EXPERIMENTATION AND VALIDATION OPENNESS FOR LONGTERM
EVOLUTION OF VERTICAL INDUSTRIES IN 5G ERA AND BEYOND
[H2020 - Grant Agreement No.101016608]

Deliverable D3.4

Network Apps Certification Tools and Marketplace development (final)

Editor David Artuñedo (TID)

Contributors (ATOS, MAG, INTRA, NCSRD, FOGUS)

Version 1.0

Date June 30th, 2023

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 not 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	*March 16th, 2023*	*David Artuñedo (TID)*	*Edit ToC*
0.6	*June 5th, 2023*	*David Artuñedo (TID)*	*First Draft*
0.7	*June 14th, 2023*	*Ricardo Marco (ATOS)* *Yannis Karadimas (MAG)*	*Review of the document*
0.8	*June 21st 2023*	*David Artuñedo (TID)* *Inés de Ibargüen (TID)*	*Quality review and final edition*
0.9	*June 26th 2023*	*Javier Garcia (TID)* *Harilaos Koumaras (NCSRD)* *Dimitris Tsolkas (FOG)*	*Final review*
1.0	*June 30th 2023*	*David Artuñedo (TID)*	*Final version*

LIST OF AUTHORS

Partner ACRONYM	*Partner FULL NAME*	*Name & Surname*
TID	*TELEFONICA INVESTIGACIÓN Y DESARROLLO*	*Javier Garcia* *David Artuñedo* *Jorge Moratinos*
MAG	*MAGGIOLI*	*Yannis Karadimas*
INTRA	*INTRASOFT INTERNATIONAL SA*	*Angela Dimitriou*
ATOS	*ATOS IT SOLUTIONS ANDSERVICES IBERIA SL*	*Ricardo Marco* *Sonia Castro*
FOGUS	*FOGUS INNOVATIONS & SERVICES P.C*	*Dimitris Tsolkas* *Anastasios-Stavros Charismiadis* *Katerina Giannopoulou*
NCSRD	*NATIONAL CENTER FOR SCIENTIFIC RESEARCH"DEMOKRITOS"*	*George Makropoulos* *Harilaos Koumaras* *Dimitris Fragkos* *Anastasios Gogos* *Dimitris Kyriazanos* *Thanos Papakyriakou* *Eleni Charou* *John Manolopoulos*

GLOSSARY

Abbreviations/Acronym	*Description*
3GPP	*3rd Generation Partnership Project*
5GC	*5G Core*
AI	*Artificial Intelligence*
API	*Application Programming Interface*
CAPIF	*Common API Framework*
CI/CD	*Continuous Integration / Continuous Development*
CLI	*Command Line Interface*
COTS	*Commercial off-the-shelf*
CPE	*Customer Premises Equipment*
EC	European Commission
ELCM	*Experiment Life-Cycle Manager*
FoF	Factories of the Future
gNodeB / gNB	*Next Generation (5G) Base Station*
GUI	*Graphical User Interface*
HAT	*Hardware Attached on Top*
HTTP	*Hypertext Transfer Protocol*
IIoT	*Industrial Internet of Things*
JSON	*JavaScript Object Notation*
JWT	*JSON Web Token*
KPI	*Key Performance Indicator*
LDAP	*Lightweight Directory Access Protocol*
LTE	*Long Term Evolution*
LTE-A	*Long Term Evolution Advanced*
LTE-M	*Long Term Evolution for Machines*
MAC	*Mandatory Access Control*
MANO	*Management and Orchestration*
NB-IoT	*Narrow Band – Internet of Things*
NEF	*Network Exposure Function*
Network App	*Network Application*
NFV	*Network Function Virtualization*
NSA	*Non StandAlone*
NSD	*Network Service Descriptor*
OAuth	*Open Authorization*
OEM	*Original Equipment Manufacturer*
ONAP	*Open Network Automation Platform*
OSM	*Open Source MANO*
PoP	*Platform to Platform*
PR	*Pull request*
QoS	*Quality of Service*
R&D	*Research and Development*

REST	*Representational State Transfer*
SA	*StandAlone*
SDK	*Software Development Kit*
SLA	*Service Level Agreement*
SME	*Small Medium Companies*
SSH	*Secure Shell*
SSL	*Secure Sockets Layer*
SSM	*Service Specific Manager*
TSL	*Transport Layer Security*
UE	*User Equipment*
UI	*User Interface*
vAPP	*Vertical Application*
VDU	*Virtual Display Unit*
VNF	*Virtual Network Function*
VulnDB	*Vulnerability Database*
WiFi	*Wireless Fidelity*

EXECUTIVE SUMMARY

The primary objective of EVOLVED-5G is to provide enhanced experimentation facilities on top of which third party experimenters (such as SMEs or any service provider and target vertical users) will have the opportunity to test their applications. EVOLVED-5G responds to the 5G PPP ICT-41-2020 5G innovations for verticals with third party services call. The EVOLVED-5G project works to make this vision a reality by promoting the development of a Network App ecosystem centered around a experimental 5G facility. This facility will offer the tools and procedures needed for the development, verification, validation, and certification of Network Apps as well as for their smooth operation on top of actual 5G network infrastructures and mechanisms for market release.

