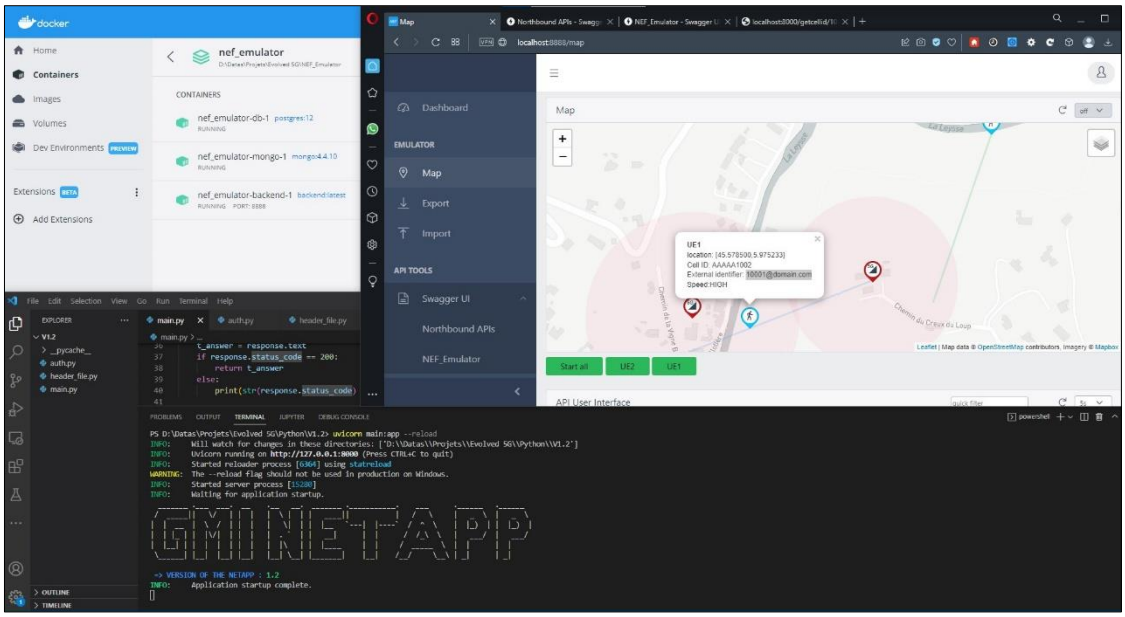


Figure 12 NEF and NetApp running on same computer

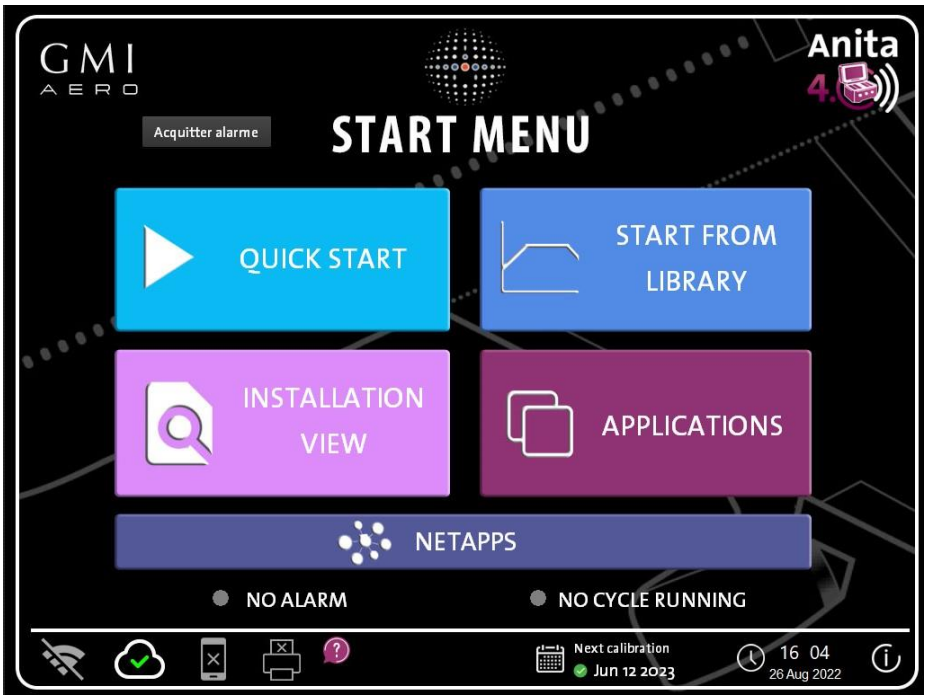


Figure 13 vApp running on distant computer

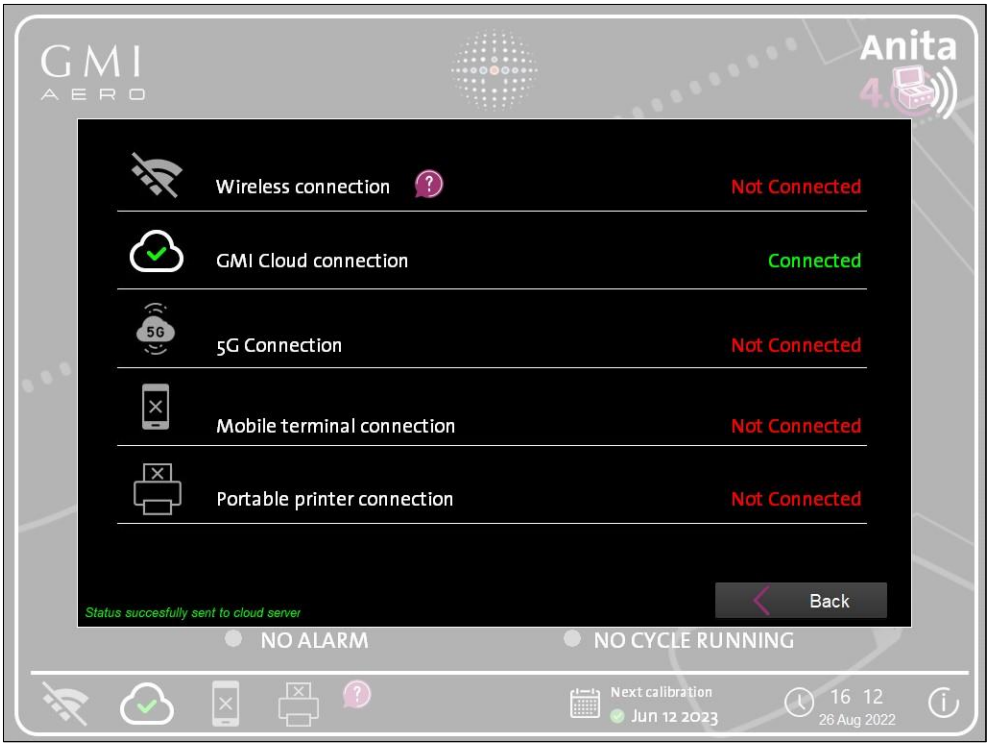


Figure 14 vApp Anita computer network connected with NEF and NetApp

3.1.1.2 Test Cases of Efficiency in FoF Operations (FoF) pillar

3.1.1.2.1 Occupational safety analysis NetApp

In order to ensure the safety applicability by the personnel in a construction site or in a factory (e.g., the helmets usage) the machine vision vApp is coupled with a QoS assurance NetApp in order to adapt the vApp's video analytics procedure according to the available QoS. Table 4 provides the respective test case.

Table 4 Test case for the Occupational safety analysis NetApp

EVOLVED-5G Test Case Template	CAF_1	QoS Assurance Functional Test	KPI Metric: QoS
Scenario (storyline)	The scenario considers a machine vision vApp that detects usage of safety helmets (or other safety equipment) by the personnel in a construction site or in a factory. To ensure the safety, the detection and the subsequent actions by the vApp, the whole process must act as close to the real time as possible. To achieve that, the data flow, consisting of the video input and output of the detection results, must be unhindered. Machine vision algorithms have some minimum requirements for the video quality to work reliably. All this need certain network bandwidth. Therefore, the coupled NetApp with the machine vision vApp will provide network quality (QoS) measurements feedback to the machine vision vApp. Then the vApp will be capable of adapting its operation accordingly. For example, in a reported decreasing network bitrate, the vApp will decrease the frame rate or the video quality in order to reassure service continuation of the service, preventing the increase of video latency which may decrease real-time safety.		
Testing Infrastructure (Pre-conditions)	The NetApp, vApp and NEF Emulator are running on the local computer (localhost). Creation of a scenario in NEF Emulator using 3 cells.		
Target KVI	The main KVI is to adapt the vApp's video analytics procedure according to the available QoS. If there is a QoS degradation vApp will stop the video analytics as it is not reliable to run machine vision algorithms.		
Test Case Sequence	1. NEF Emulator is started. 2. NetApp is started and configured to receive QoS guaranteed messages. 3. vApp is started. 4. Virtual robot starts moving from the point very close to a cell (good QoS area) at a straight line until it exits QoS guaranteed area. 5. In some point NetApp receives a message about degrading bitrate and sends message to vApp. 6. vApp stops video analytics as it is not reliable to run machine vision algorithms on it anymore.		

3.1.1.2.2 Industrial grade 5G connectivity NetApp

The industrial grade 5G Connectivity NetApp aims at providing guaranteed QoS to the sensor data collection vApp of the factory or the industrial space. The respective test case is provided in Table 5.

Table 5 Test case for the Industrial grade 5G connectivity NetApp

EVOLVED-5G Test Case Template	ININ_1	IoT 5G QoS Assurance Functional Test	KPI Metric: QoS
Scenario (storyline)	The scenario considers the use of Industrial IoT gateway (acting as vApp) for the collection of data from the factory sensors, enhanced by the coupling of a QoS NetApp for adding reliability to the enhanced IIoT system (i.e., coupled vApp with NetApp) in case of Network QoS degradation. Thus, the main target of the use case is to provide guaranteed QoS to factory operation related "sensor data collection service" being one of multiple services run on a UE. The UE is subscribed to 5G connectivity service with assured Service Level Agreement (SLA) and Service Level Specification (SLS) and with corresponding QoS profile that is configured by the vertical application (IoT Management UI) through the NetApp for that specific UE (IoT Gateway) and corresponding "external id". This way, the IoT Management can dynamically enable or disable services running on the UE (IoT Gateway) depending on the current QoS monitoring parameters levels received from 5G NEF and targeted SLS thresholds, thus providing required QoS for critical services, i.e., sensor data collection service in our particular case.		
Testing Infrastructure (Pre-conditions)	The SUT is the local deployment of the NEF emulator, NetApp and vertical application. Required: • IoT Management, IoT collector and IoT gateway are deployed and operational. • Predefined scenario in NEF emulator to trigger "Session with QoS" notifications. • SUT features: ○ IoT Management API to configure QoS profile (currently available via UI, API not implemented yet) ○ traffic services on the 5G IoT Gateway (critical traffic - sensor data collection, BE traffic – e.g., video) ○ collection of KPIs from monitoring events received by NEF will be stored in the IoT Collector (vertical app) Procedure: Validation of the "AsSessionwithQoS API" response on the IoT Management (Vertical App) for UE (IoT Gateway) – if successful: 1. The services on IoT Gateway will start (critical traffic - sensor data collection, Best Effort (BE) traffic – e.g., video). Collection of KPIs from monitoring events will be stored in IoT Collector (vertical app).		

Target KPI/KVI	Main target KPI: UE is subscribed to configured 5G Quality of Service Identifier (5QI) session Reference points: NEF API Reference protocol stack: Internet Protocol (IP) Transmission Control Protocol (TCP) Hypertext Transfer Protocol Secure (HTTPS) Target value: UE is subscribed to 5QI=82
Test Case Sequence	1. UE is provisioned with "external id" and "qos profile" in IoT Management UI (vertical app) 2. IoT Management UI registers "external id" in the 5G NEF via NetApp 3. NetApp subscribes to "Monitor Event API" in the 5G NEF 4. NetApp subscribes to "Session with QoS API" in the 5G NEF 5. UE is assigned the correct QoS profile, IoT Management starts the IoT Gateway services: sensor data collection service (factory operation traffic) and video streaming service (BE traffic) 6. NetApp receives "QOS_NOT_GUARANTEED" notification from the 5G NEF and notifies IoT Management

The test environment with pre-conditions enabled for ININ NetApp is presented in the following figure:

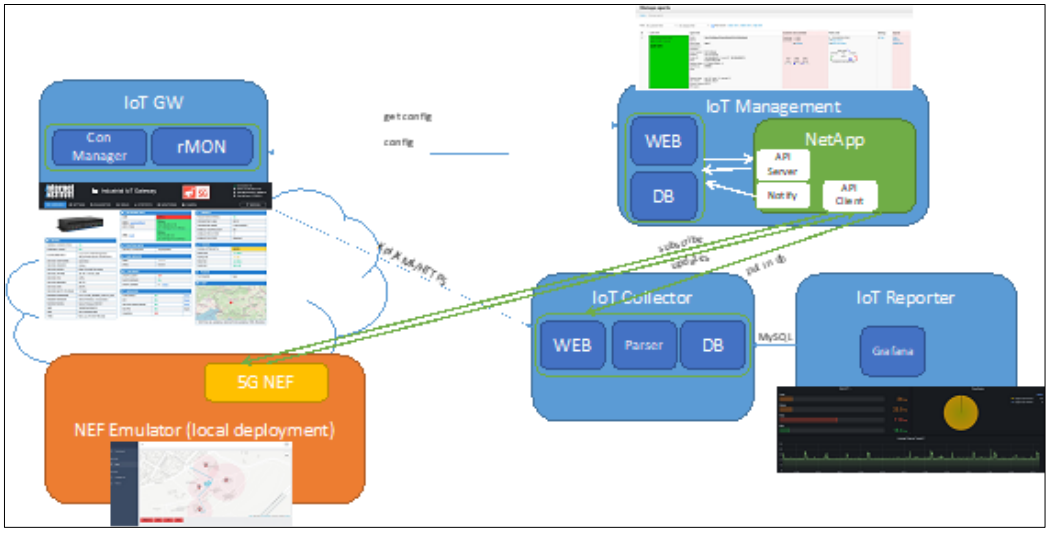


Figure 15 Test environment with pre-conditions enabled for ININ NetApp

3.1.1.2.3 Anomaly Detection NetApp

The network infrastructure of an industrial environment is very important to remain secure and any anomaly to be early detected. In that direction the anomaly detection NetApp is coupled with programmable network switches, acting as vApps, in order to enhance the network awareness of the programmable switches and give them the capability to act and adapt their policies accordingly.

Table 6 Test case for the Anomaly detection NetApp

EVOLVED-5G Test Case Template	**Zortenet_1**	**Anomaly Detection**	**KPI Metric: Proper Alert**

Scenario (storyline)	The scenario of the anomaly detection vApp system is to be enhanced in terms of functionality by being coupled with the respective anomaly detection NetApp that will report the connected Cell of each UE together with a relative alarm level based on the defined policies of the factory. More specifically, the current version of the Anomaly Detection NetApp is monitoring the cell usage of each UE and reports accordingly to the vAPP alarm system. The user can define policies about the cell usage of each UE. An anomaly alert is triggered when a UE starts using a non-permitted cell, based on the defined policies that are reflecting the factory guidelines and regulations.
Testing Infrastructure (Pre-conditions)	NEF Emulator-NetApp are locally installed and running 1. vApp – NetApp – NEF emulator are deployed and running locally 2. NEF Emulator has the default scenario. 3. No policies are enforced (database of NetApp is empty)
Target KPI/KVI	• The targeted KPI/KVI of this use-case is the ability of the Vertical App user to manage policies and receive anomaly alerts. • Removing a policy and validating that the relevant alerts are no longer received by the user. • vApp receiving logs from the monitored endpoint of NEF-emulator. • Exiting the vApp and validating if alerts are being received for the policies created in the previous session.
Test Case Sequence	• vApp user creates a subscription for one or multiple UEs. • NetApp receives the subscription request then forwards the request to NEF emulator using the **/nef/api/v1/3gpp-monitoring-event/v1/netApp/subscriptions** • NetApp is receiving callbacks from NEF emulator using the registered callback endpoint. • vApp user enforces a policy regarding which cells the specific UE is allowed to operate. • NetApp receives and stores the policy. • NetApp is performing anomaly detection using applied policies. • If any policy is violated alerts are being sent to vApp. The vApp user is able to receive alerts of the anomaly.

3.1.1.2.4 Smart irrigation 5G Agriculture NetApp

The Smart Irrigation NetApp aims to act as a central point for collecting and pre-processing the set of measurements created by an array of sensors distributed on a plantation. This reduces the need for processing power and features, which in turn reduces the cost per sensor and ease their installation. For the case of UMA-CSIC the validation of the NetApp has been performed in an automatic manner using the Open5Genesis Framework. For this purpose, a test case has been defined and implemented in the Málaga platform: the sequence of actions of the Test Case have been implemented using *pytest*, while the coordination of the experiment has been performed

by the ELCM. Docker has been used for the containerization of the NetApp, and both kinds of vertical apps.

The definition of the experiment in the ELCM is the following:

```
UmaCsicTestCasefull:
  - Order: 1
    Task: Run.CliExecute
    Config:
    Parameters: ./testfull.sh
    CWD: <Path to the Script>
- Order: 2
    Task: Run.PublishFromPreviousTaskLog
    Config:
    Pattern: ".*Failed.*"
    VerdictOnMatch: "Fail"
    VerdictOnNoMatch: "Pass"
    Standard: True
    Distributed: False
```

Figure 16 Definition of the experiment in the ELCM

The first step of the experiment uses the "Run.CliExecute" task which starts a script called "testfull.sh" given its location in the machine. The second step uses the task "Run.PublishFromPreviousTaskLog" to search the ".*Failed.*" pattern in the logs of the first step, if there is a match then the experiment is considered failed, otherwise it passes.The script "testfull.sh" initialises containers for the NetApp and vApps in the beginning of each iteration, and then it starts the test sequence.

All the iterations have nine steps that involve a pytest invocation. Pytest also uses a plugin to attempt two retries in case of a failed step, a step is considered successful if it passes at least in one retry. The script retrieves the output of the steps to decide if the iteration is succeeded or not. Finally, at the end of every iteration, it stops and deletes and all the containers are deleted.

Table 7 Test case for the Smart irrigation 5G Agriculture NetApp

EVOLVED-5G Test Case Template	**UMA_1**	**Smart Irrigation NetApp Functional Test**	**Correctness**
Scenario (storyline)	The UMA-CSIC NetApp cannot be validated in isolation, considering that its functionality depends on the measurements generated by external sensors, while the collected results are expected to be retrieved by another Vertical Application. For this reason, the scenario of this test is composed by all these elements (the NetApp itself, a set of simulated sensors and a test Vertical App), which are controlled by the Validation Framework.		
Testing Infrastructure (Pre-conditions)	• The NEF emulator is running and accessible. • The NEF emulator is configured with three different cells and three devices, one for each cell.		