The main goal of the present deliverable is to outline the final release of those tools, initiatives, and methodologies that are implemented for the realization of the **EVOLVED-5G Certification Process and Environment**, with a focus on the architectural components' implementation, effectiveness, and security. These architectural components define the **Network App Certification Process and the Marketplace**, which are used to distribute the certified Network Apps.

This document presents the outcome of task *"T3.4: Network App Certification Tools and Marketplace development"* providing a final description of the selected tools for the design of the automated Certification Process, the Test engine for running certification tests, and the description of tests that are part of the Certification process.

The deliverable represents an upgrade on D3.2 "*NetApp Certification Tools and Marketplace development"* in the following sense:

- The CI/CD services supporting the implementation of the certification process has been revisited, including final versions of:
  - NEF, including token integration with CAPIF.
  - CAPIF, including new methods that are fundamental for certifying API usage during certification testing
  - TSN, for certifying the usage of deterministic communication in the Network Application
  - New tools, such as Helm for deployment automation
  - Removal tools as Terraform, no longer necessary as K8s environments are static there is no need for K8s clusters re-deployment.
- Openshift, as container execution platform, has been upgraded to version 4.10.
- A new Kubernetes cluster has been created for EVOLVED-5G project in COSMOTE premises as part of the Athens certification environment with the purpose of hosting various applications.
- The conceptualization of the certification pipeline definition is now complete, ready for implementation in WP5
- Marketplace:
  - The two-step design process for the Marketplace is finalised as well as the final high-level architecture.

- All components that were envisioned or intermediately released and reported in D3.2, are now completed and finalised, achieving the full implementation of Marketplace. It is worth highlighting the full implementation of the blockchain integration module, the full integration of the marketplace with the open repository (antifactory) as well as the full integration of the marketplace with OpenShift, and marketplace-ready upgrade for network app versioning.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Scope

The scope of this deliverable is to describe the final implementation and details for both the EVOLVED-5G Certification Environment and the Marketplace. The intention is to make the document accessible to a broad variety of research individuals and communities.

The target audiences are the same as identified in D3.2 *"NetApp Certification Tools and Marketplace development"* but it goes one step further as it expands potential communities by targeting Global Certification Authorities.

- **Project Consortium**: To validate that all objectives and proposed technological advancements have been analyzed and to ensure that, through the proposed implementations, further work can be derived. Furthermore, the deliverable sets a common understanding among the consortium with regards to:
    - The implementation details of the Certification Environment related to the Network Apps lifecycle in the context of the EVOLVED-5G project, including tools and technologies to be used.
    - The implementation details of the Marketplace, the final step of the Network Apps lifecycle in the context of the EVOLVED-5G project.
- **Industry 4.0/Industry 4.0 developers and Factories of the Future (FoF) vertical groups**: To set a common understanding of the technologies and the design principles that underline the Certification Process design and its implementation, along with the Marketplace release process.
- **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 being designed to secure interoperability beyond vendor-specific implementation and across multiple domains. The same categorization can be applicable beyond the Industry 4.0 domain.
- **Global Certification authorities**: Independent certification authorities and partnerships (like the Global Certification Forum) aiming at creating certification programs to help ensure global interoperability in future networks.
- **The scientific audience, general public, and the funding EC Organization**: To document the work performed by the project and justify the effort reported for all relevant activities. The scientific audience can also get an insight of the design approach and underlying components behind the EVOLVED-5G Certification Process and Marketplace implementation.

1.2 Objectives

This deliverable is the final deliverable of WP3 and describes the outcome of task *"T3.4- Network App Certification Tools and Marketplace development"*. It provides the final upgrade on D3.2 *"NetApp Certification Tools and Marketplace development"* and outlines the final implementation details of the EVOLVED-5G Certification Process and

Environment. This includes a focus on implementation efficiency, and security aspects of its architectural components that define Network App certification Process and the final release of the so-called certified Network Apps to the Marketplace.