Target KPI/KVI	The target KPI/KVI is to run the test sequence 25 repeated times and if all Iterations behave as expected then the test is Successful. In order to consider that an iteration is *Successful*: - All steps in the **Test Case Sequence** need to be performed. - The output of the execution of all steps is as expected. Reliability is a secondary measurement. Its definition is: The ratio of successful vs. total number of iterations.
Test Case Sequence	This sequence refers to the actions performed in each test iteration: 1. The NetApp, and both vApps containers are created and started. 2. Using the ''Cell Management '' endpoints, two crop areas are defined, with the same configuration of the cells previously configured in the NEF emulator except for one (This is for test updating methods). 3. Datalogger vApp will read a CSV file with known data and POST some historics using ''Historic Management endpoints" from NetApp. This endpoint will use the Monitoring Event API (with one time request) to retrieve the cell id of the stationary UEs (dataloggers). 4. Using the ''Historic Management '' endpoints, GET the historics registered in the NetApp and confirm that the result is the expected. 5. Use prediction function from statistics vApp endpoints and test if it is the expected output. 6. Verify that all other function endpoints of the statistics App work properly. 7. UPDATE one cell number with the same identifier of the other cell. An error is expected. 8. GET the historics attached to a specific cell. Then, UPDATE the cell number (with the cell unused in NEF-Emulator) of the cell selected in this step. Finally, GET again the historics attached to that cell. The same output is expected. 9. UPDATE data of one historic and check that it is done correctly. 10. DELETE another cell using ''Cell Management '' and confirm the cell has been deleted along with its attached historics. 11. DELETE one historic of the last existing cell and confirm that it is deleted. 12. RESET database using ''Database utils'' endpoint. Verify that the database is cleared. NetApp and both vApps containers are destroyed.

3.1.1.3 Test cases of Security Guarantees and Risk Analysis (SEC) pillar

3.1.1.3.1 Traffic Management NetApp

An industrial space is a closed environment, where only registered traffic flows within the specific domain are allowed in order to reassure the integrity and the security of the underlying network. In that perspective and given the fact that a 5G network will be used in industrial spaces as an infrastructure of the underlying network, it is important the current traffic management systems to be updated with the current registered traffic flows.

Table 8 Test case for the Traffic Management NetApp

EVOLVED-5G Test Case Template	**8BELLS_1**	**Returning list of IP addresses connected to the network**	**KPI Qualitive Metric: UEs list**
Scenario (storyline)	Eight Bell's NetApp, being developed for security guarantees risk analysis pillar and in the scope of the EVOLVED-5G project, offers the accurate measurement of traffic over an interface, specific to a device and perform a simple check of "unregistered" traffic outside the 5G Core Network.		
Testing Infrastructure (Pre-conditions)	To measure the validity of the returned list, a predefined scenario with cells and UEs (representing the workers connected to the network) has to be created in the NEF Emulator. The process takes place every time a new UE connects to the network. NEF Emulator-NetApp are locally installed 1. End-to-end communication between the vApp - NetApp - NEF Emulator 2. Create the predefined scenario in NEF Emulator with cells and UEs		
Target KPI/KVI	The target KPI based on the use case is the return of the expected ipv4 addresses list, which are present in the network. Secondary KPI: Lessening of the burden of a congested device.		
Test Case Sequence	1. The NetApp calls the endpoint of the NEF Emulator and retrieves a list of all active IP addresses in the network. 2. The NetApp and the vApp communicate by ssh connection. 3. The list is passed to the vApp, which create the appropriate firewall rules. 4. Any traffic that is not registered in these IP addresses of the list is dropped. Repeat the process every time a new UE connects to the network		

3.1.1.3.2 5G-Enabled Security Information and Event Management (SIEM) NetApp

The issue of integrating the various IP networks that are currently in use in an industrial setting with a 5G Non-Public Network (NPN) that supports mobile devices used by employees as well as 5G-enabled devices on the production line is faced by the department in charge of the IT and network systems. Interfacing with the 5G network's core network and integrating the data

exchanged with 5G into the factory's security information and event management system are the major objectives (SIEM).

Table 9 Test case for the 5G enabled SIEM NetApp

EVOLVED-5G Test Case Template	**FOGUS_1**	**5G-Enabled SIEM**	**KPI Metric: Qualitative**
Scenario (storyline)	In an industrial environment the department that is in charge of the IT and network systems faces the problem of unifying the various existing IP networks with a 5G Non-Public Networks (NPN) that supports mobile devices of the employees and 5G-enabled devices in the production line. The main scope is to interface with the core network of the 5G network and embed the information exchanged with 5G to the factory's security information and event management system (SIEM).		
Testing Infrastructure (Pre-conditions)	For the testing process, a real or emulated 5G connectivity infrastructure with end devices attached should be available. This infrastructure should expose the standardized 3GPP NEF APIs (northbound APIs) related network events, analytics, location etc. Optionally the API interaction with those APIs shall be based on the secure and trust framework of 3GPP CAPIF. In addition, the provisioning of a SIEM system (in the role of vertical App) should be up and running. The test is decoupled of the traffic type and the services that are supported by the 5G and other existing industrial networks. For the experiment dummy traffic is fed to the 5G network to trigger the generation of events or data to the 5G northbound APIs.		
Target KPI/KVI	The NetApp should guarantee the seamless and reliable information exchange among the vApp (SIEM system) and the 5G NPN. The target indicators for validating this will be measured at the vApp side and above the IP layer of the protocol stack. An initial list of indicators is the following: - Percentage of successfully captured real events / data - Quality of representation of the information at the vApp (SIEM) Synchronization level among the main interacting entities, i.e., the SIEM and the 5G NPN		
Test Case Sequence	• The 5G NPN is up and running with end devices attached. • The SIEM system is up and running • The FOGUS NetApp is up and running and authorised for communication with the 5G NPN • Experimentation traffic is generated related to the usage of the network by end devices • Data related to Network events, Analytics, Location updates etc. are generated and are available at the 5G NPN • In a continuous manner / for a monitoring period T:		

	o The NetApp of FOGUS requests (periodically) information from the network and also receives events shared through the 5G northbound APIs o The NetApp feeds the SIEM system with appropriately formatted data The SIEM merges the visualisations of the 5G network inputs with those coming from the other networks under monitoring.

3.1.1.3.3 ID Management and Access Control NetApp

The Openness of the 5G network creates security challenges that should be addressed by authorizing access to specific users and third parties. The ID management and access control netapp provides this capability to the respective vertical app, securing the access to the 5G core by external third parties and minimizing by this way any malicious behavior.

Table 10 Test case for the ID Management and Access Control NetApp

EVOLVED-5G Test Case Template	**IQB_1**	**ID Management and Access Control**	**Authentication/Authorization**
Scenario (storyline)	Secure authentication and authorization involve the following parties: The 3rd Party NetApp, the NetApp developed by IQB, as well as the NEF Emulator. The 3rd Party should be able to authenticate and be authorized to consume the NEF Emulator APIs through the IQB NetApp, using OpenID Connect. The location monitoring subscription created will then generate call backs that shall be received on a test server.		
Testing Infrastructure (Pre-conditions)	The SUT is the local deployment of the NEF Emulator, NetApp, Keycloak SaaS, a 3rd Party (Postman) and a test server to receive call backs. The pre-conditions are the following: • NEF Emulator and NetApp are locally installed on a Virtual Machine and host respectively. • 3rd Party NetApp (could be a Postman instance), which must be connected to the same Wi-Fi network with the computer. • An image of a Keycloak must have been pulled in order to run it inside a docker container. Keycloak will provide OpenID Connect functionality. • A realm (IQB-NetApp), client (client-NetApp) and user (sampleuser) must have been specified in Keycloak by the 3rd Party via Keycloak API consumption or the management console. 1. Create the predefined scenario in NEF Emulator with cells and UE • A cell with 1 UE. • A second cell for the 3rd party to receive a call back when the cell ID of the UE changes.		

	2. Start the emulators, 3rd Party instance(s) as well as the Keycloak container 3. Start the NetApp. The NetApp will communicate with all other instances and establish end-to-end communication between them 4. Start the call backs server in the same machine as the NEF Emulator
Target KPI/KVI	The targeted KPI/KVI resulting from the test case is the ability of the 3rd Party to be authenticated and authorized to consume NEF APIs and receive call backs. Secondary KPIs include: • Verification that the 3rd Party is unable to consume APIs when it has not authenticated. End-to-end latency
Test Case Sequence	1. From the 3rd Party attempt to create a location monitoring subscription. The request should result in failure since the 3rd Party has not been authenticated and thus is not authorized to consume NEF APIs 2. From the 3rd Party login to the NetApp by the /login endpoint (which uses OpenID Connect). 3. From the 3rd Party create a location monitoring subscription by performing a request on the NetApp. The request shall be authorized using the OAuth2.0 token provided by the NetApp through Keycloak during the login process. 4. Receive callbacks on the 3rd Party when the cell ID of the UE changes. 5. From the 3rd Party logout from the NetApp Attempt to read active subscriptions. The request should result in failure since the 3rd Party is no longer authenticated.

3.1.1.4 Test cases of Production Line Infrastructure (PLI) pillar

3.1.1.4.1 Localization NetApp

Localization is an important add-on feature provided by the 5G core system, which can significantly enhance the awareness of the industrial robots and other moving UGVs. The test case for the localization NetApp validates the proper coupling between Robots/vApps and the localization NetApp by assessing the location awareness that is achieved by this coupling.

Table 11 Test case for the Localization NetApp

EVOLVED-5G Test Case Template	**UML_01**	**Localization Assurance Functional Test**	**Reliability, Location KPIs**
Scenario (storyline)	The main idea of the NetApp is to provide a rough global position estimate based on the 5G core capabilities since an agent can retrieve the information of the cellular antennas that relate to, and those antennas have geographical coordinates known. Any IoT device which has motion behaviours must localize itself. Indoor localization is still an open problem since the agent needs global		

	references. With this test, we want to prove that an agent can react based on cell id information linked to a pose.
Testing Infrastructure (Pre-conditions)	The SUT is the local deployment of the NEF emulator, NetApp and vertical application. It is required multiple functionalities: • A Simulator with an agent model which can navigate autonomously given a set of locations (vApp). In the real scenario for the next phase of the validation tests, the simulator can be exchanged by a real robot. • Predefined scenario in NEF emulator which simulates an agent switching from different cellular antennas and the relation agent pose and antenna which is connected to can be retrieved by the emulator. • Localization NetApp features: o Must read the cell are connected (Cell ID) o Must associate Cell ID to geographical position of the antenna (Cannot be retrieved yet since the emulator do not provide that information) o From the antennas that the agent is connected it is providing a global geographical pose of the agent (Cannot be done since there is no direct geographical information retrieved from the emulator yet). This can define a KPI based on the accuracy in the localization. For now, it is not used since it cannot be measured. It is assumed a functional test instead. Procedure: Validation of the Localization NetApp communication with the emulator end to end (E2E). The Cell ID information is retrieved and is associated to a location where the robot can navigate to. It is emulating a robot navigating to the antenna is connected with. If the Cell Id is read properly. 1. The robot must navigate to the location of the Cell is reading If the robot is navigating to a location and the Cell ID measurement changes in the way. The robot must interrupt its current navigation goal and go to the new location associated to the Cell ID is reading.
Target KPI/KVI	Main target KPI: NetApp is configured to read Cell ID values from the UE Reference points: NEF API Reference protocol stack: IP/TCP/HTTPS and Data Distribution Service (DDS) Target value: Agent navigates to the cell that the UE is connected to.

Test Case Sequence	• Start Tiago simulator (via PAL Robotics container) • Run UML Tiago controller • Run NEF emulator • Start “device 3” routine • Run LocalizationNetApp • Wait until the “device 3” change coverage zone in the emulator • The robot moves in a different location when a new cell id is received If there is proper communication between the NetApp and the simulated TiAGo robot, as soon as the “device 3” changes cell coverage the robot should change its direction.

3.1.1.4.2 Teleoperation NetApp

Teleoperated machines and robotics arms will play a crucial role in the Industry 4.0 environments. Therefore, the enhancement of them with network QoS awareness is very important for safety and productivity reasons. The Teleoperation NetApp adds this functionality to already existing robotic arms that are coupled with them, and this improvement is assessed in the Test case for the Teleoperation NetApp.

Table 12 Test case for the Teleoperation NetApp

EVOLVED-5G Test Case Template	**PAL_01**	**Teleoperation NetApp test case – Communication between NetApp and real TiAGo robot**	**Reliability**
Scenario (storyline)	The scope of this test is to check the correct communication with the TiAGo real robot., the NetApp and the NEF Emulator. The Tiago arm will be teleoperated through our vertical app with the geomagic touch haptic device. The expected result is that the real TiAGo robot [16] stops any movement when the QoS is no longer guaranteed. And continue to move after the QoS is guaranteed again. The QoS is based on the QoS updates received from the NetApp through the NEF emulator.		
Testing Infrastructure (Pre-conditions)	Running all the simulation steps described below (“Test Case Sequence”) and wait at least 1 change of direction by the robot.		
Target KPI/KVI	We are expecting the robot to perform adapt on the different input provided by the NetApp. And that the connection permits a decent teleoperation experience.		

Test Case Sequence	The sequence in order to perform the aforementioned test is as follows: • Start real TiAGo • Run NEF emulator • Start "device 3" routine • Run Teleop_NetApp • Run the Teleoperation • Start Teleoperating • Wait until the "device 3" changes coverage zone in the emulator and that the QoS is no longer guaranteed • The robot should stop completely and start again when the QoS is guaranteed again

3.1.2 Test case reports

The following sections describe the test case reports for each NetApp as partitioned within the four pillars of the project. The template for reporting the test cases is comprised of 4 main cells: the details of each step, the expected results of each step, the actual results where the expected or the unexpected behaviour of the aforementioned step can be described and finally the results of each step, where 4 options can be defined Pass / Fail, Not executed or a Suspended status.

3.1.2.1 Interaction between Employees and Machines (IEM) pillar

3.1.2.1.1 Test report- Remote Assistance in AR

As the manual test was conducted locally, the secondary KPI about end-to-end latency was put aside and kept for a later test phase in Malaga's infrastructure. Otherwise, all test steps passed successfully.

Table 13 Test report for IMM NetApp -General information

Test Case ID	*IMM-Netapp-1*
Test Case Description	*Qualitative*
Created By	*IMM*
Tester's Name	*Charles Bailly*
Date Tested	*02-05-2022*
Test Case Results	***Pass***

Table 14 Test report for IMM NetApp Pre-Conditions

S #	Prerequisites:
1	Running NEF emulator with defined scenario (4 UEs: worker, remote helper, and 2 additional UEs to trigger QoS notifications)
2	Two instances of the NetApp running on local computer (1 containerized)
3	vApp installed on the AR headset (Hololens 2 on same Wi-Fi network than the computer. Another vApp instance running on local computer

Table 15 report for IMM NetApp -Test Case Sequence

Step #	Step Details	Expected Results	Actual Results	Result per step
1	*Start NEF emulator and NetApp instances.*	*1) Emulator started, UEs are moving* *2) NetApp connects to emulator and clean existing subscriptions*	*As expected*	*Pass*
2	*Start vApp instances*	*Both vApp started, users can see each other camera in AR.*	*As expected*	*Pass*
3	*vApps automatically create high-level request to monitor UE locations and QoS*	*High-level requests sent to NetApps*	*As expected*	*Pass*
4	*NetApp receives requests, create corresponding endpoints and send API calls to NEF*	*Endpoints created, API call triggered and Flask server ready to receive notifications*	*As expected*	*Pass*
5	*Upon entering a new cell, the monitored UE trigger location and QoS notifications. NetApp receives the POST request and send high-level msg to their vApp.*	*High-level message created by NetApp and sent to vApp*	*As expected*	*Pass*
6	*vApp receives msg and display its content in AR to notify end-users*	*High-level msg displayed in AR*	*As expected*	*Pass*

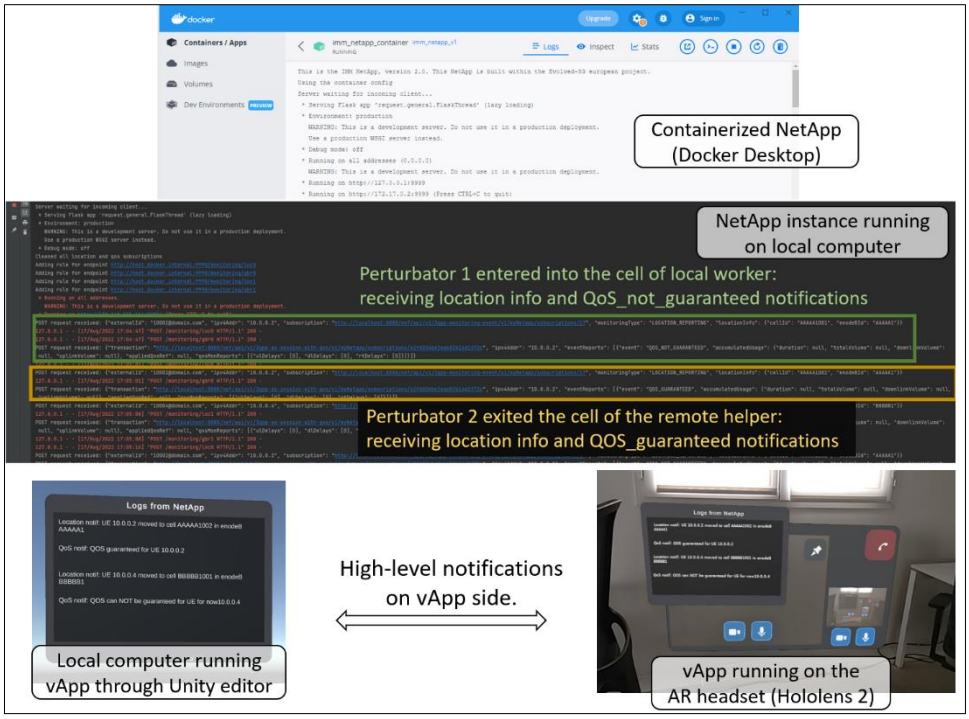


Figure 17 Screenshots of the IMM manual test execution

Both NetApps instances receive requests from the vApp instances, communicate with the emulator and send back high-level notifications to vApps, as shown in Figure 17.

3.1.2.1.2 Test report- Chatbot Assistance NetApp

The validation test was performed using the virtual environment existing in the NEF Emulator while the configurations were based on dummy data stored in a dummy database created for the purpose of assisting the developments.

Table 16 Test report for INF NetApp-General information

Test Case ID	INF-Chatbot Assistance-01
Test Case Description	As different worker authorization levels exist and also the factory is considered to be sectioned in cells, there is a different list of machines that will appear on each worker's chatbot according to his location and his authorization level. The SUT is the local deployment of the NEF emulator, NetApp and vertical application. The targeted KPI stemming from the test case is the return of the expected machines' list related to the correct cell as defined by the factory administrator.
Created By	*Sakkas Christos*
Tester's Name	*Sakkas Christos*
Date Tested	*17-05-2022*

Validation Results	Pass

Table 17 lists the preconditions that need to apply for the execution of the test. Additionally, Figure 18 shows the NEF emulator set-up. Four different factory cells were used, according to the four 5G antennas that are placed in the virtual factory area. Each cell could include multiple factory areas and the configurations are shown in Figure 19. Consequently, each factory area includes multiple machines with only the ones with status 3 flagged as malfunctioning (Figure 20).