This document gives a (i) final description of the selected tools for the design of the automated Certification process, (ii) the Test engine for running certification tests, and (iii) the description of tests that will be part of the **Certification process**. Commercial on the shelf tools, such as SonarQube, Trivy or Debricked have been selected to cover the Certification requirements listed in D2.1 [1], as well as additional implementations developed in EVOLVED-5G such as the Network Exposure Function (NEF) Emulator, Time Sensitive Networking (TSN) or Common API Framework (CAPIF) have been used to facilitate the Certification process.

Finally, a complete **EVOLVED-5G Marketplace** has been developed and deployed to enable certified Network Apps publication and release.

1.3 Structure

This deliverable is built upon D3.2 "*"NetApp Certification Tools and Marketplace development"* hence it follows the same chapter distribution:

- Section 1. Introduction: This section describes the Deliverable target audience, objectives and structure.
- Section 2. Certification Environment: This section describes the Certification Environment focusing on the CI/CD toolset with a collection of industry leading software tools for automation.
- Section 3. Certification Tools: This section describes the specific tools that have been developed by EVOLVED-5G project for Network App Certification, namely CAPIF Core Function, NEF Emulator and TSN Application Function, and commercial of the shelf tools selected to cover certification requirements, such as SonarQube, Trivy and Debricked.
- Section 4. Certification Process: This section provides an overview of the Certification process, including the multiple steps that are being implemented in the Certification Pipeline to automate the Certification Process.
- Section 5. Marketplace: This section describes the EVOLVED-5G Marketplace, developed specifically for this project. This platform is used to publish Network Apps upon the completion of the Certification process. It also contains the details of the Marketplace and Open Repository Integration (that was a separate section in D3.2).

2 Certification Environment

The tools selected for building the certification environment were initially described in D3.2 [2]. In this deliverable we provide an update on the selected tools and the final details on how the tools have been integrated into the EVOLVED-5G Certification Environment.

This environment facilitates the execution of the Certification Process and, also, the execution of most of the Steps that define this process, as described in section 4. The environment follows industry's best practices for CI/CD methodologies, and the selected tools are widely regarded as leading references within open-source communities of the industry

This section revisits CI/CD TID´s [3] solution and principles, it highlights the updates on main tools that articulate the CI/CD system, namely the Open Repository to store artifacts, Jenkins for Automation, Helm to deploy Containers, Robot Framework to build automated test and finally, it describes the Certification execution platforms where Network Applications will be deployed, and tests will be executed.

2.1 CI/CD Support for Certification process

An initial description of the CI/CD method was added to D3.2, in section 2.1. From this initial overview, the following upgrades have been performed:

- The management of the infrastructure as code no longer involves the use of the Terraform tool. Indeed, as K8s environments are static there is no need for K8s clusters re-deployment.
- Helm tool has been added to automate the Network Application deployments, and auxiliary tools deployment such as NEF Emulator, CAPIF and TSN.

The final composition of the CI/CD environment is displayed in Figure 1.

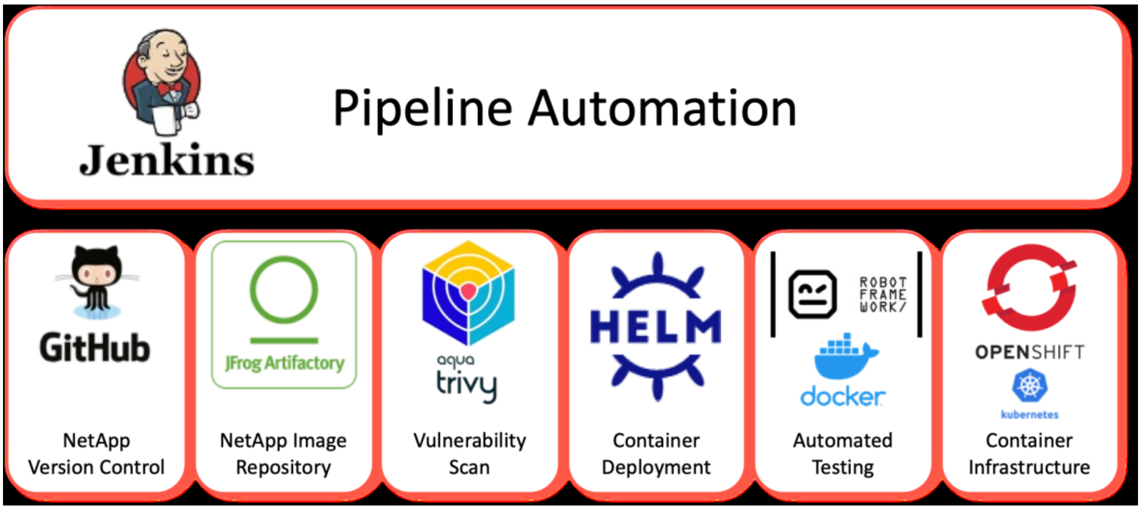


Figure 1: CI/CD Toolset used in EVOLVED-5G Certification Environment

2.2 Open Repository (GitHub)

EVOLVED-5G uses GitHub [4] as control version tool, along with git well-known best practices. EVOLVED-5G repository shows the following numbers:

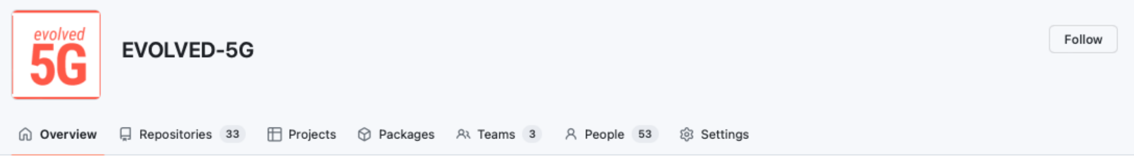
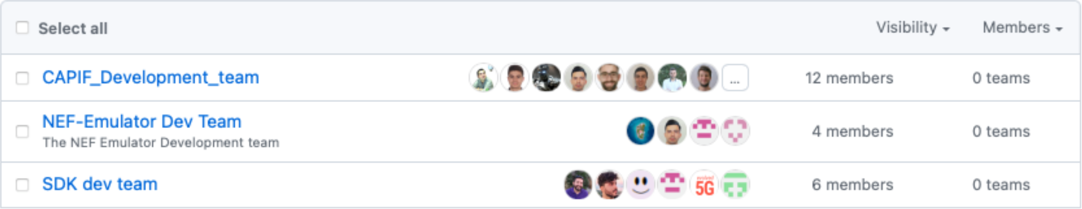


Figure 2: EVOLVED-5G Repository in GitHub

The project has 33 repositories registered, mostly stemming from the Network Applications, but also from auxiliary tools such as NEF Emulator, CAPIF, TSN, Marketplace.

GitHub has also fostered collaboration between EVOLVED-5G partners, supporting 3 teams working in some of the tools of the project, as depicted in Figure 3.

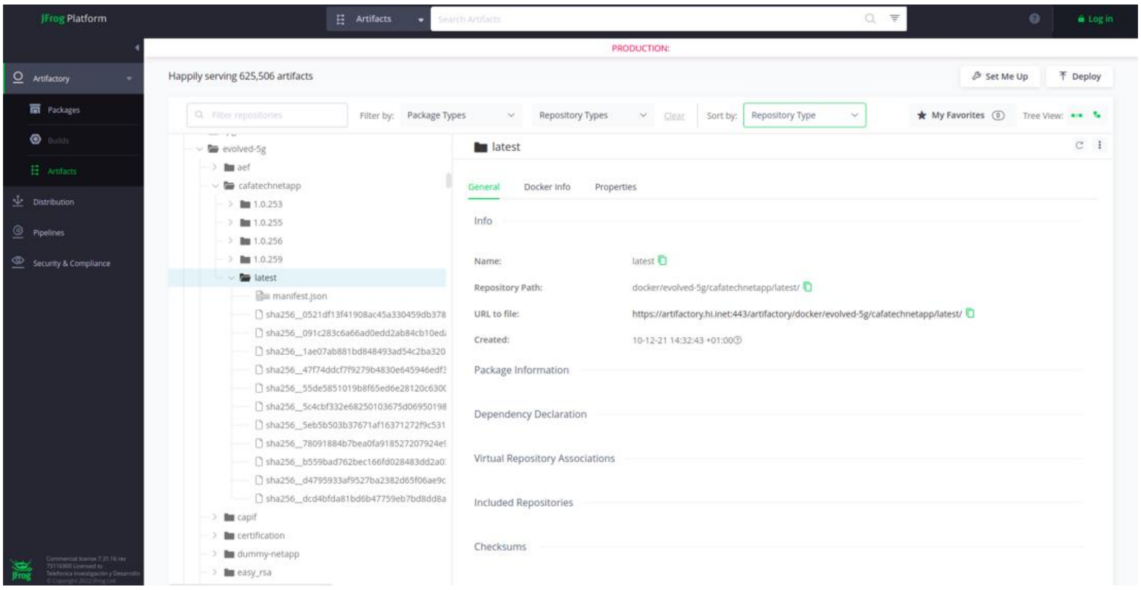


□ Select all	Visibility ▾	Members ▾
□ CAPIF_Development_team	12 members	0 teams
□ NEF-Emulator Dev Team The NEF Emulator Development team	4 members	0 teams
□ SDK dev team	6 members	0 teams

Figure 3: Teams working collaboratively in EVOLVED-5G GitHub repository

2.2.1 Image Management

As described in Deliverable 3.2 (section 2.2), Docker is used to build the images of the Network Applications, and JFrog Artifactory [5] to store the images.