Table 17 Test report for INF NetApp Pre-Conditions

S #	Prerequisites:
1	NEF Emulator-NetApp are locally installed
2	End-to-end (network) communication between the vApp - NetApp - NEF Emulator
3	Create the predefined scenario in NEF Emulator with cells and UEs representing workers
4	Machine assignment information in different cells inside the factory, stored in the Database

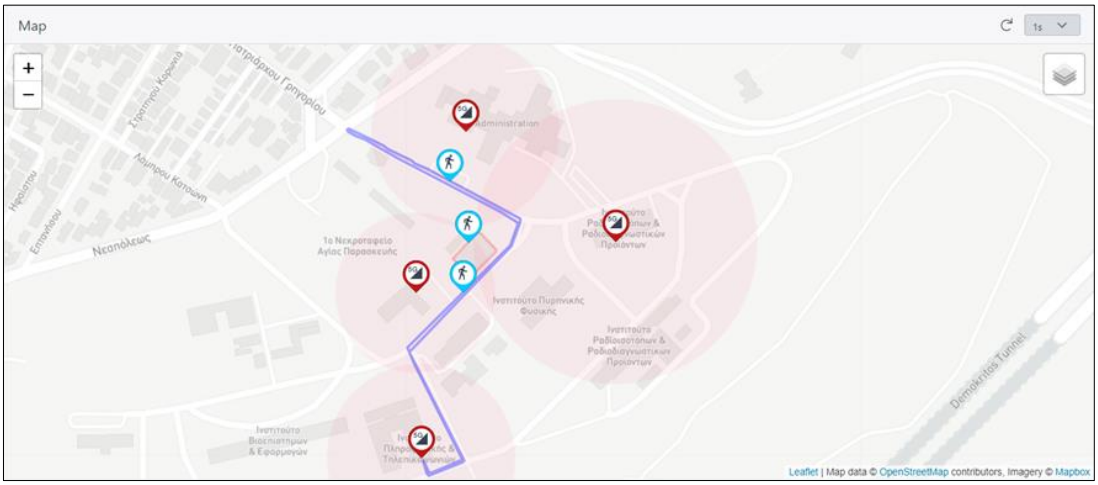


Figure 18 NEF Emulator set-up for the INF validation test

areas (5r × 5c)

id	name	antenna	coordinates	active
1	Area_1	Antenna_1	x1,y1	1
2	Area_4	Antenna_1	x4,y4	1
3	Area_2	Antenna_2	x2,y2	1
4	Area_3	Antenna_1	x3,y3	1
5	Area_5	Antenna_3	x1,y2	1

Figure 19 Dummy database table for factory cells

machines (7r × 5c)

id	name	area	status	active
1	machine_1	1	1	1
2	machine_2	2	1	1
3	machine_3	1	3	1
4	machine_4	3	2	1
5	machine_5	3	3	1
6	machine_7	2	3	1
7	machine_8	2	3	1

Figure 20 Dummy database table for area machines

To initiate the test, we consider a worker placed inside of a specific factory cell. The user types "**work"** in the chatbot and this initiates the procedure of searching for malfunctioning machines under the cell area. After searching the database, the results are displayed on the screen (Figure 21).

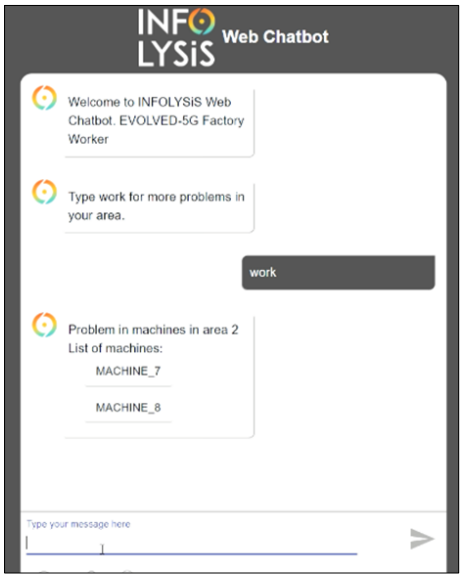


Figure 21 Example of the chatbot test

The validation test sequence is provided in Table 18 and the overall result was successful. To support the results, a screenshot of what is happening behind the scenes is also provided in Figure 22.

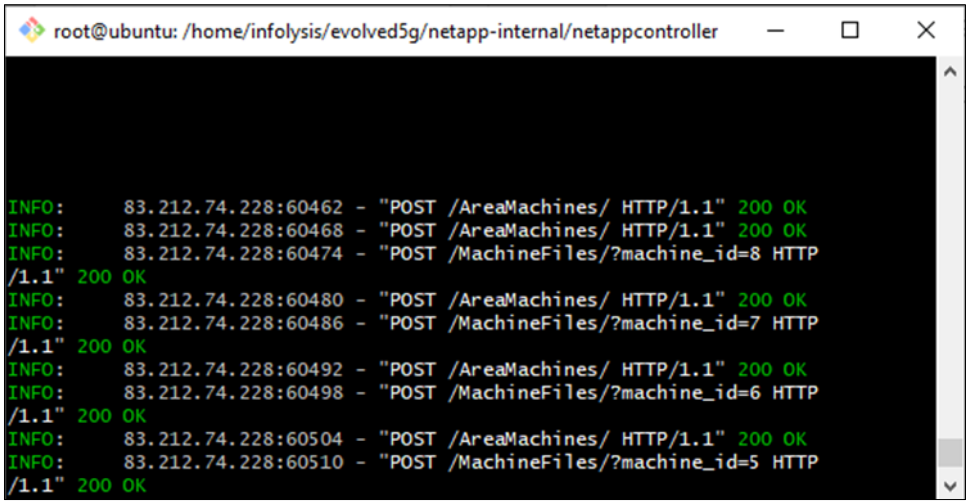


Figure 22 Results of post requests during the validation test

Table 18 Test report or INF NetApp -Test Case Sequence

Step #	Step Details	Expected Results	Actual Results	Result per step
1	*User types "**work**" in Chatbot App. This command is transmitted to the Web Chatbot Backend*	*Web Chatbot backend receives the work command from client successfully*	*As expected*	*Pass*
2	*The Web Chatbot Backend calls the endpoint of the NetApp with the appropriate worker id.*	*Web chatbot backend receives 200 OK response*	*As expected*	*Pass*
3	*The NetApp calls the endpoint of NEF emulator and retrieves the area (cell id) of the worker*	*NetApp receives a 200 OK response*	*As expected*	*Pass*
4	*The NetApp accesses the database and retrieves the machines under the area of the worker according to the worker id.*	*Retrieve machines successfully from database*	*As expected*	*Pass*
5	*The NetApp returns the list to the vApp.*	*vApp receives successfully the list*	*As expected*	*Pass*

6	*The vApp displays the list on the screen and if the list is what expected the validation process ends*	*vApp displays the expected list on the screen*	*As expected*	*Pass*

3.1.2.1.3 Digital/Physical Twin NetApp

The test case has been performed within the virtual environment of the NEF Emulator using the additional tools and installations, as described in the test case template. The test was successful, and the details are presented in the following table.

Table 19 Test report for GMI NetApp-General information

Test Case ID	*GmiAeroNetApp*
Test Case Description	*Test of the Location KPI*
Created By	*GMI AERO*
Tester's Name	*SAUER Marc-Olivier*
Date Tested	*01/06/2022*
Test Case Results	***Pass***

Table 20 Test report for GMI NetApp Pre-Conditions

S #	Prerequisites:
1	*From a container, Running NEF emulator with specific scenario* • *2 UE's* • *2 Cells*
2	*From a container, Running the NetApp on the same computer*
3	*vApp running on the same computer for the test*

Table 21 Test report or GMI NetApp -Test Case Sequence

Step #	Step Details	Expected Results	Actual Results	Result per step
1	*Run the NEF Emulator and*	*UE are moving on the path and alternate on 2 cells*	*As expected*	*Pass*
2	*Run NetApp*	*NetApp is waiting vApp request*	*As expected*	*Pass*
3	*Run vApp and call Location request function*	*Retrieve the Location data*	*As expected*	*Pass*
4	*Compare Location with programmed location of the Cell in the NEF Emulator*	*Location data is the same*	*As expected*	*Pass*

The EVOLVED5G Digital-Physical twin can be accessed from the menu of the NetApp as can be seen in Figure 23. The steps of the process that are described in the table above, are illustrating in Figures 24-28 below.

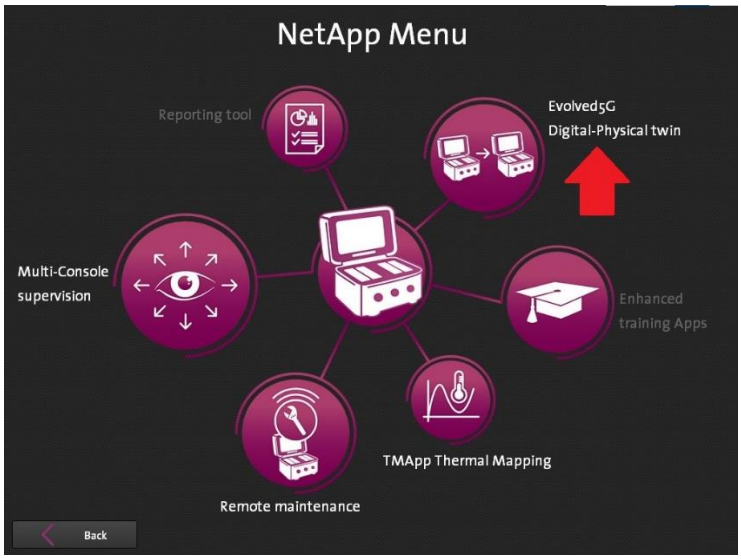


Figure 23 Access Evolved5G screen from menu

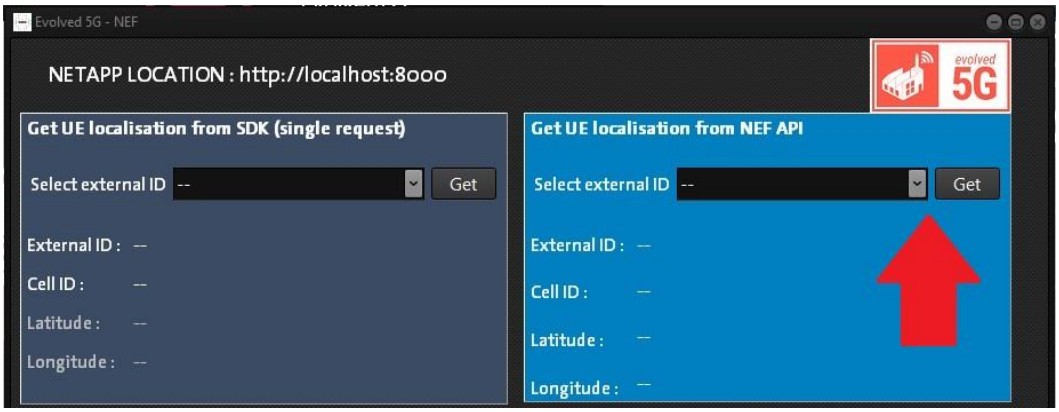


Figure 24 Select UE external ID

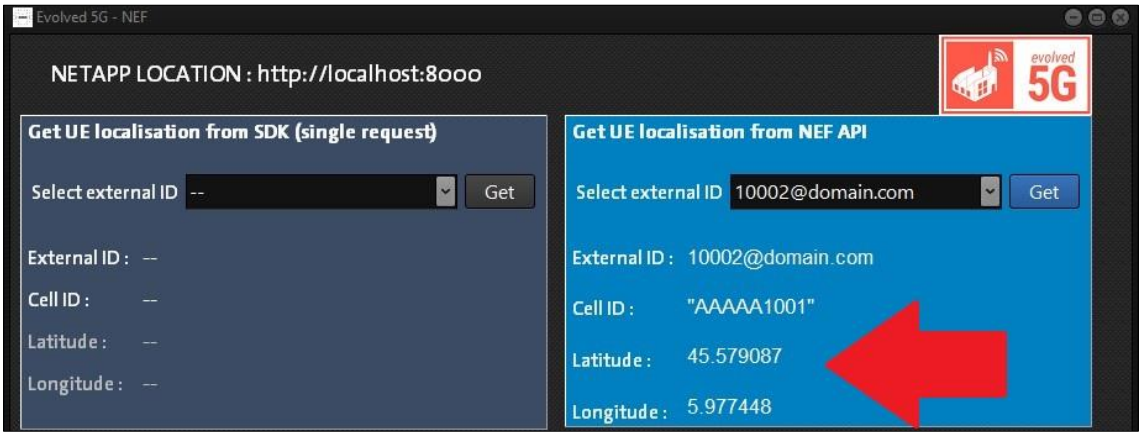


Figure 25 Retrieve UE Latitude and Longitude

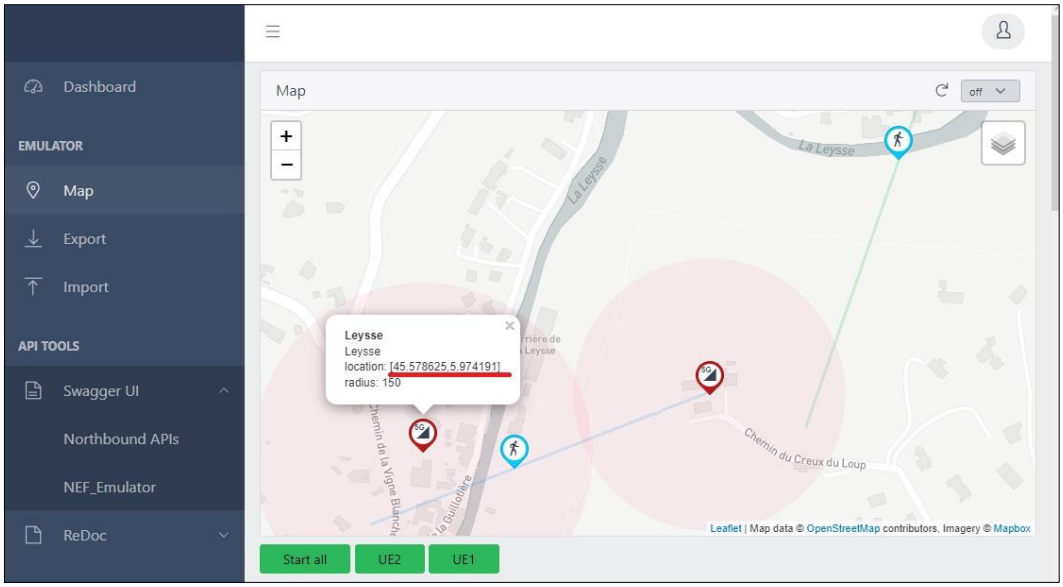


Figure 26 Compare with coordinates on NEF Emulator

Figure 27 Retrieve the exact position of UE from another system

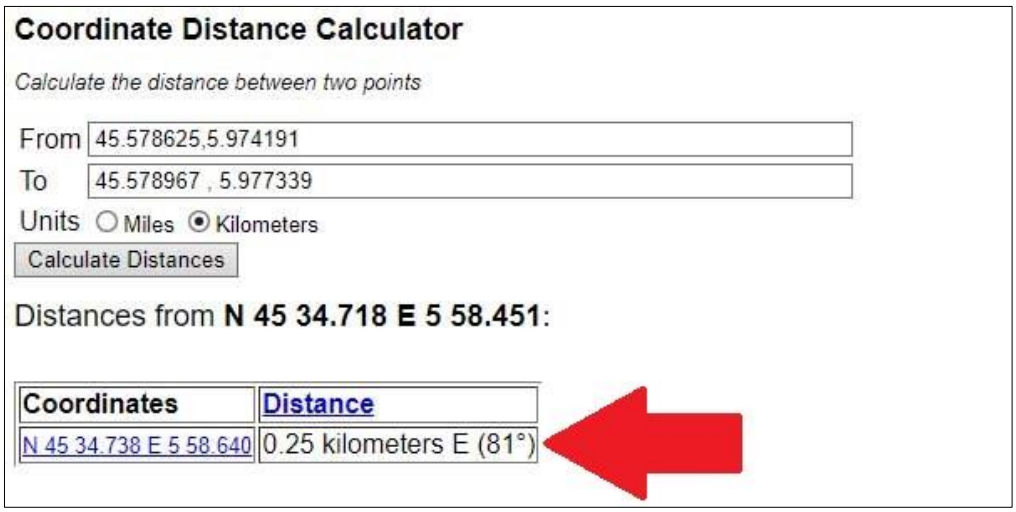


Figure 28 Obtain the LOC-KPI (passed)

3.1.2.2 Efficiency in FoF Operations (FoF) pillar

3.1.2.2.1 Occupational safety analysis NetApp

For the NetApp related to the Occupational safety analysis, the result of the test was successful and the details are presented in the following table.

Table 22 Test report CAF NetApp-General information

Test Case ID	*CAFA-NetMapper-01*
Test Case Description	*NetApp receives QoS messages from NEF Emulator and forwards them to vApp which changes its computer vision behaviour according to the QoS state.*
Created By	*Märten Rannu*
Tester's Name	*Märten Rannu*
Date Tested	*24-05-2022*
Test Case Results	***Pass***

Table 23 Test report CAF NetApp Pre-Conditions

S #	Prerequisites:
1	*NEF Emulator 1.4.1 installed locally. Scenario consists of 3 cells, 2 UE's and 1 path.*
2	*NetApp (CAFA-NetMapper) running in container in local computer*
3	*vApp (CAFA-CV Computer Vision application SafeLyzer) running in container in local computer*
4	*IP camera connected to local network*

Table 24 Test report CAF NetApp-Test Case Sequence

Step #	Step Details	Expected Results	Actual Results	Result per step
1	*NetApp creates QoS awareness subscription to NEF Emulator for UE 10.0.0.1*	*NetApp's terminal prints created subscription*	*As expected*	*Pass*
2	*NetApp's Flask server receives QoS change notifications from the NEF Emulator*	*NetApp's Flask server's terminal prints received message*	*As expected*	*Pass*
3	*NetApp's Flask server sends the message to the vApp's Flask server*	*vApp's Flask server's terminal prints received message*	*As expected*	*Pass*
4	*vApp receives the message and changes its computer vision behaviour*	*When message is "QOS_NOT_GUARANTEED", the marker detection output video frame has text "Detection not available" on its top left corner. When*	*As expected*	*Pass*

		"QOS_GUARANTEED", then it has the camera coordinates.		

The case has been performed within the virtual environment of the NEF Emulator using the additional tools and installations, as described in the test case template. The test was successful, and the details are presented in the following figures.