Happily serving 625,506 artifacts

- evolved-5g
  - aef
  - cafatechnetapp
    - 1.0.253
    - 1.0.255
    - 1.0.256
    - 1.0.259
    - latest
      - manifest.json
      - sha256__0521df13f41908ac45a330459db378
      - sha256__091c283c6a66ad0edd2ab84cb10ed
      - sha256__1ae07ab881bd848493ad54c2ba320
      - sha256__47f74ddcf7f9279b4830e645946edf
      - sha256__55de5851019b8f65ed6e28120c630
      - sha256__5c4cbf332e68250103675d06950198
      - sha256__5eb5b503b37671af16371272f9c531
      - sha256__78091884b7bea0fa918527207924e
      - sha256__b559bad762bec166fd028483dd2a0
      - sha256__d4795933af9527ba2382d65f06ae9c
      - sha256__dcd4bfda81bd6b47759eb7bd8dd8a
  - capif
  - certification
  - dummy-netapp
  - easy_rsa

latest

Info

Name: latest

Repository Path: docker/evolved-5g/cafatechnetapp/latest/

URL to file: https://artifactory.hi.inet:443/artifactory/docker/evolved-5g/cafatechnetapp/latest/

Created: 10-12-21 14:32:43 +01:00

Package Information

Dependency Declaration

Virtual Repository Associations

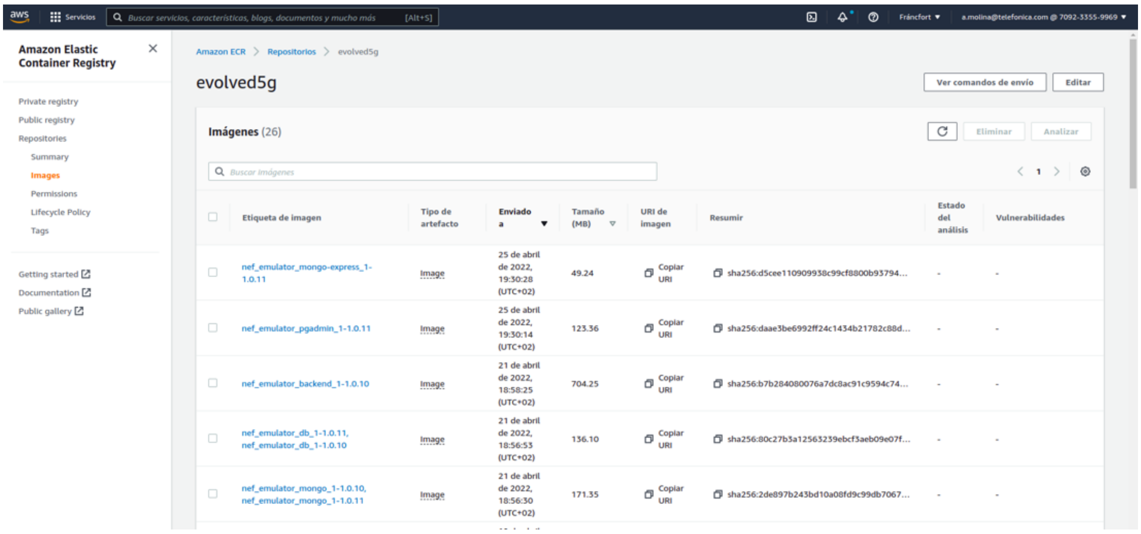
Included Repositories

Checksums

Commercial license 7.31.16 rev 73116900 Licensed to Telefonica Investigación y Desarrollo
© Copyright 2022 JFrog Ltd

Figure 4: JFrog Artifactory screenshot

Additionally, Amazon Web Services (AWS) Elastic Container Registry [6] is an AWS managed container image registry service that enables images to be pushed and pulled (i.e Docker, OCI [7]) from anywhere as it is available in the Public Cloud.