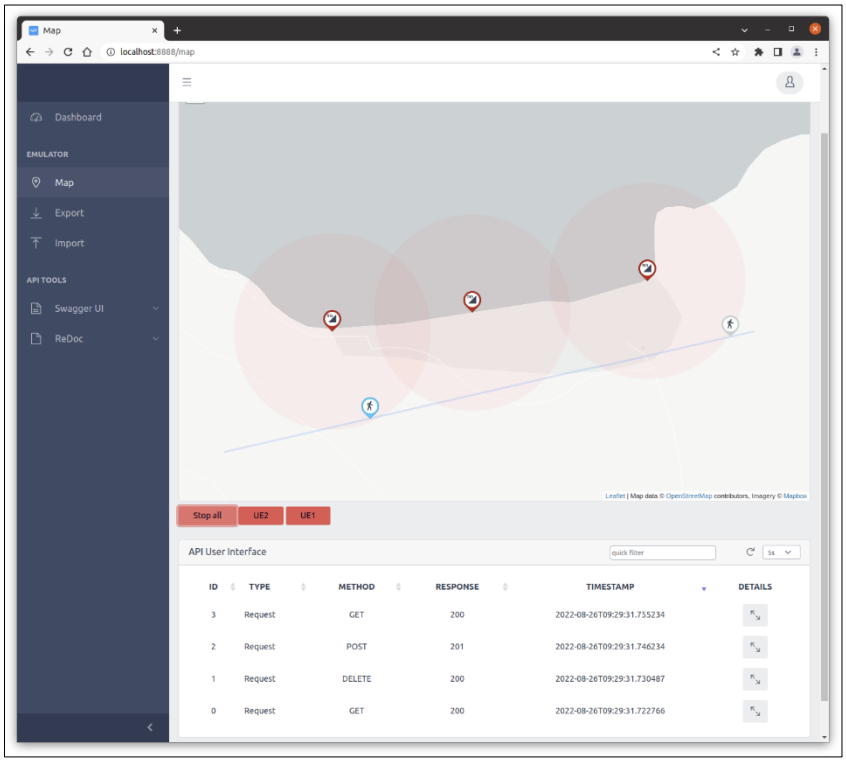


Figure 29 Test case scenario in the NEF Emulator Map view

```
backend_1         | INFO:     172.20.0.1:41198 - "GET /api/v1/UEs?skip=0&limit=100 HTTP/1.1" 200 OK
backend_1         | WARNING:root:UE(202010000000001) with ipv4 10.0.0.1 handovers to Cell 3, Building 1
backend_1         | WARNING:root:Monitoring Event subscription not found
backend_1         | WARNING:root:QOS_NOT_GUARANTEED
mongo_1           | {"t":{"$date":"2022-08-26T09:34:18.290+00:00"},"s":"I",  "c":"NETWORK",  "id":22943,   "ctx":"listener","msg":"Connection accepted","attr":{"remote":"172.20.0.4:51444","connectionId":109,"connectionCount":1}}
mongo_1           | {"t":{"$date":"2022-08-26T09:34:18.290+00:00"},"s":"I",  "c":"NETWORK",  "id":51800,   "ctx":"conn109","msg":"client metadata","attr":{"remote":"172.20.0.4:51444","client":"conn109","doc":{"driver":{"name":"PyMongo","version":"4.1.1"},"os":{"type":"Linux","name":"Linux","architecture":"x86_64","version":"5.15.0-46-generic"},"platform":"CPython 3.9.7.final.0"}}}
mongo_1           | {"t":{"$date":"2022-08-26T09:34:18.293+00:00"},"s":"I",  "c":"NETWORK",  "id":22943,   "ctx":"listener","msg":"Connection accepted","attr":{"remote":"172.20.0.4:51446","connectionId":110,"connectionCount":2}}
mongo_1           | {"t":{"$date":"2022-08-26T09:34:18.294+00:00"},"s":"I",  "c":"NETWORK",  "id":51800,   "ctx":"conn110","msg":"client metadata","attr":{"remote":"172.20.0.4:51446","client":"conn110","doc":{"driver":{"name":"PyMongo","version":"4.1.1"},"os":{"type":"Linux","name":"Linux","architecture":"x86_64","version":"5.15.0-46-generic"},"platform":"CPython 3.9.7.final.0"}}}
mongo_1           | {"t":{"$date":"2022-08-26T09:34:18.294+00:00"},"s":"I",  "c":"NETWORK",  "id":22943,   "ctx":"listener","msg":"Connection accepted","attr":{"remote":"172.20.0.4:51448","connectionId":111,"connectionCount":3}}
mongo_1           | {"t":{"$date":"2022-08-26T09:34:18.295+00:00"},"s":"I",  "c":"NETWORK",  "id":51800,   "ctx":"conn111","msg":"client metadata","attr":{"remote":"172.20.0.4:51448","client":"conn111","doc":{"driver":{"name":"PyMongo","version":"4.1.1"},"os":{"type":"Linux","name":"Linux","architecture":"x86_64","version":"5.15.0-46-generic"},"platform":"CPython 3.9.7.final.0"}}}
mongo_1           | {"t":{"$date":"2022-08-26T09:34:18.303+00:00"},"s":"I",  "c":"ACCESS",   "id":20250,   "ctx":"conn110","msg":"Authentication succeeded","attr":{"mechanism":"SCRAM-SHA-256","speculative":true,"principalName":"root","authenticationDatabase":"admin","remote":"172.20.0.4:51446","extraInfo":{}}}
backend_1         | CRITICAL:root:{'value': 82, 'type': 'DC-GBR', 'priority': 19, 'PDB': 10, 'PER': 0.0001, 'MDBV': 255, 'service': 'Discrete Automation'}
backend_1         | CRITICAL:root:DC-GBR
backend_1         | CRITICAL:root:http://172.17.0.2:5555/monitoring/callback
backend_1         | CRITICAL:root:http://localhost:8888/nef/api/v1/3gpp-as-session-with-qos/v1/CAFA-NetMapper/subscriptions/6308927b8826e33bf3304242
backend_1         | CRITICAL:root:Before response
^[[3;3~backend_1         | Inside qos_reference_match at qosInformation.py {'value': 82, 'type': 'DC-GBR', 'priority': 19, 'PDB': 10, 'PER': 0.0001, 'MDBV': 255, 'service': 'Discrete Automation'}
backend_1         | INFO:     172.20.0.1:41198 - "GET /api/v1/UEs?skip=0&limit=100 HTTP/1.1" 200 OK
backend_1         | INFO:     172.20.0.1:41198 - "GET /api/v1/UEs?skip=0&limit=100 HTTP/1.1" 200 OK
backend_1         | INFO:     172.20.0.1:41198 - "GET /api/v1/utils/monitoring/last_notifications?id=3 HTTP/1.1" 200 OK
backend_1         | INFO:     172.20.0.1:41198 - "GET /api/v1/UEs?skip=0&limit=100 HTTP/1.1" 200 OK
```

Figure 30 Test case scenario in the NEF Emulator Map view

3.1.2.2.2 Smart irrigation 5G Agriculture NetApp

Considering that the UMA-CSIC NetApp development team and the Málaga validation platform administrators are closely related, it was decided to make use of the Open5Genesis Framework for the realization of this validation, which has been fully automated.

The following tables show a detailed view of the main objectives of the test along with the steps that form the test case, leading to the individual results for each of them.

Table 25 Test report UMA NetApp -General information

Test Case ID	*UMA-Smart Irrigation NetApp-01*
Test Case Description	*The objective of the following functional test is to assess the correct functionality of the environment composed by NEF Emulator, NetApp and the Vertical Applications. Considering the complete set of actions that cover the expected use of the NetApp in a real environment and the proper behaviour of both Vertical Applications when they are integrated with the NetApp.* *The target KPI/KVI is to run the test sequence 25 repeated times, and if all Iterations behave as expected then the test is Successful.* *In order to consider that an iteration is Successful:* 1. *All steps in the **Test Case Sequence** need to be performed.* 2. *The output of the execution of all steps is as expected.* *Reliability is a secondary measurement. Its definition is: The ratio of successful vs. total number of iterations.*
Created By	*Rafael López Gómez*
Tester's Name	*Rafael López Gómez*
Date Tested	*04-05-2022*
Test Case Results	**Pass**

Table 26 Test report UMA NetApp Pre-Conditions

S #	Prerequisites:
1	*The NEF emulator is running and accessible.*
2	*The NEF emulator is configured with three different cells and three devices, one for each cell.*

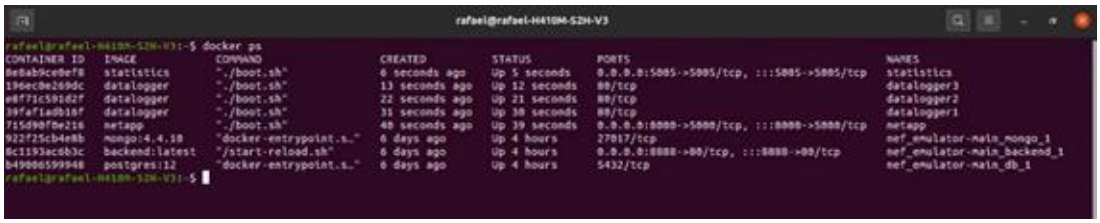


Figure 31 UMA-CSIC NetApp prerequisites and components running during a test

Table 27 Test report UMA NetApp -Test Case Sequence

Step #	Step Details	Expected Results	Actual Results	Result per step
1	*The NetApp, and both vApps containers are created and started*	*Containers starts successfully*	*As Expected*	*Pass*
2	*Using the "Cell Management" endpoints, two crop areas are defined, with the same configuration of the cells previously configured in the NEF emulator except for one (This is for test updating methods)*	*Both cells are stored in the NetApp database*	*As expected*	*Pass*
3	*Datalogger vApp will read a CSV file with known data and POST some historics using "Historic Management endpoints" from NetApp. This endpoint will use the Monitoring Event API (with one time request) to retrieve the cell id of the stationary UEs (dataloggers)*	*Historics are created in the NetApp database*	*As expected*	*Pass*
4	*Using the "Historic Management" endpoints, GET the historics registered in the NetApp and confirm that the result is the expected*	*GET calls retrieve the historics correctly with a 200 OK response and the data is confirmed to be correct*	*As expected*	*Pass*
5	*Use prediction function from statistics vApp endpoints and test if it is the expected output*	*The prediction function answer is the expected*	*As expected*	*Pass*

6	*Verify that others function endpoints, such as the temperature and humidity, of the statistics App work properly*	*Assess that the function endpoints generate the correct calculations*	*As expected*	*Pass*
7	*UPDATE one cell number with the same identifier of the other cell. An error is expected*	*Get a callback with a 500 error response*	*As expected*	*Pass*
8	*GET the historics attached to a specific cell. Then, UPDATE the cell number (with the cell unused in NEF-Emulator) of the cell selected in this step. Finally, GET again the historics attached to that cell*	*Both responses have the same data due to they are historics of the same cell (even though the ID has changed)*	*As expected*	*Pass*
9	*UPDATE data of one historic and check that it is done correctly*	*Data of the historic has been updated*	*As expected*	*Pass*
10	*DELETE another cell using "Cell Management " and confirm the cell has been deleted with its attached historics*	*Obtain a 404 - Not Found error when querying data after the cell has been deleted*	*As expected*	*Pass*
11	*DELETE one historic of the last existing cell and confirm that it is deleted*	*The historic is not found in the list of historics*	*As expected*	*Pass*
12	*RESET database using "Database utils" endpoint. Verify that the database is cleared*	*GET endpoints of cells and historics answers with an empty list*	*As expected*	*Pass*
13	*Both NetApp and vApp containers are destroyed*	*All containers are stopped and removed*	*As expected*	*Pass*

For the case of UMA-CSIC the validation of the NetApp has been performed in an automatic manner using the Open5Genesis Framework. The processes that are running on top of the ELCM are illustrated in Figure 32.

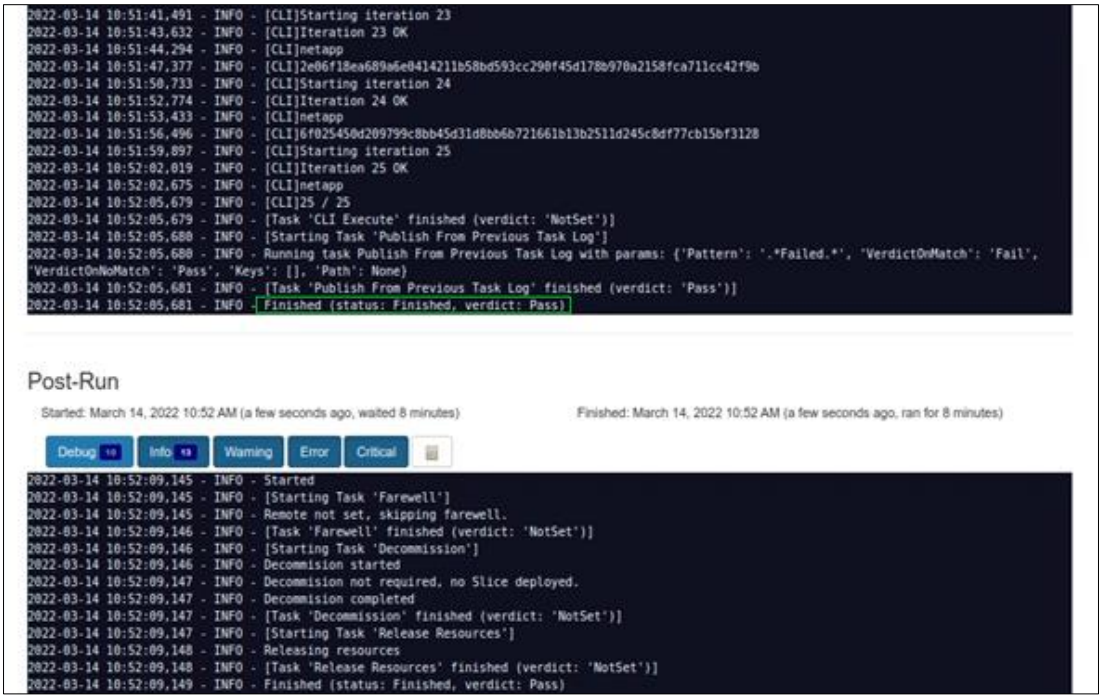


Figure 32 UMA-CSIC NetApp validation running on the ELCM

3.1.2.2.3 Industrial Grade 5G connectivity NetApp

The test case has been performed within the virtual environment of the NEF Emulator, while the data flow required for the test execution have been based on synthetically generated data. The test was successful, and the details are presented in the following table.

Table 28 Test report ININ NetApp- General information

Test Case ID	*ININ-IoT Assurance Functional Test-01*
Test Case Description	*Main target of the use case is that UE is subscribed to 5G connectivity service with assured SLA/SLS and with corresponding QoS profile that is configured by the vertical application (IoT Management UI) through the NetApp for specific UE (IoT Gateway) and corresponding "external id". This way, the IoT Management can dynamically enable or disable services running on the UE (5G IoT Gateway) depending on the current QoS monitoring parameters levels received from 5G NEF and targeted SLS thresholds.*
Created By	*Luka Korsic*
Tester's Name	*Luka Korsic*
Date Tested	*26.04.2022*
Test Case Results	***Pass***

Table 29 Test report ININ NetApp Pre-conditions

S #	Prerequisites:
1	*IoT Management, IoT collector and IoT gateway are deployed and operational*
2	*Predefined scenario in NEF emulator to trigger "Session with QoS" notifications*
3	*SUT feature - IoT Management API to configure QoS profile*

4	*SUT feature - traffic services on the 5G IoT Gateway (critical traffic - sensor data collection, BE traffic – e.g., video)*
5	*SUT feature - collection of KPIs from monitoring events received by NEF will be stored in the IoT Collector*

Table 30 Test report ININ NetApp- Test case Sequence

Step #	Step Details	Expected Results	Actual Results	Result per step
1	*Provision "external id" and "qos profile" for selected UE in IoT management UI*	*UE is provisioned with "external id" and "qos profile" in IoT Management UI*	*As expected*	*Pass*
2	*IoT Management UI registers "external id" in the 5G NEF via NetApp*	*UI is registered in 5G NEF via NetApp, 200 OK is received from NEF*	*As expected*	*Pass*
3	*NetApp subscribes to "Monitor Event API" in the 5G NEF*	*Monitor Event API subscription confirmation received from NEF*	*As expected*	*Pass*
4	*NetApp subscribes to "Session with QoS API" in the 5G NEF*	*Session with QoS API subscription confirmation received from NEF, UE is assigned the correct QoS profile*	*As expected*	*Pass*
5	*Start gateway/UE services via IoT Management*	*Critical traffic (e.g., sensor data) and BE traffic (e.g.) is forwarded by the IoT Gateway*	*As expected*	*Pass*
6	*IoT Management signals to UE to disable "BE" traffic to save bandwidth for critical traffic based on "QoS NOT GUARANTEED" notification received*	*BE traffic forwarding is suspended*	*As expected*	*Pass*
7	*IoT Management signals to UE to enable "BE" based on "QoS GUARANTEED" notification received*	*BE traffic forwarding is resumed*	*As expected*	*Pass*

Each single step of the test sequence can be further illustrated in the following Figures:

- step 1: Provision "external id" and "qos profile" for selected UE in IoT management UI

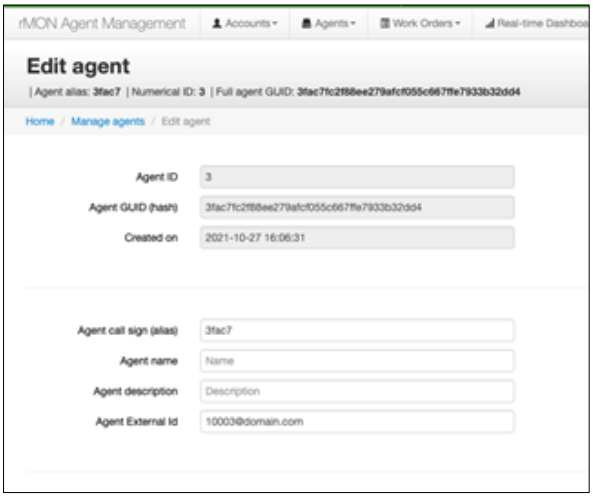


Figure 33 Provision "external id" and "qos profile" for selected UE

- **steps 2 to 4:** IoT Management UI registers "external id" in the 5G NEF via NetApp
- NetApp subscribes to "Monitor Event API" in the 5G NEF
- NetApp subscribes to "Session with QoS API" in the 5G NEF

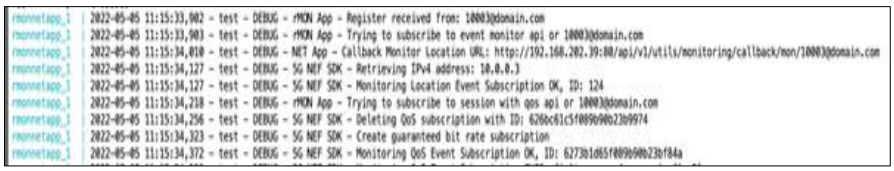


Figure 34 Subscription to Monitoring event and AsSessionwithQoS API

Figure 35 Initiation of the gateway/UE services via IoT Management

- **step 5:** Start gateway/UE services via IoT Management can be seen in Figure 34 above.
- **steps 6 and 7:** IoT Management signals to UE to disable "BE" traffic to save bandwidth for critical traffic based on "QoS NOT GUARANTEED" notification received,
- IoT Management signals to UE to enable "BE" based on "QoS GUARANTEED" notification received, as depicted in Figure 36.

Figure 36 QoS Guaranteed and QoS_NOT_Guaranteed notifications

3.1.2.2.4 Anomaly Detection NetApp

The testing and validation of the Anomaly Detection NetApp took place on Zortenet's premises while the NetApp and the NEF emulator are deployed on a local docker environment.

Table 31 Test report ZORT NetApp -General Description

Test Case ID	*Zortenet – Anomaly Detection NetApp*
Test Case Description	*Anomaly Detection*
Created By	*Andreas Oikonomakis*
Tester's Name	*Andreas Oikonomakis*
Date Tested	*07-06-2022*
Test Case Results	***Pass***

Table 32 Test report ZORT NetApp Pre-conditions

S #	Prerequisites:
1	*vApp – NetApp – NEF emulator are deployed and running locally*
2	*NEF Emulator has the default scenario.*
3	*No policies are enforced (database of NetApp is empty)*

Table 33 Test report ZORT NetApp -Test case sequence

Step #	Step Details	Expected Results	Actual Results	Result per step
1	*Vertical App users create a subscription for one or multiple UEs.*	*Http code 200 from NetApp*	*As expected*	*Pass*
2	*NetApp receives the subscription request then forwards the request to NEF emulator using the /nef/api/v1/3gpp-monitoring-event/v1/netApp/subscriptions*	*Http code 200 from NEF Emulator*	*As expected*	*Pass*
3	*NetApp is receiving callbacks from NEF emulator using the registered callback endpoint.*	*Http code 200 from NEF Emulator on the endpoint that receives the callbacks*	*As expected*	*Pass*
4	*Vertical App user enforces a policy regarding which cells the specific UE is allowed to operate.*	*Http code 200 from NetApp*	*As expected*	*Pass*
5	*NetApp receives and stores the policy.*	*Check the policy db that the policy record exists from the dedicated endpoint*	*As expected*	*Pass*
6	*NetApp is performing anomaly detection using applied policies.*	*Vertical App UI is receiving logs or alerts*	*As expected*	*Pass*

7	*If any policy is violated alerts are being sent to Vertical App.*	*Vertical App UI is receiving logs or alerts*	*As expected*	*Pass*
8	*The Vertical App user is able to receive alerts of the anomaly.*	*Vertical App UI is receiving logs or alerts*	*As expected*	*Pass*

During the execution of steps 1 and 2, the vApp user creates a subscription for UE 1003@example. The NetApp receives the subscription request and relays it to NEF Emulator. NEF Emulator receives the request and returns a http response with code 200 to NetApp. Finally, this response is reaching the vApp user from the NetApp API. The overall process can be seen in the figure below.