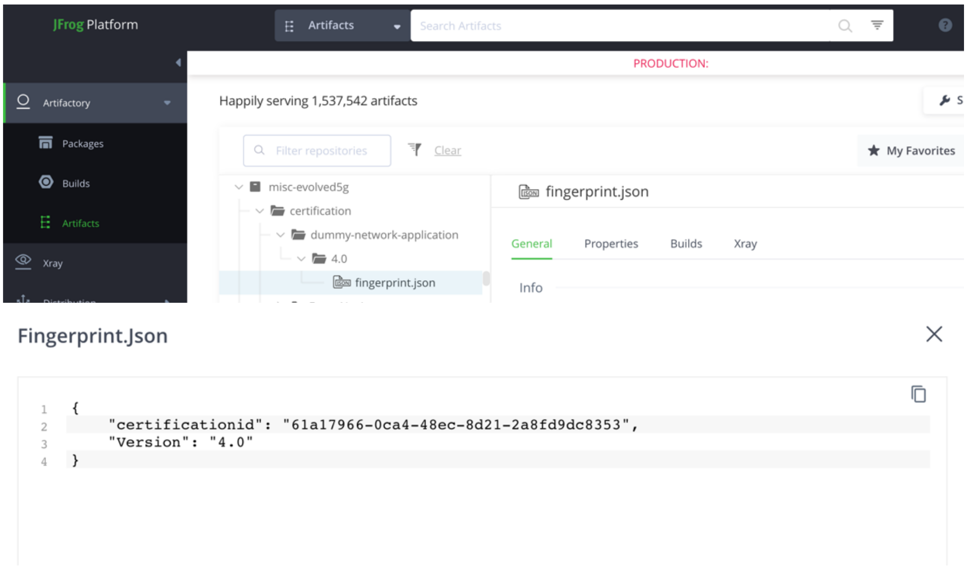
evolved5g

Imágenes (26)

□	Etiqueta de imagen	Tipo de artefacto	Enviado a	Tamaño (MB)	URI de imagen	Resumir	Estado del análisis	Vulnerabilidades
□	nef_emulator_mongo-express_1-1.0.11	Image	25 de abril de 2022, 19:30:28 (UTC+02)	49.24	Copiar URI	sha256:d5cee110909938c99cf8800b93794...	-	-
□	nef_emulator_pgadmin_1-1.0.11	Image	25 de abril de 2022, 19:30:14 (UTC+02)	123.36	Copiar URI	sha256:daae3be6992ff24c1434b21782c88d...	-	-
□	nef_emulator_backend_1-1.0.10	Image	21 de abril de 2022, 18:58:25 (UTC+02)	704.25	Copiar URI	sha256:b7b284080076a7dc8ac91c9594c74...	-	-
□	nef_emulator_db_1-1.0.11, nef_emulator_db_1-1.0.10	Image	21 de abril de 2022, 18:56:53 (UTC+02)	136.10	Copiar URI	sha256:80c27b3a12563239ebcf3aeb09e07f...	-	-
□	nef_emulator_mongo_1-1.0.10, nef_emulator_mongo_1-1.0.11	Image	21 de abril de 2022, 18:56:30 (UTC+02)	171.35	Copiar URI	sha256:2de897b243bd10a08fd9c99db7067...	-	-

Figure 5: AWS Elastic Container Registry screenshot

Artifactory holds Network Applications' images both for Validation and Certification purposes. It also stores the fingerprint json file generated during Certification process that is required to upload the Network Applications to the Marketplace, as explained in section 5.4.



Happily serving 1,537,542 artifacts

fingerprint.json

Fingerprint.Json

```
{
    "certificationid": "61a17966-0ca4-48ec-8d21-2a8fd9dc8353",
    "Version": "4.0"
}
```

Figure 6: Fingerprint JSON File example

2.3 AUTOMATION TOOL

EVOLVED-5G uses Jenkins (already introduced in D3.2, section 2.3) [8] as an open-source pipeline automation tool.

Indeed, Jenkins automates all the testing of the Certification Process. Some of these tests can run in parallel such as the Network Application quality assessment tests that rely on external tools, as shown in Figure 7.