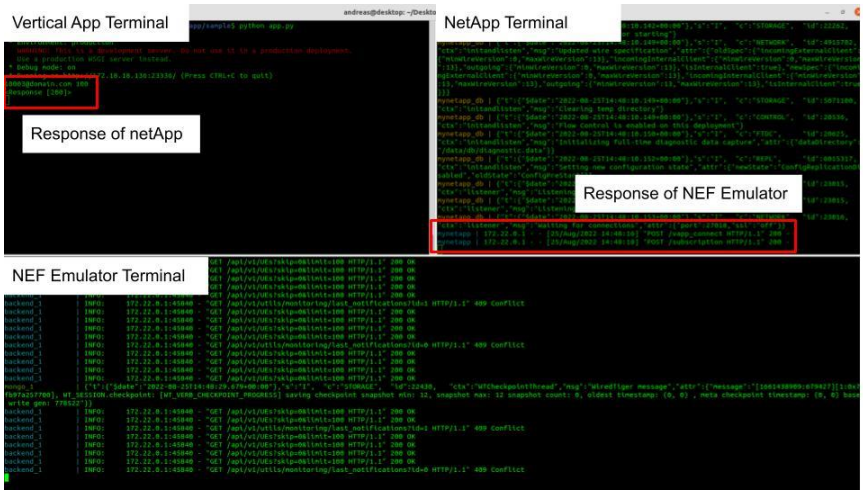


Figure 37 vApp user creates a subscription and the NetApp receives the subscription request

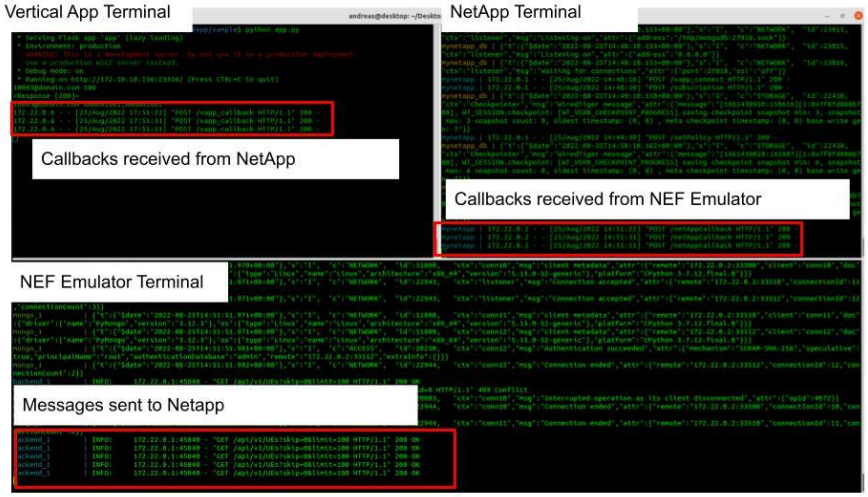


Figure 38 NEF emulator is sending messages to NetApp

Step 3 describes the process where the NEF emulator is sending messages to NetApp about the requested UE. The NetApp receives those messages and relays them to Vertical App as seen in Figure 38.

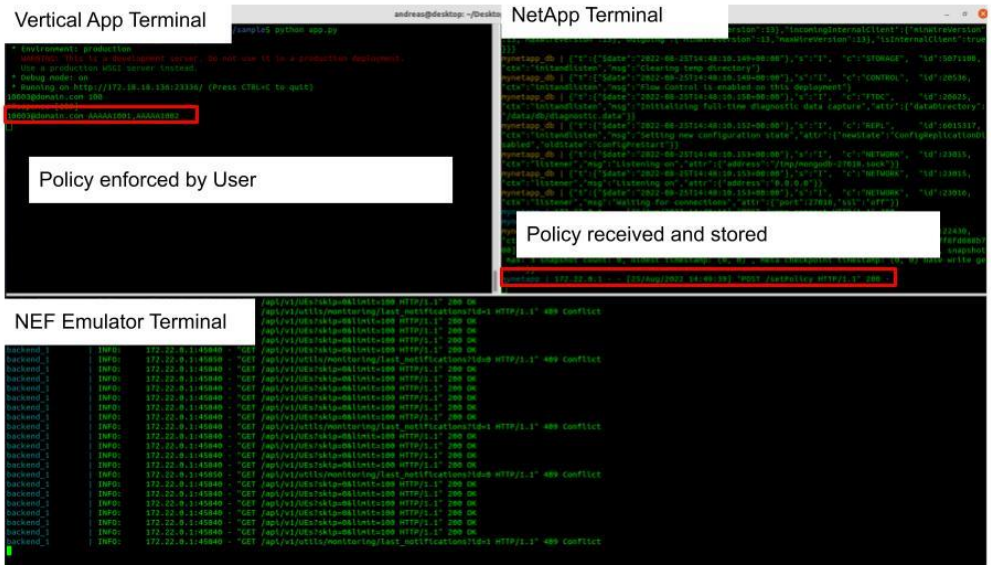


Figure 39 vApp enforces a policy about a specified UE.

During steps 4 and 5 the vApp enforces a policy about the specified UE regarding its cell usages. The specific policy is then received and stored to a local database, as can be seen in the figure above.

The next steps describe the process of the differentiation of the logs from the anomaly detection framework and are displayed to separate panels accordingly, as depicted in Figure 40 and Figure 41.

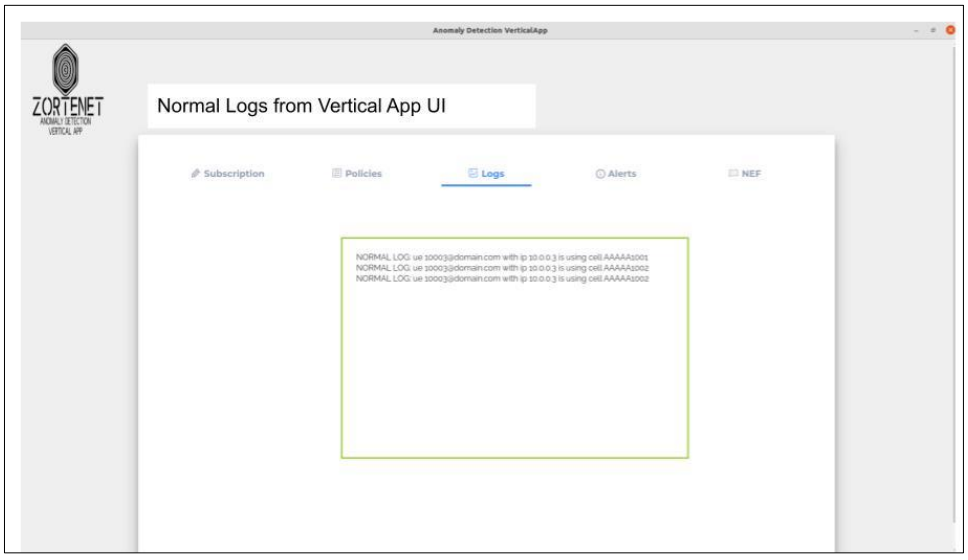


Figure 40 Panel displaying the nornal logs

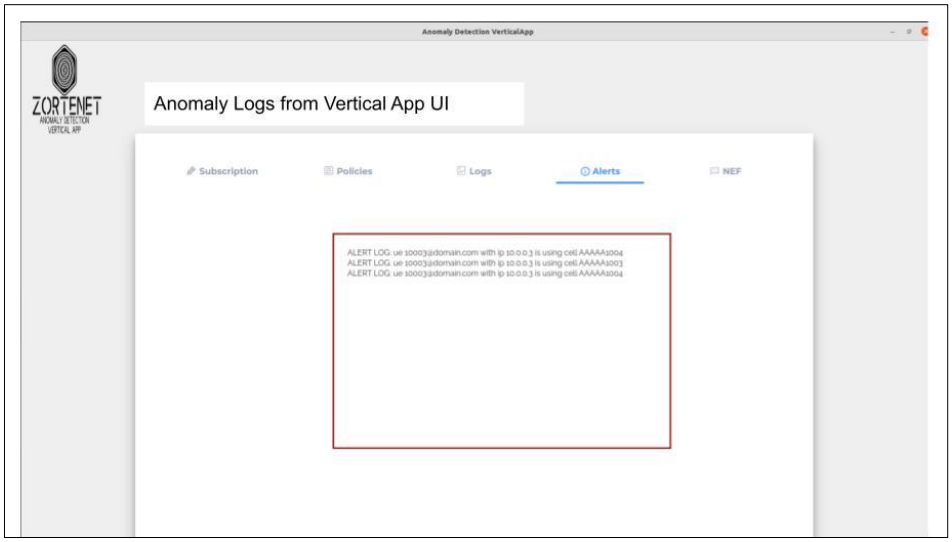


Figure 41 Panel displaying the detected anomaly logs

3.1.2.3 Security Guarantees and Risk Analysis (SEC) pillar

3.1.2.3.1 Traffic Management NetApp

For the Traffic Management NetApp, the test case sequence was followed in order to confirm that the expected behaviour and the actual behaviour of the NetApp are matched. The result of the manual validation test was successful and the details are presented in the following table.

Table 34 Test report 8Bells NetApp -General Description

Test Case ID	*8BELLS – Traffic Management NetApp*
Test Case Description	*Eight Bell's NetApp, being developed for security guarantees risk analysis pillar and in the scope of the EVOLVED-5G project, offers the accurate measurement of traffic over an interface, specific to a device and perform a simple check of "unregistered" traffic outside the 5G Core Network.*
Created By	*Ioannis Margaritis*
Tester's Name	*Ioannis Margaritis*
Date Tested	*07-06-2022*
Test Case Results	***Pass***

Table 35 Test report 8Bells NetApp Pre-conditions

S #	Prerequisites:
1	*NEF emulator - NetApp locally installed*
2	*End to end communication NEF emulator – NetApp - vApp*
3	*vApp's firewall in operation*
4	*Predefined scenario created in NEF emulator*

Table 36 Test report 8Bells NetApp -Test case sequence

Step #	Step Details	Expected Results	Actual Results	Result per step
1	*The NetApp calls the endpoints of the NEF emulator and retrieves a list of all active IP addresses in the network.*	*List of IP addresses is retrieved*	*As expected*	*Pass*
2	*NetApp and vApp communicate via ssh connection.*	*Successful ssh communication NetApp - vApp*	*As expected*	*Pass*
3	*The list is passed to the vApp, which creates the appropriate firewall rules.*	*The list is successfully passed to the vApp*	*As expected*	*Pass*
4	*Any traffic that is not registered in these IP addresses is dropped.*	*Traffic is dropped based on the appropriate firewall rules*	*As expected*	*Pass*
5	*Repeat the process every time a new UE connects to the network.*	*Process repeated successfully*	*As expected*	*Pass*

The results stemming from the execution of the aforementioned process are depicted in Figure the folowing Figures. Figure 42 illustrates the details on the predefined scenario set up and Figure 43 shows the notification as retrived from the NEF emulator.

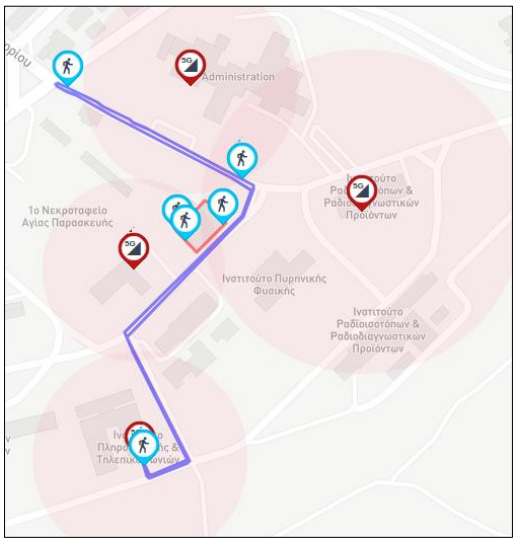


Figure 42 Predefined scenario prepared in NEF Emulator

```
ioannis@ioannis-VirtualBox:~/Desktop/temp/app$ python3 -m flask run --host=0.0.0
.0
 * Serving Flask app '/home/ioannis/Desktop/temp/app/main.py' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployme
nt.
 * Running on http://10.0.2.15:5000/ (Press CTRL+C to quit)
New notification retrieved:
{'transaction': 'http://localhost:8888/nef/api/v1/3gpp-as-session-with-qos/v1/my
Netapp/subscriptions/624443db0417e9432b57bb3a', 'ipv4Addr': '10.0.0.3', 'eventRe
ports': [{'event': 'QOS_NOT_GUARANTEED', 'accumulatedUsage': {'duration': None,
'totalVolume': None, 'downlinkVolume': None, 'uplinkVolume': None}, 'appliedQosR
ef': None, 'qosMonReports': [{'ulDelays': [0], 'dlDelays': [0], 'rtDelays': [0]}
]}]}
172.18.0.4 - - [30/Mar/2022 14:51:23] "POST /monitoring/callback HTTP/1.1" 200 -
New notification retrieved:
{'transaction': 'http://localhost:8888/nef/api/v1/3gpp-as-session-with-qos/v1/my
Netapp/subscriptions/624443db0417e9432b57bb3a', 'ipv4Addr': '10.0.0.3', 'eventRe
```

Figure 43 Notification retrieval from NEF Emulator by the NetApp

3.1.2.3.2 5G-Enabled Security Information and Event Management (SIEM) NetApp

The validation tests of the 5G-Enabled SIEM NetApp were performed locally, on FOGUS's premises, using all necessary additional elements (NEF Emulator and Vertical Application). A set of pre-conditions for the execution of the tests, is defined below:

- NEF Emulator is installed and deployed (as a set of containers) on a host of the local network.
- Deploy a predefined scenario in NEF Emulator, with 8 cells and 1 UE as shown in Figure 44.
- Vertical Application (AlienVault OSSIM with FOGUS Add-ons) is installed and deployed (as a Virtual Machine) on a separate host of the local network as depicted in Figure 45.
- FOGUS NetApp is installed and deployed (as a set of containers) on a separate host of the local network.

Figure 44 Deployed scenario in NEF Emulator

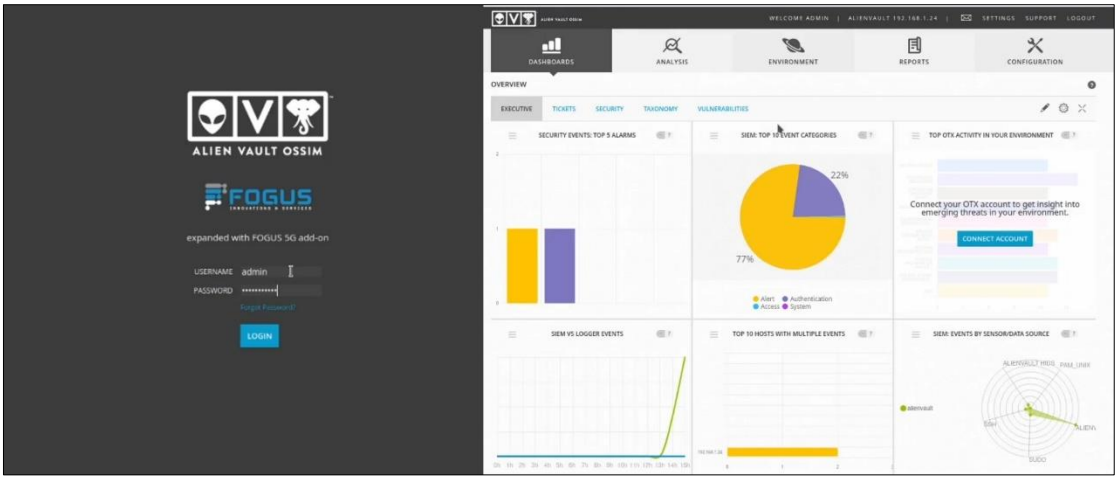


Figure 45 Vertical Application portal

All the tests were executed successfully, and a report is presented in the following table.

Table 37 Test report FOG NetApp -General Description

Test Case ID	*FOG – 5G SIEM*
Test Case Description	*In an industrial environment the department that is in charge of the IT and network systems faces the problem of unifying the various existing IP networks with a 5G NPN that supports mobile devices of the employees and 5G-enabled devices in the production line. The main scope is to interface with the core network of the 5G network and embed the information exchanged with 5G to the factory's security information and event management system (SIEM).*
Created By	*Stavros Charismiadis*
Tester's Name	*Stavros Charismiadis*
Date Tested	*04-05-2022*
Test Case Results	**Pass**

Table 38 Test report FOG NetApp Pre-conditions

Step #	Prerequisites:
1	*NEF Emulator is installed and deployed (as a set of containers) on a host of the local network.*
2	*Deploy a predefined scenario in NEF Emulator, with 8 cells and 1 UE.*
3	*Vertical Application (AlienVault OSSIM with FOGUS Add-ons) is installed and deployed (as a Virtual Machine) on a separate host of the local network.*
4	*FOGUS NetApp is installed and deployed (as a set of containers) on a separate host of the local network.*

Table 39 Test report FOG NetApp-Test case sequence

Step #	Step Details	Expected Results	Actual Results	Result per step
1	*Create a subscription for a*	• *A 201 HTTP Response and a*	*As expected*	*Pass*

	single callback from the NetApp frontend	*MonitoringEventReport object are returned* • *A 5G UE asset is created in OSSIM*		
2	*Create a subscription for up to 50 potential callbacks from the NetApp frontend*	• *A 201 HTTP Response and a MonitoringEventSubscription object are returned* • *50 callbacks appeared in the NetApp Dashboard*	*As expected*	*Pass*
3	*Update of UE location in OSSIM dashboard (after a subscription for a single callback)*	• *The UE location changed also in the OSSIM*	*As expected*	*Pass*

Step 1 "*Create a subscription for a single callback from the NetApp frontend*", step 2 "*callbacks appeared in the NetApp Dashboard*" and step 3 "*Update of UE location in OSSIM dashboard*" are depicted in Figures 46, 47 and 48 respectively.

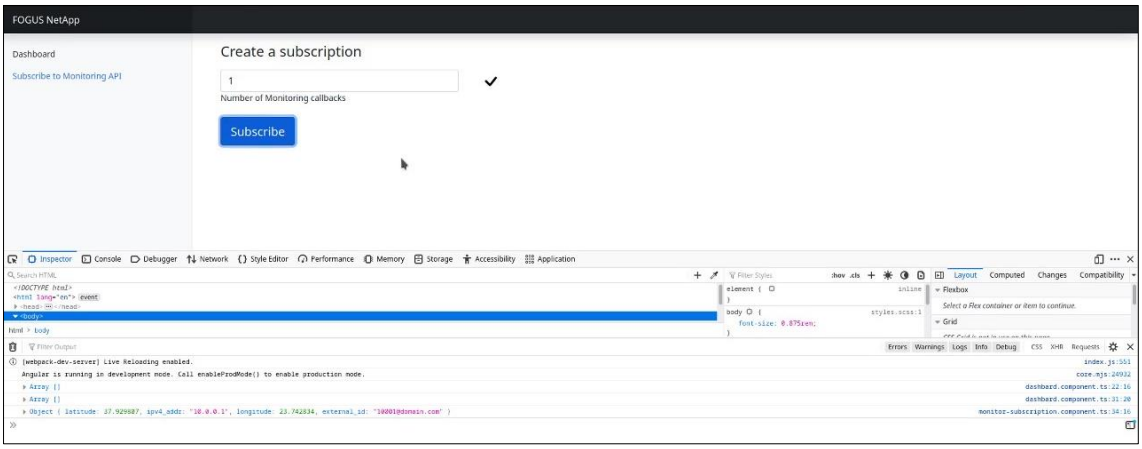


Figure 46 Subscription for a single callback

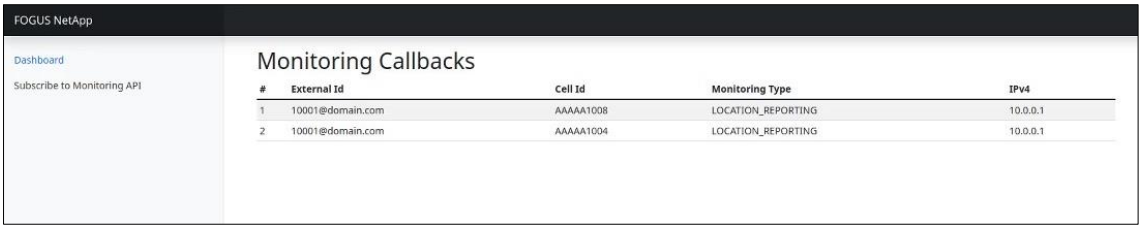


Figure 47 Callbacks returning to NetApp from multi-time monitoring subscription

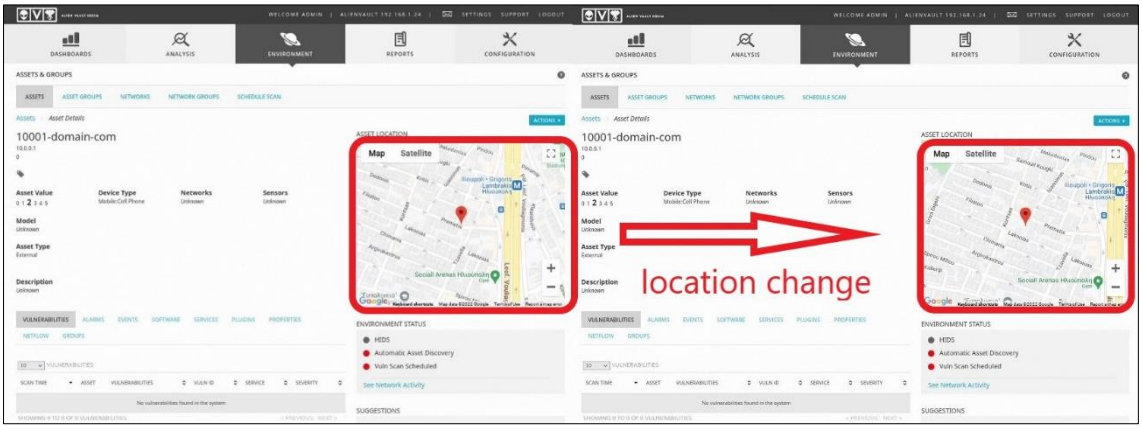


Figure 48 Location update on Vertical Application after receiving a Monitoring Event from NEF Emulator

3.1.2.3.3 ID Management and Access Control NetApp

The test scenario defined in the general information section was run locally, after ensuring the pre-conditions were met. The test case sequence was followed, confirming the expected behavior and the actual behavior of the NetApp matched. Several unit tests were then created in order to facilitate the execution of the test case in the future.

Table 40 Test report IQB NetApp-General information

Test Case ID	*IQB – Identity and Access Management*
Test Case Description	*Secure authentication and authorization involve the following parties: The 3rd Party NetApp, the IQB NetApp, as well as the NEF Emulator. The 3rd Party should be able to authenticate and be authorized to consume the NEF Emulator APIs through the IQB NetApp, using OpenID Connect. The location monitoring subscription created will then generate callbacks that shall be received on a test server. The SUT is the local deployment of the NEF Emulator, NetApp, Keycloak Software as a Service (SaaS), a 3rd Party (Postman) and a test server to receive callbacks.*
Created By	*Ioannis Stylianou*
Tester's Name	*Ioannis Stylianou*
Date Tested	*28-04-2022*
Test Case Results	***Pass***

Table 41 Test report IQB NetApp Pre-Conditions

S #	Prerequisites:
1	*NEF Emulator and NetApp are locally installed on a Virtual Machine and host respectively*
2	*3rd Party NetApp (could be a Postman instance), which must be connected to the same Wi-Fi network with the computer.*
3	*An image of a Keycloak must have been pulled in order to be run inside a docker container (for OpenID Connect functionality)*
4	*A realm (IQB-NetApp), client (client-NetApp) and user (sampleuser) must have been specified in Keycloak by the 3rd Party via Keycloak API consumption or the management console.*
5	*A predefined scenario in NEF Emulator with at least 2 cells and 1 UE switching b*
6	*The emulators, 3rd Party Instance as well as the Keycloak container have been started*
7	*The NetApp has been started and established end-to-end communication between other instances*
8	*The callbacks server has been started*

Table 42 Test report IQB NetApp-Test Case Sequence

Step #	Step Details	Expected Results	Actual Results	Result per step
1	*Attempt to create a location monitoring subscription from the 3rd Party using the respective NetApp endpoint*	*Failure due to unauthorized request*	*As expected*	*Pass*
2	*Login to the NetApp using the /login endpoint*	*Success and returned authorization token which is stored in session*	*As expected*	*Pass*
3	*The 3rd Party creates a location monitoring subscription*	*Success since the 3rd Party has been authenticated*	*As expected*	*Pass*
4	*Start the simulation in NEF Emulator, wait until the UE changes cell ID*	*Callbacks are received when the cell ID of the UE changes*	*As expected*	*Pass*
5	*Logout from the NetApp using /logout endpoint*	*Logout succeeds and the token is removed from session*	*As expected*	*Pass*
6	*Attempt to read active subscriptions*	*Failure due to unauthorized request*	*As expected*	*Pass*

The results of the unit tests that were created to facilitate the testing can be seen in the figure below.

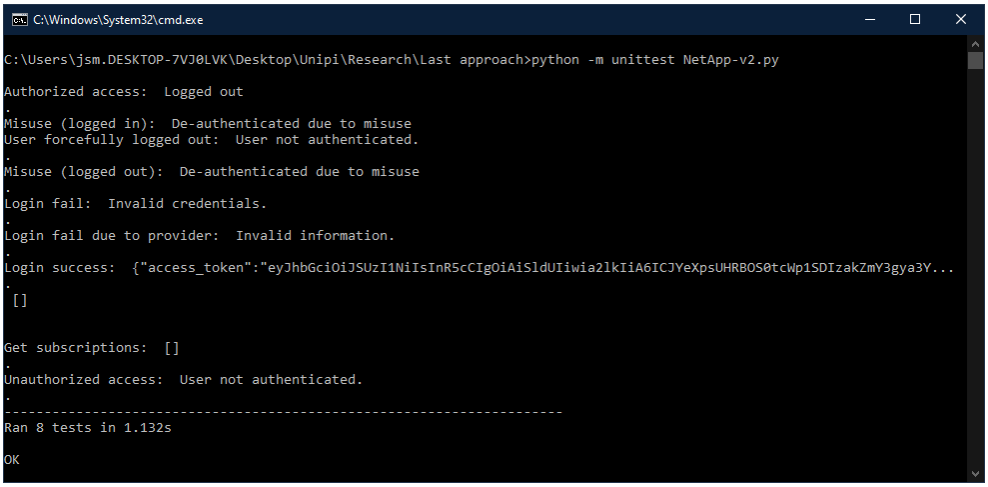


Figure 49 IQB Unit Tests Execution Results

3.1.2.4 Production Line Infrastructure (PLI) pillar

3.1.2.4.1 Localization NetApp

The manual validation tests of the Localization NetApp were performed locally, using all the necessary elements, as defined in the pre-conditions of the test case.

Table 43 Test report UML/PAL NetApp-General information

Test Case ID	*LocalizationNetApp-01*
Test Case Description	*The scope of this test is to check the correct communication between the TiAGo simulated robot, the NetApp and the NEF Emulator. The expected result is that the simulated TiAGo robot changes direction based on the location updates received from the NetApp through the NEF emulator.*
Created By	*Sergi Grau-Moya -Thomas Peyrucain*
Tester's Name	*Sergi Grau-Moya -Thomas Peyrucain*
Date Tested	*18-05-2022*
Test Case Results	***Pass***

Table 44 Test report UML/PAL NetApp Pre-Conditions

S #	Prerequisites:
1	*NEF Emulator-NetApp are locally installed*
2	*End-to-end (network) communication between the vApp - NetApp - NEF Emulator*
3	*Create the predefined scenario in NEF Emulator with cells and UEs representing robots*
4	*PAL docker is installed on the host machine*

The case has been performed within the virtual environment of the NEF Emulator using the additional tools and installations, as described in the test case template. The setup of the NEF emulator and the predefined scenario is presented in the following figure.

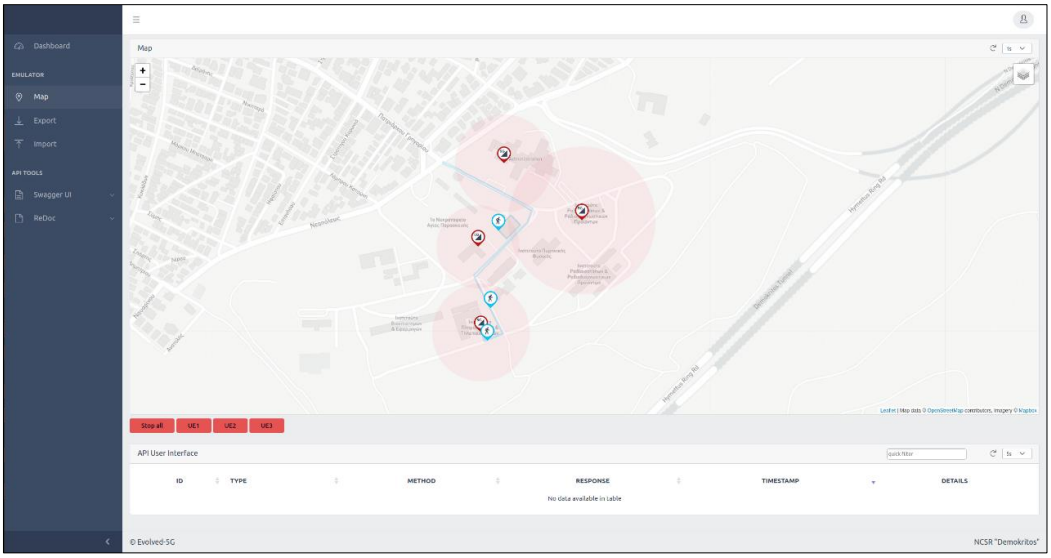


Figure 50 NEF Emulator set-up

Table 45 Test report UML/PAL NetApp -Test Case Sequence

Step #	Step Details	Expected Results	Actual Results	Result per step
1	Start Tiago simulator (via PAL Robotics container)	A Gazebo window is started with a TIAGo robot that publish the right topics	As expected	Pass
2	Run NEF emulator	The emulator is available and spawning with the right Cells and UEs	As expected	Pass
3	Start "device 3" routine	UE 3 should move around the map	As expected	Pass
4	Run flask endpoint	flask endpoint with a valid connection to the emulator	As expected	Pass
5	Run LocalizationNetApp	Creates a valid subscription to the flask endpoint and publishing a topic on /cellid with the proper cell ID number	As expected	Pass
6	Wait until the "device 3" changes	The robot in simulation should change	As expected	Pass

	coverage zone in the emulator	*direction or perform motions when CellID is changing*		

The simulated environment reflecting the positioning of the robot as described in step 3, is depicted in Figure 51. The publishing of the correct cell ID is presented in the command line in the command line interface of the NetApp, as shown in Figure 52.

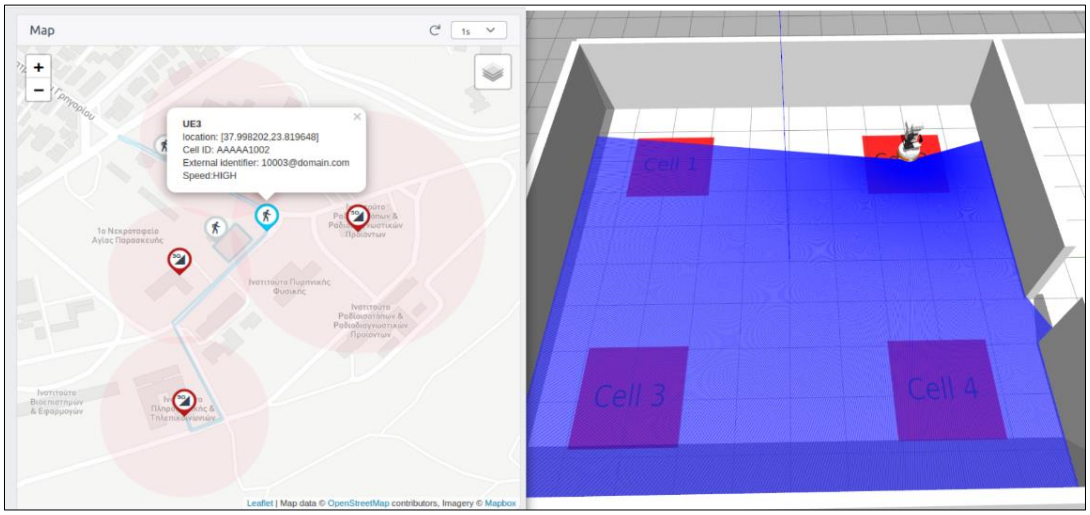


Figure 51 Simulated Tiago robot positioned at the Cell ID 2 location matching the UE 3 Cell ID connection

```
INFO:     127.0.0.1:58154 - "GET /cellid HTTP/1.1" 200 OK
[INFO] [1661496050.957639118] [cellid_node]: Publishing: CellID "2"
INFO:     127.0.0.1:58158 - "GET /cellid HTTP/1.1" 200 OK
[INFO] [1661496051.458483412] [cellid_node]: Publishing: CellID "2"
INFO:     127.0.0.1:58160 - "GET /cellid HTTP/1.1" 200 OK
[INFO] [1661496051.957778752] [cellid_node]: Publishing: CellID "2"
INFO:     127.0.0.1:58162 - "GET /cellid HTTP/1.1" 200 OK
[INFO] [1661496052.457625561] [cellid_node]: Publishing: CellID "2"
```

Figure 52 LocalizationNetApp publishing the current Cell_ID of UE 3

3.1.2.4.2 Teleoperation NetApp

The manual validation tests of the Teleoperation NetApp were performed locally, using all the necessary installations, as defined in the pre-conditions of the test case.

Table 46 Test report PAL NetApp General information

Test Case ID	*PAL-TeleopNetApp-02*
Test Case Description	*The scope of this test is to check the correct communication with the TiAGo real robot., the NetApp and the NEF Emulator. The TiAGo arm will be teleoperated through our vertical app with the geomagic touch haptic device. The expected result is that the real TiAGo robot stops any movement when the QoS is no longer guaranteed. And continue to move after the QoS is guaranteed again. The QoS is based on the QoS updates received from the NetApp through the NEF emulator.*
Created By	*Thomas Peyrucain*

Tester's Name	*Thomas Peyrucain*
Date Tested	*18-05-2022*
Test Case Results	***Pass***

Table 47 Test report PAL NetApp Pre conditions

S #	Prerequisites:
1	*NEF Emulator-NetApp are locally installed*
2	*End-to-end (network) communication between the vApp - NetApp - NEF Emulator*
3	*Create the predefined scenario in NEF Emulator with cells and UEs representing robots*
4	*Real Tiago robot is up to date with the Teleoperation NetApp installed*

In the figure below, a successful interaction between the UE 3 from the NEF Emulator and the robot is presented.

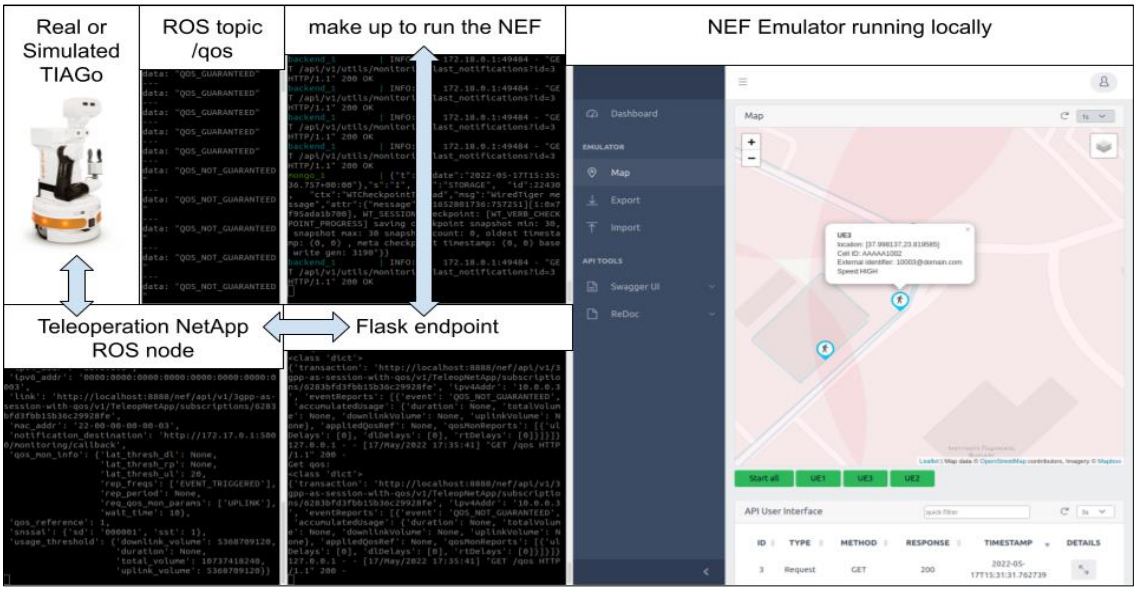


Figure 53 Initial state of the NEF Emulator with a valid connection with the robot

Table 48 Test report PAL NetApp-Test case sequence

Step #	Step Details	Expected Results	Actual Results	Result per step
1	*Start real Tiago*	*TIAGo robot should start and publish the right topics*	*As expected*	*Pass*
2	*Run NEF emulator*	*The emulator is available and spawning with the right ells and UEs*	*AS expected*	*Pass*

3	*Start "device 3" routine*	*UE 3 should move around the map*	*As expected*	*Pass*
4	*Run flask endpoint*	*flask endpoint with a valid connection to the emulator*	*As expected*	*Pass*
5	*Run TeleopNetApp*	*Creates a valid subscription to the flask endpoint and publishing a topic on /qos with the proper QoS*	*As expected*	*Pass*
6	*Run the Teleoperation*	*The robot should move the arm around to calibrate the ft sensor and switch to WBC, then when pressing the button on the haptic device and moving around the arm should move*	*As expected*	*Pass*
7	*Wait until the "device 3" changes coverage zone in the emulator and the QoS is no longer guaranteed*	*The arm should stop moving even if you are moving the pencil of the haptic device*	*As expected*	*Pass*
8	*Wait until the "device 3" changes coverage zone in the emulator and the QoS is guaranteed again*	*The teleoperation should work again as in step 6*	*As expected*	*Pass*

The successful execution of the tests can be seen in the following figures. When the QoS is guaranteed the robot motion match the movement of the haptic device.