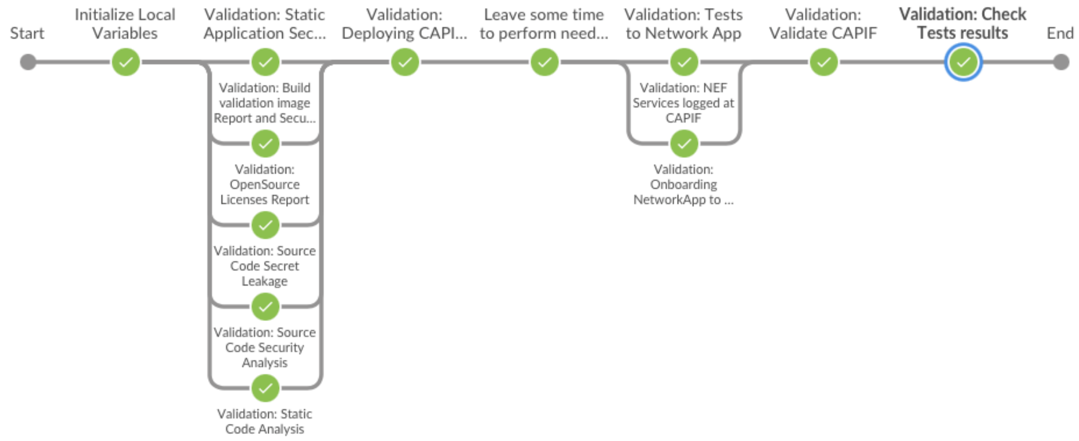

Start
Initialize Local Variables
Validation: Static Application Sec...
Validation: Deploying CAPI...
Leave some time to perform need...
Validation: Tests to Network App
Validation: Validate CAPIF
Validation: Check Tests results
End
Validation: Build validation image Report and Secu...
Validation: OpenSource Licenses Report
Validation: Source Code Secret Leakage
Validation: Source Code Security Analysis
Validation: Static Code Analysis
Validation: NEF Services logged at CAPIF
Validation: Onboarding NetworkApp to ...

Figure 7: Certification Pipeline graphic visualization

Some of the steps of the Certification process are related to Certification Environment assessment and preparation, for instance, deploying CAPIF, NEF and TSN tools for the Network Application to interact with.

The steps defined for the Certification process are described in section 4.2.

2.4 Infrastructure Deployment Tool

Initially, Terraform tool was initially selected to manage Infrastructure as Code and create the Kubernetes infrastructure to deploy the Network Applications in those clusters. However, Kubernetes environments that were created for the Certification process and are described in section 2.6 do not require dynamic creation and destruction of Kubernetes cluster. Instead, stable Kubernetes clusters are used for this purpose.

Therefore, we have selected Helm [9] to deploy Network Applications in Kubernetes environments. Indeed, Helm is a popular open-source tool designed to simplify the deployment and management of applications within Kubernetes clusters. It provides a package manager-like approach for defining, installing, and upgrading applications as sets of pre-configured resources called "charts."

At its core, Helm consists of two major components: the Helm client and the Helm server, also known as Tiller (although Tiller has been deprecated in recent versions of Helm). The client-side component is a command-line interface that enables users to interact with Helm's functionality, while Tiller (or its equivalent) runs as a server-side component inside the Kubernetes cluster, facilitating the deployment and management of charts.

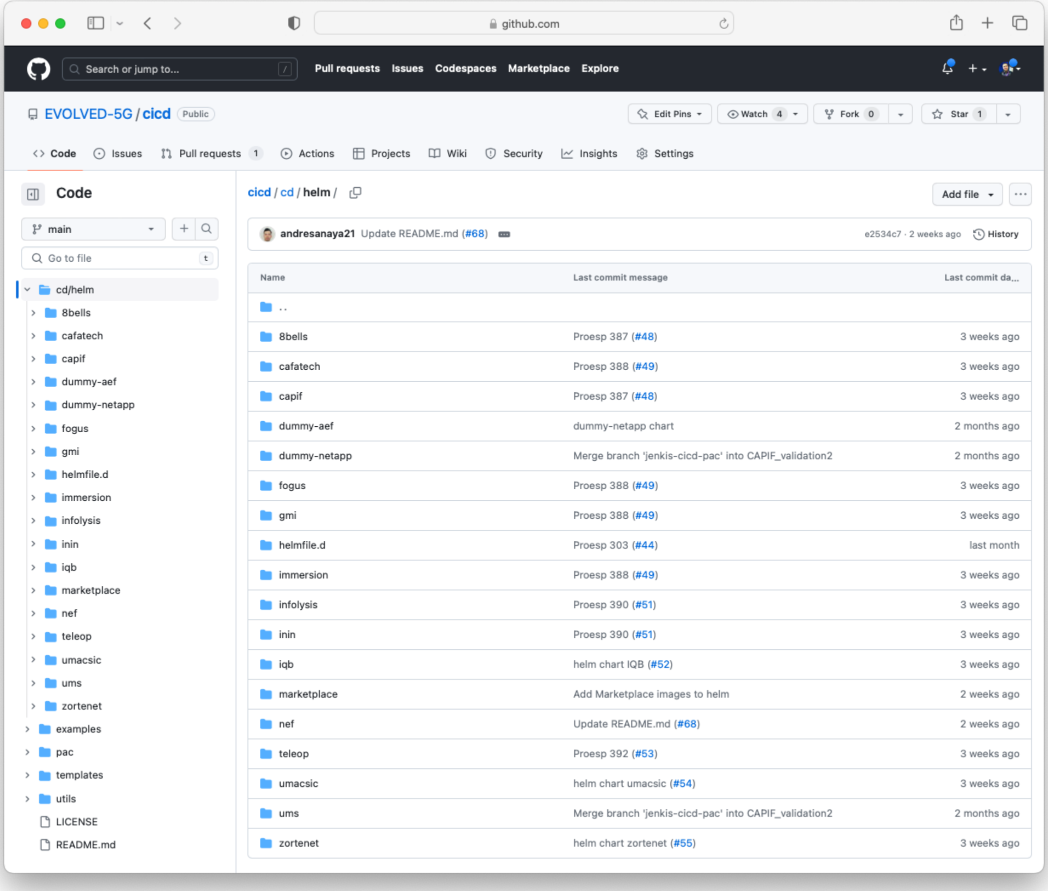
A chart, in Helm's terminology, is a collection of files that describe a specific application or service, including its Kubernetes manifests, dependencies, and configuration parameters. By leveraging charts, Helm enables the encapsulation and reusability of application definitions, making it easier to share and deploy complex applications with consistent configurations across different environments.