Figure 54 TIAGo matching movement of the haptic device when QoS guaranteed

When the QoS is not guaranteed the robot stops and the motions and the haptic device motions are no more taken into account to prevent unwanted motions due to high latency:

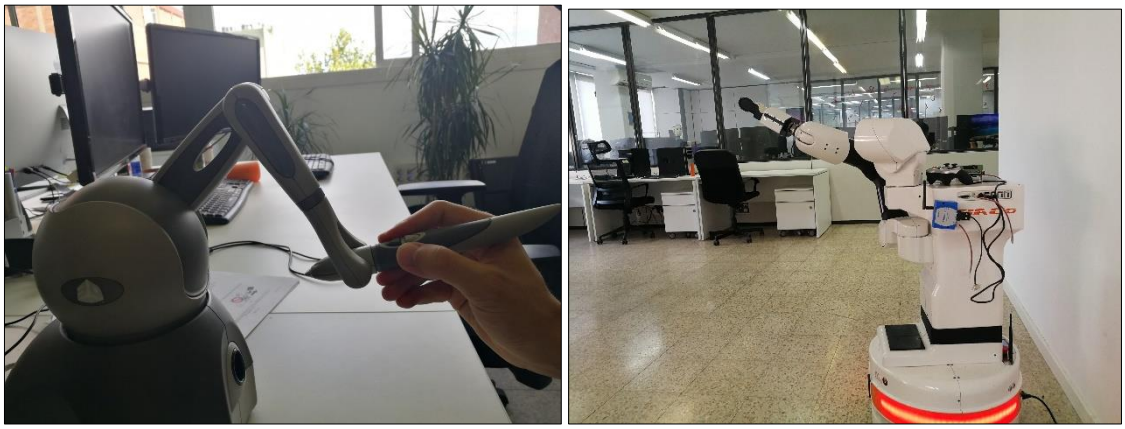


Figure 55 TIAGo not moving when QoS not guaranteed

4 AUTOMATED VALIDATION TESTS

4.1 OVERVIEW

The validation process includes the evaluation of the functionality of the NetApps when they are used along with the vertical App (vApp). The target is to prove that the vApps reach the desired performance for a set of KPIs defined by the vertical/SME while making use of the NetApps. Additionally, the validation of a NetApp with a vApp guarantees that the vApp makes use of all the capabilities exposed by the NetApp. This section describes the concept of the automated validation tests as the second step in the framework of the NetApps' validation process. The automated validation tests have been defined with scope to assess the NetApp's proper functionality as well as its correct interaction with other 5G network components such as CAPIF and NEF. The EVOLVED-5G validation environment is based on several tools created within the context of the 5GENESIS project, such as the Dispatcher and ELCM for the top-level component orchestration, while leveraging the use of general-purpose tools like OpenTAP, custom scripts and vendor specific management tools for controlling separate elements of the testbeds geared up to the automation of the validation process. Those tools are also described in detail in the following sections.

4.2 TOOLS FOR AUTOMATION

This subsection focuses on the characteristics and role of the of CI/CD automation server tools, as part of the automated validation tests (namely Jenkins, Robot Framework and the integration with the ELCM). Jenkins will be used to automate every process in EVOLVED-5G, following one-click automation best practices, RobotFramework will validate the proper performance of CAPIF and NEF, as well as their integration with NetApps and ELCM will coordinate experiments and manage the 5G Network infrastructure.

4.2.1 Jenkins

As described in Deliverable 3.2 [1], Jenkins is an industry-leading open-source automation server in charge of orchestrating a chain of actions related to building, deploying and testing software. In this section there is a focus on the role of Jenkins in the validation phase. In order to automate the validation process within the lifecycle of the NetApps, EVOLVED-5G within the overall architecture, has defined a validation pipeline (further description can be found in section 4.3.1, where all validation steps are thoroughly depicted).

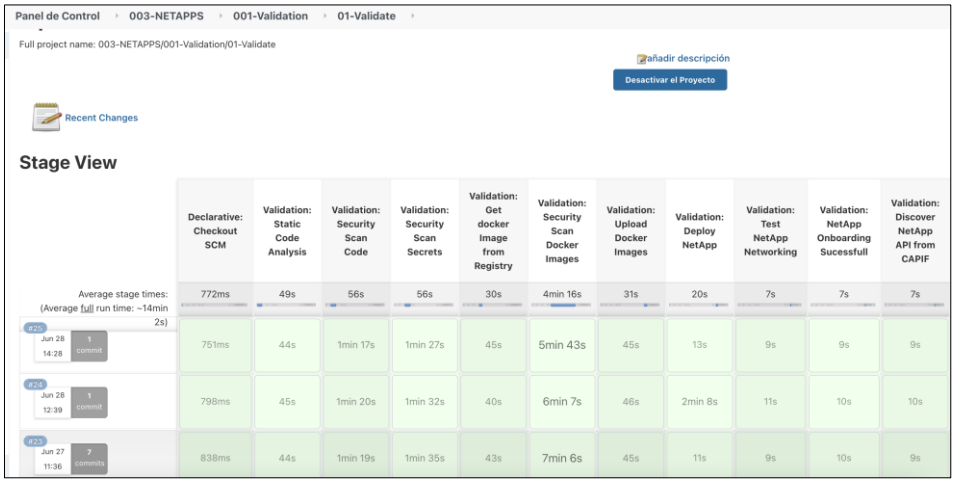


Figure 56 Jenkins pipeline flow 1

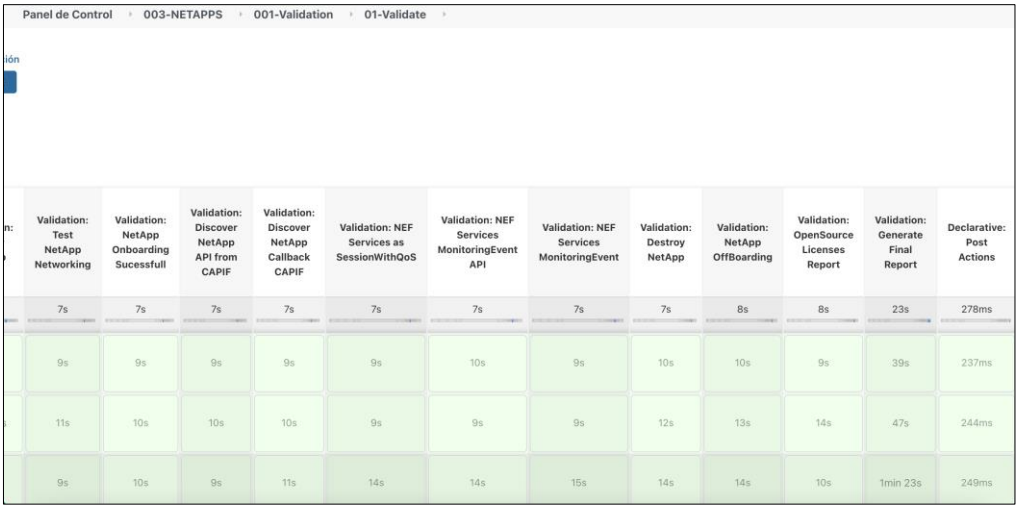


Figure 57 Jenkins pipeline flow 2

In Figure 56 and Figure 57 all the steps that are defined in the validation process (i.e., validation steps) are executed by a specific slice of the pipeline, called “stage”. In each of these steps, Jenkins will be the orchestrator for executing commands wrote in each of them. Figure 58 illustrates an example of the definition of a stage in a pipeline, where the execution of specific commands is automatically performed, aiming to generate a Sonarqube [17] report to a specific GitHub Repository.

```
stage('SonarQube Analysis and Wait for Quality Gate') {
    steps {
        dir ("${WORKSPACE}/") {
            withSonarQubeEnv('Evol5-SonarQube') {
                sh '''
                    ${SCANNERHOME}/bin/sonar-scanner -X \
                        -Dsonar.projectKey=Evolved5g-${NETAPP_NAME}-${GIT_NETAPP_BRANCH} \
                        -Dsonar.projectBaseDir="${WORKSPACE}/${NETAPP_NAME}/" \
                        -Dsonar.sources="${WORKSPACE}/${NETAPP_NAME}/src/" \
                        -Dsonar.host.url=http://195.235.92.134:9000 \
                        -Dsonar.login=$SQ_TOKEN \
                        -Dsonar.projectName=Evolved5g-${NETAPP_NAME}-${GIT_NETAPP_BRANCH} \
                        -Dsonar.language=python \
                        -Dsonar.sourceEncoding=UTF-8
                '''
            }
        }
    }
}

stage('Get SonarQube Report') {
    when {
        expression {
            return REPORTING;
        }
    }
    steps {
        dir ("${WORKSPACE}/") {
            //TODO: IMPROVE WAIT FOR REPORT READY
            //qualityGateStatus=true provokes an error
            sh '''
            sleep 15
            sonar-report \
                --sonarurl="http://195.235.92.134:9000" \
```

Figure 58 Jenkins’s pipeline definition.

4.2.2 Robot Framework

As described in Deliverable 3.2 [1], Robot Framework [18] is the tool used for automated testing purposes in EVOLVED-5G. In the validation process, RobotFramework will automatically test the deployment and configuration of the Common API Framework (CAPIF) and NEF Emulator. For this purpose, several test suites have been developed, following as a reference the 3GPP TS 23.222 [19], 3GPP TS 29.522 [11] and TS 29.122 [12] specifications.

4.2.3 ELCM

The ELCM is the component that is responsible for the coordination of the experiments during validation and the management of the resources in the 5G platform. The ELCM allows the implementation of Test Cases (which are used to define experiments and thus the Validation process) by defining the set of actions that are executed as part of the test.

Each action is defined by configuring a Task. Tasks include the functionality necessary for using a certain interface and are used for managing different pieces of software or hardware during an experiment execution. For example, the ELCM includes generic Tasks that allow the execution of a script by interfacing with the command line, or more specific Tasks that control the movement of a certain UE inside the NEF Emulator.

Once a Test Case for a Validation process has been defined, the execution of the experiment can be requested, via the Dispatcher. The ELCM then manages the execution of the Tasks described in the Test Case while managing the correct access to the necessary resources, which allows the concurrent execution of multiple experiments in the testbeds. However, for situations where exclusive access to all the resources is desirable it is possible to force certain experiments to run alone within the platform.

4.2.4 OpenTAP

OpenTAP is an open-source test sequencing engine mainly developed by Keysight Technologies [20]. It provides functionality for the management of heterogeneous equipment, the collection of results and the definition of the test logic, based on a set of test steps. In the context of EVOLVED-5G it is used as an additional execution engine that is managed by the ELCM using specific tasks.

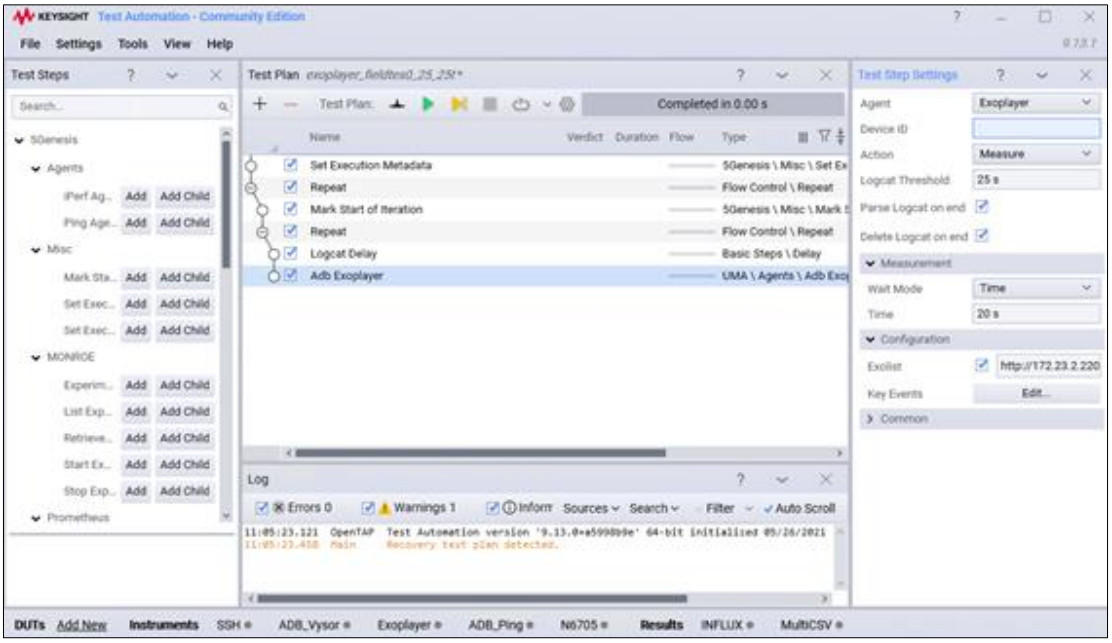


Figure 59 UI of OpenTAP

OpenTAP includes an initial set of test steps, instruments (that handle the communication with different software or hardware components) and result listeners (the entities that process the generated results, usually sending them to any possible storage solution outside of OpenTAP) that support some of the most common scenarios. However, it is highly extensible, allowing end-users to implement new functionality that can be seamlessly integrated to the UI.

4.3 VALIDATION PIPELINE

4.3.1 Validation Steps

The validation process mainly targets the testing of the NetApp in various conditions and against a variety of infrastructure, as well as the NetApp – vApp integration. For that reason, the validation process is a multifaceted procedure that involves tests relevant with (I) the successful deployment of the NetApp in a virtualized environment, (ii) the availability of proper infrastructure and exposure of the expected APIs, (iii) the effective communication of the NetApp with the 5G infrastructure and (iv) the performance of the NetApp. Next section presents the validation steps in detail.

NetApp Validation will consist of 19 steps, which are described below:

Step1 - Static Code Analysis: This step will obtain some quality code metrics in order to ensure good quality in the developed code. For this operation, Jenkins will use Debriked Compliance external tool [21].

Step 2 - Vulnerability scan of NetApp: This step will scan NetApp images trying to find all vulnerabilities. For this operation, Jenkins will use Trivy Compliance external tool [22].

Step 3 Secret leakage scan of NetApp: This step will check if there has been any secret leakage in the git history of the NetApp repository. Trivy will be used for this step as well.

Step 4 - Vulnerabilities scan of NetApp Images: This step will obtain some quality code metrics in order to ensure good quality in the developed code.

Step 5 - Image certification and push to Registry: In this step, Jenkins will sign images and documents in order to certificate the authority of those files. Jenkins will use OpenSSL [23] and Cosign tools [24].

Step 6 - Deploy of the NetApp in Kubernetes. Jenkins will deploy the NetApp in the selected infrastructure: OpenShift, Malaga or Athens and will continue with the execution if the deployment goes correctly.

Step 7 - Test open ports of the NetApp: Jenkins will check that declared ports in Dockerfile of the NetApp are open for connectivity purpose. Connectivity tools such as NMAP [25] or Telnet [26] will be used.

Step 8 – Test NEF and CAPIF APIs availability: The next step in the validation pipeline of Jenkins is to test the availability of the NEF and the CAPIF APIs. The tests are implemented using Robot Framework (please, refer to Section 4.2.2) and are following a testing plan defined for each one of these APIs. NEF is expected to be exposed by the NEF emulator and CAPIF by the CAPIF core function, both deployed either in the OpenShift infrastructure or in one of the Malaga or Athens infrastructures.

Step 9 - Onboarding NetApp in CAPIF Core Function (API Invoker) (CAPIF Event Report): With the NetApp Deployed, next step is to check that the NetApp registers in CAPIF Core Function properly as an API Invoker. In order to certify the registration, Jenkins will query CAPIF Core Function Database to double check that the NetApp have been registered properly by CAPIF Core Function.

Step 10 - Discover APIs using CAPIF Core Function (CAPIF Report): Once the NetApp has registered in CAPIF Core Function as API Invoker, next steps is for the NetApp to Discover APIs

published in CAPIF. To certify that NetApp is using Discover API from CAPIF Core Function, Jenkins will parse the logs from CAPIF Core Function Discover API to double check that the REST API have been consumed by the NetApp.

Step 11 - NetApp callback CAPIF Core Function (CAPIF Report): As part of the API dialogue between the NetApp and CAPIF Core Function, the NetApp will receive CAPIF Events. These events are notifications coming from CAPIF Core Function to the NetApp, and the CAPIF Event callback URL is set by the NetApp in the API Invoker Registration. Jenkins will certify that CAPIF Events are received by NetApp properly parsing logs from CAPIF Core Function EventsAPI. In the logs, "200 OK" responses from the NetApp must appear as the result of Events sent.

Step 12,13,14 - NEF Services API: The NetApp will discover NEF API using CAPIF Core Function Discovery API. Two APIs are exposed by NEF: AsSessionWithQoS and MonitoringEvent. As part of the API dialogue between the NetApp and NEF Services, the NetApp will receive NEF Events. Those events will come from the NEF emulator and the NetApp will have to receive them. Parsing logs will be necessary to verify this communication works appropriately.

Step 15 - NetApp – vertical App integration. Integration of the NetApp with its vertical app. If the virtual APP is always deployed on a public URL, this step can be automated as well.

Step 16 - Destroy NetApp: Continuing with the lifecycle of the NetApp, it is time to destroy the NetApp. Jenkins will clean all containers deployed and its configurations in the infrastructure.

Step 17 – Offboarding a NetApp: The NetApp must unregister from CAPIF Core Function as part of the NetApp removal clean up. Jenkins will do a CAPIF Core Function Database query to confirm that the NetApp has correctly unregistered.

Step 18-Open SourceScan Report: Validation pipeline will collect information about open-source Licenses used by the NetApp. For this operation, Jenkins will use Debriked Compliance external tool.

Step 19- Generate Validation Report Test Results: Each stage of the pipeline generates a partial report with information about the results obtained. The purpose of this stage is to aggregate all the information in a final report and send it to the developer.

Step #	NetApp Steps
1	Static code Analysis (SonarQube)
2	Vulnerability scan of NetApp (Trivy)
3	Secret leakage scan of Netapp (Trivy)
4	Container Image Certification Tool (Trivy)
5	Upload NetApp to Docker Registry (AWS Evolved5G Registry)
6	Deploy NetApp (Jenkins Pipeline) in K8 (Athens or Malaga)
7	Test open ports of the NetApps (declared in Dockerfile) (NMAP/Telnet)
8	Test NEF and CAPIF APIs' availability
9	Onboarding NetApp in CAPIF Core Function (API Invoker) (CAPIF Event Report)
10	Discover APIs using CAPIF Core Function (CAPIF Report)
11	NetApp callback CAPIF Core Function (CAPIF Report)
12	NEF Services AsSessionWithQoS API
13	NEF Services MonitoringEvent API
14	NEF Services Monitoring Events
15	**NetApp - vAPP integration (MANUAL)**
16	Destroy NetApp (Jenkins Pipeline)
17	Offboarding a NetApp (CAPIF) (CAPIF Event Report)
18	Open SourceScan Report (Debriked report)
19	Generate Validation Report Test Results

Figure 60 Stages of the validation pipeline in Jenkins

Finally, in Figure 61 NetApp Validation Process, we can see a detailed flow that all NetApps will have to go through in order to be validated.

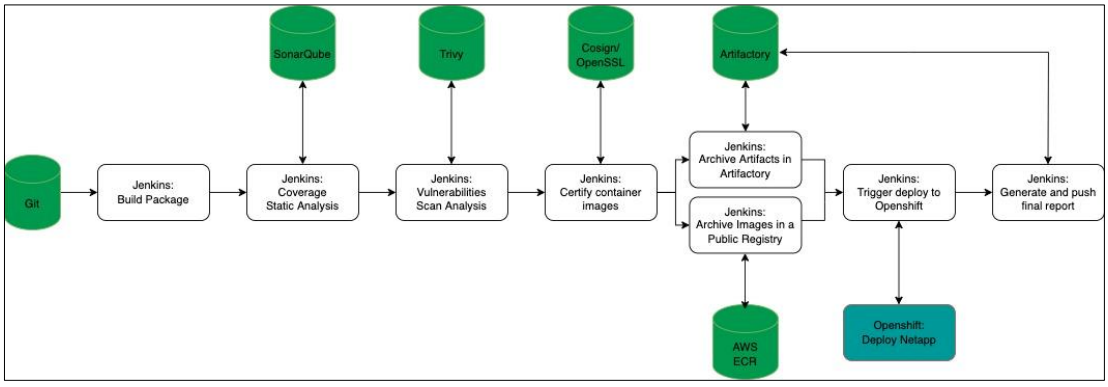


Figure 61 NetApp Validation Process

4.3.2 Open-repository onboarding

Upon completion of the process, images generated will be uploaded, along with the results of the validation to Artifactory Validation folder. Later, in order to continue with the required NetApp certification procedure, the Certification Process will retrieve the NetApp image from the Validation folder in Artifactory.

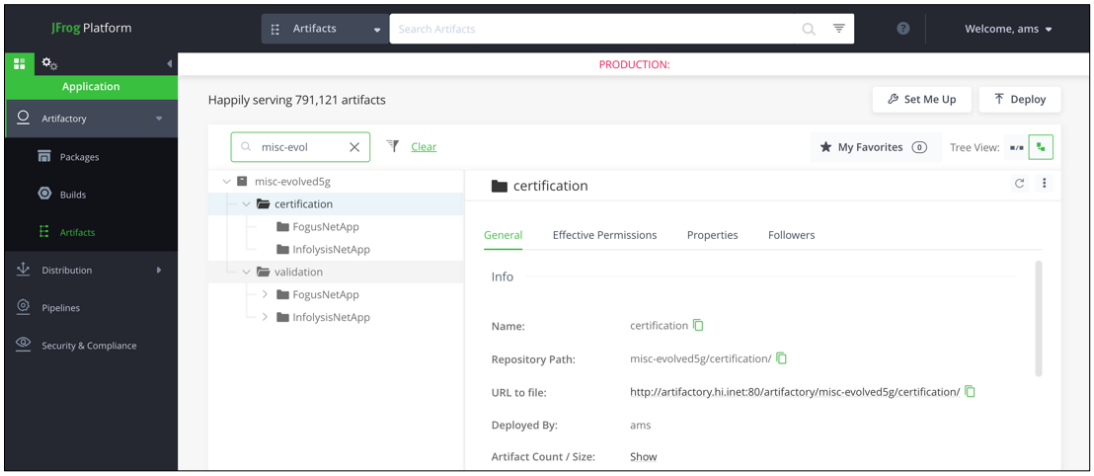


Figure 62 Validated image onboarded to Open-Repository

Only Validated NetApps will be suitable to being pushed to the Certification Phase. This is a condition established by EVOLVED-5G by design and based on the rationale that Certification resources will have some costs, and going through a successful validation of the NetApp guarantees that chances of NetApp being certified will be higher.

5 VULNERABILITY AND CODE ANALYSIS

This section not only describes main tools used for security assessment during validation but also summarizes reports that are obtained after using them.

The provision and selection of security tools respond to the need to address static code analysis and a scan of a possible range of relevant vulnerabilities during the NetApp validation phase. To this end, the following commercial and open-source tools have been selected whose main features and how they work have been thoroughly explained in the D3.2 [1]:

- Trivy [22]: for scanning vulnerabilities.
- Nmap [25], Telnet [26] or Netstat: for testing connectivity issues.we
- Sonarqube [17]: for code quality assurance.
- Debricked [21]: for code licensing analysis and compliance.

Each tool responds to a particular security need as detailed in the following sections below. All these tools have been integrated into the Jenkins Validation Pipelines steps previously described in section 4.3.

5.1 CONTAINER (NETAPP) VULNERABILITY SCAN (TRIVY)

Trivy is an open-source security tool that helps find vulnerabilities in containers before they become an issue in a production environment. Trivy has a very large database of vulnerabilities and many other sources that can provide information about any other potential vulnerabilities. As depicted in Figure 63 Trivy vulnerabilities report, different sections that form the report after the container vulnerability scan can be found.

Summary

Severity	Number of vulnerabilities
CRITICAL	37
HIGH	304
MEDIUM	266
LOW	537
UNKNOWN	47

Vulnerabilities

Severity	ID	Title	PkgName	InstalledVersion	FixedVersion
CRITICAL	CVE-2021-22945 (https://nvd.nist.gov/vuln/detail/CVE-2021-22945)	curl: use-after-free and double-free in MQTT sending	curl	7.74.0-1.3+deb11u1	
CRITICAL	CVE-2021-30473 (https://nvd.nist.gov/vuln/detail/CVE-2021-30473)	aom_image.c in libaom in AOMedia before 2021-04-07 frees memory that i ...	libaom0	1.0.0.errata1-3	

Figure 63 Trivy vulnerabilities report

Indeed, Trivy displays all vulnerabilities discovered in the container images. Developers can see how this report displays a detailed description of the problem, including the name of the affected library and the version that fixes the problem (if any).

5.2 OPEN PORTS TESTING (NMAP, TELNET AND NETSTAT)

Nmap, telnet or netstat are well-known tools in the industry that in the framework of EVOLVED-5G are used for testing connectivity issues. In the validation pipeline these tools will be used to verify that after the deployment of the NetApp, all ports are correctly opened.

```
Active Internet connections
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  192.168.8.185.58257    13.107.136.9.https     ESTABLISHED
tcp4       0      0  192.168.8.185.58256    13.107.136.9.https     ESTABLISHED
tcp4       0      0  192.168.8.185.58255    52.109.76.225.https    ESTABLISHED
tcp4       0      0  192.168.8.185.58250    64.62.250.111.https    ESTABLISHED
tcp4       0      0  192.168.8.185.58249    64.62.250.111.https    ESTABLISHED
tcp4       0      0  localhost.http-alt     localhost.58248        ESTABLISHED
```

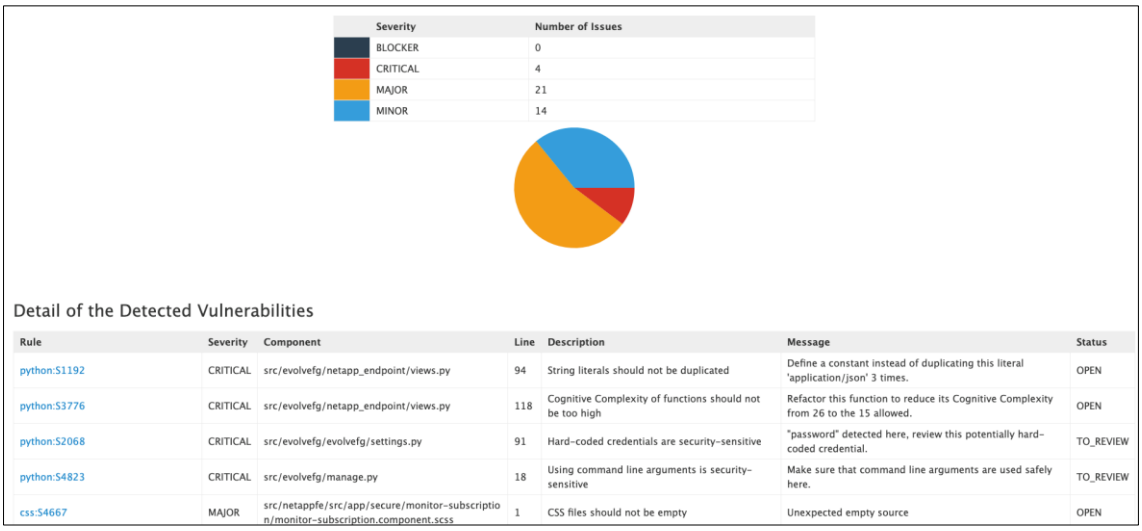
Figure 64 Netstat output

As depicted in Figure 64, the output of one of these tools (in this case, netstat) and the information that it provides about which IP addresses and ports are open. This will help to ensure that connectivity can be established on the specified ports.

5.3 Syntax Analysis For Source Code (Sonarqube)

Sonarqube is an open-source quality assurance tool used for measuring the overall quality of the code. Among the metrics that SonarQube uses are the following:

- Complexity: based on the cyclomatic complexity (number of flow paths through the code)
- Duplications: number of duplicated blocks of lines.
- Issues: number of issues raised for the first time on New Code.
- Maintainability: total amount of code smells issues.
- Reliability: number of bugs issues in the code.
- Security: number of vulnerabilities issues.
- Size: Numbers of classes, lines, numbers of comments, etc….



Severity	Number of Issues
BLOCKER	0
CRITICAL	4
MAJOR	21
MINOR	14

Detail of the Detected Vulnerabilities

Rule	Severity	Component	Line	Description	Message	Status
python:S1192	CRITICAL	src/evolvefg/netapp_endpoint/views.py	94	String literals should not be duplicated	Define a constant instead of duplicating this literal 'application/json' 3 times.	OPEN
python:S3776	CRITICAL	src/evolvefg/netapp_endpoint/views.py	118	Cognitive Complexity of functions should not be too high	Refactor this function to reduce its Cognitive Complexity from 26 to the 15 allowed.	OPEN
python:S2068	CRITICAL	src/evolvefg/evolvefg/settings.py	91	Hard-coded credentials are security-sensitive	"password" detected here, review this potentially hard-coded credential.	TO_REVIEW
python:S4823	CRITICAL	src/evolvefg/manage.py	18	Using command line arguments is security-sensitive	Make sure that command line arguments are used safely here.	TO_REVIEW
css:S4667	MAJOR	src/netappfe/src/app/secure/monitor-subscription/monitor-subscription.component.scss	1	CSS files should not be empty	Unexpected empty source	OPEN

Figure 65 SonarQube report example

Sonarqube report includes a summary with a pie chart, as well as detailed information about each error. This report provides a description of the problem, as well as the affected line and its severity.

5.4 Open-Source License Scanning (Debricked)

Debricked tool allows generating a report containing all the information about the license of the code. This tool allows developers to spend less time on security and legal aspects, and more

time on writing good quality code. Debricked can help in identifying non-compliant packages licenses and suggesting solution to solve those problems.

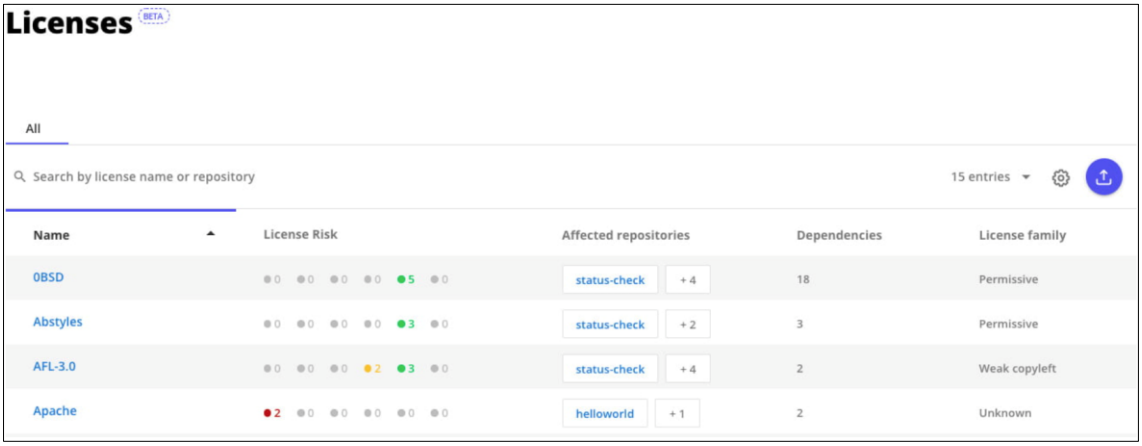


Figure 66 Debricked licenses report

Debricked report provides a view of all potential license risks in the code, as well as which parts of the code are affected. In Figure 66, it is clear that the report analysis shows that the repository contains two vulnerabilities that must be addressed in order to prevent any non-compliant use of the Apache license.

6 CONCLUSION

This deliverable presents in detail the work performed in the context of WP5 and more specifically in Task 5.2 which is driving the current deliverable. One of the primary purposes of the specific task, during the lifetime of the project, is to define and provide validation tests to all the NetApps by tightly interacting with WP3 and WP4. Towards this end, from the former specific tools supporting the development process are being utilised while the latter provides inputs to WP5 through the developed NetApps that are to be validated.

Task 5.2 focuses on the NetApps Validation and Onboarding to the Open Repository and to that end section 2 presents the validation framework that is being utilised for the implementation of the tests that the NetApps should undergo. This framework includes the Continuous Integration / Continuous Development (CI/CD) tools, the Kubernetes clusters and 5G Network functions that will be installed in the EVOLVED-5G platforms (Athens-Malaga). Section 3 aggregates the manual validation test cases of the NetApps that have been performed as the initial step towards the final validation process, as well as the test case reports for each test case and each NetApp. Sections 4 describes the role of the of CI/CD automation server tools and the steps comprising the validation process of the NetApps. Finally in section 5 a synopsis of the security tools integrated in some of the validation steps during the validation phase is presented.

7 REFERENCES

[1] EVOLVED-5G, "Deliverable D3.2 “NetApp Certification Tools and Marketplace development(intermediate) https://evolved-5g.eu/wp-content/uploads/2022/07/EVOLVED-5G-D3.2_FINAL.pdf".

[2] Terraform. [Online]. Available: https://www.terraform.io/docs.

[3] A. ECR. [Online]. Available: https://aws.amazon.com/es/ecr/.

[4] J. Artifactory. [Online]. Available: https://www.jfrog.com/confluence/display/JFROG/JFrog+Documentation.

[5] Openshift. [Online]. Available: https://docs.openshift.com/.

[6] [Online]. Available: https://projectcalico.docs.tigera.io/reference/installation/api.

[7] [Online]. Available: https://kubernetes.io/docs/tasks/tools/.

[8] [Online]. Available: https://containerd.io/.

[9] [Online]. Available: https://metallb.universe.tf/.

[10] [Online]. Available: https://kubevirt.io/.

[11] 3GPP, "TS 29.522 "5G System; Network Exposure Function Northbound APIs; Stage 3", Release 15, v17.6.0," https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3437, June 2022.

[12] 3GPP, "TS 29.122 "T8 reference point for Northbound APIs", Release 15, v17.6.0, June 2022," [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3239.

[13] EVOLVED-5G, "D3.1 Implementations and integrations towards EVOLVED-5G framework," [Online]. Available: https://evolved-5g.eu/wp-content/uploads/2022/01/EVOLVED-5G-D3.1-v1.0.pdf.

[14] EVOLVED-5G, "Deliverable D4.1 “5G Exposure Capabilities for Vertical Applications," [Online]. Available: https://evolved-5g.eu/wp-content/uploads/2022/03/EVOLVED-5G-D4.1_v2.0-final.pdf.

[15] Almudena Díaz Zayas, Giuseppe Caso, Özgü Alay, Pedro Merino, Anna Brunstrom, Dimitris Tsolkas, Harilaos Koumara, "A Modular Experimentation Methodology for 5G Deployments: The 5GENESIS Approach," *Sensors International Journal,* vol. 20, no. 22, p. 6652; https://doi.org/10.3390/s20226652, 2020.

[16] [Online]. Available: https://pal-robotics.com/robots/tiago/ .

[17] [Online]. Available: https://www.sonarqube.org/.

[18] [Online]. Available: https://robotframework.org/.

[19] 3GPP, "TS 23.222 Common API Framework for 3GPP Northbound APIs," [Online].

[20] [Online]. Available: https://opentap.io/.

[21] [Online]. Available: https://debricked.com/tools/license-compliance.

[22] [Online]. Available: https://aquasecurity.github.io/trivy/v0.29.2/.

[23] [Online]. Available: https://www.openssl.org/.

[24] [Online]. Available: https://docs.sigstore.dev/cosign/overview/.

[25] [Online]. Available: https://nmap.org/docs.html.

[26] [Online]. Available: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/telnet.

8 ANNEXES

8.1 MANUAL TEST CASE TEMPLATE

EVOLVED-5G Test Case Template	-ID number-	-Title-	- Target Metric (KPI family) -
Scenario (storyline)	*Description of the motivation and the scope of the test in a qualitative level. What is the reference scenario in a real (industrial) environment that we want to capture with this test?*		
Testing Infrastructure (Pre-conditions)	*Information related to all the parameters that affect the values of the KPIs to be measured, network deployment and environment conditions, etc. [Any requirement that needs to be done before execution of this test case. A list of test specific pre-conditions that need to be met by the SUT including information about equipment configuration, traffic descriptor]* • *The set of software and hardware components involved + their configuration* • *The service type + the traffic that is involved in the process*		
Target KPI	*Here goes the definition of the target KPI. Each test case targets only one KPI (main KPI). However, secondary measurements from complementary KPIs can be added as well. The definition of the main KPI specializes the related target metric (the ID of the related target metric is declared in the first row of this template). More precisely, the definition of the main KPI declares:* • *The definition of the KPI + (if applicable) a secondary list of KPIs useful to interpret the values of the target KPI.* • *The reference points from which the measurement(s) will be performed* • *The reference protocol stack level where the measurement is performed* • *Target values + theoretical value space*		

Test Case Sequence	Specializes the measurement process (methodology) of the metric for the selected underlay system. In this field: • The steps to be followed for performing the measurements are specified • The iterations required, the monitoring frequency, etc., are declared.

8.2 QUALITY ASSURANCE-TEST CASE REPORT

General information

Test Case ID	*<SME-Netapp-XX>*
Test Case Description	*<Description of the Target KPI (from test case template)>*
Created By	*<Creator's Name>*
Tester's Name	*<Tester's Name>*
Date Tested	*<DD-MM-YYYY>*
Test Case Results	*<Pass/Fail/Not Executed >*

Pre-Conditions

S #	Prerequisites:
1	*<the pre-condition defined in test case's template>*
2	
3	
…	

N	

Test Case Sequence

Step #	Step Details	Expected Results	Actual Results	Result per step
1	*<Describe the details of this step>*	*<Describe the expected results on this step>*	*<As Expected / Report the unexpected results on this step>*	*< Pass / Fail / Not executed / Suspended>*
2				
3				
…				
N				