Using Helm, users can perform various operations, such as searching and installing charts from Helm repositories, customizing chart values during installation, and upgrading or rolling back deployments. Helm also provides a mechanism called "hooks" to execute pre-defined actions during specific lifecycle events of the deployed application, like running database migrations or initializing data.

Furthermore, Helm offers a templating engine that allows users to dynamically generate Kubernetes manifests by injecting values and variables into the chart templates. This feature enables configuration flexibility and promotes the separation of application logic from infrastructure concerns.

With Helm, managing the lifecycle of applications running on Kubernetes becomes more streamlined and less error prone. It promotes reusability, standardization, and versioning of deployments, which greatly simplifies the process of installing, upgrading, and maintaining complex containerized applications within Kubernetes clusters. This lifecycle management positions Helm as a perfect match for EVOLVED-5G as many of Helm features match the requirements needed for the realization of the Network Apps lifecycle. Moreover, Helm allows EVOLVED-5G Certification Pipeline to deploy the Network Applications as many times as needed in a deterministic way which is, by definition, how the Certification process was designed.

Helm chart files for Network Applications and auxiliary tools in EVOLVED-5G are located at https://github.com/EVOLVED-5G/cicd/tree/main/cd/helm



andresanaya21 Update README.md (#68) e2534c7 · 2 weeks ago

Name	Last commit message	Last commit da...
..		
8bells	Proesp 387 (#48)	3 weeks ago
cafatech	Proesp 388 (#49)	3 weeks ago
capif	Proesp 387 (#48)	3 weeks ago
dummy-aef	dummy-netapp chart	2 months ago
dummy-netapp	Merge branch 'jenkis-cicd-pac' into CAPIF_validation2	2 months ago
fogus	Proesp 388 (#49)	3 weeks ago
gmi	Proesp 388 (#49)	3 weeks ago
helmfile.d	Proesp 303 (#44)	last month
immersion	Proesp 388 (#49)	3 weeks ago
infolysis	Proesp 390 (#51)	3 weeks ago
inin	Proesp 390 (#51)	3 weeks ago
iqb	helm chart IQB (#52)	3 weeks ago
marketplace	Add Marketplace images to helm	2 weeks ago
nef	Update README.md (#68)	2 weeks ago
teleop	Proesp 392 (#53)	3 weeks ago
umacsic	helm chart umacsic (#54)	3 weeks ago
ums	Merge branch 'jenkis-cicd-pac' into CAPIF_validation2	2 months ago
zortenet	helm chart zortenet (#55)	3 weeks ago

Figure 8: Helm chart files used by Certification Process

2.5 Robot Framework Testing Engine

Robot Framework [10] is a generic open-source automation framework that has been described in detail in deliverable 3.2. Robot Framework has been used in EVOLVED-5G to automate some functional and performance tests for the auxiliary tools used in the certification process.

Every time Jenkins deploys CAPIF, NEF or TSN, automated tests are performed to verify that the deployment of tools has been completed successfully, and that the component is fully functional. These components will be used by Network Applications, and it is critical to certify that they are working properly. If an issue is detected during Certification process between the Network Application and one of these components, the latter will be related to the Network Application, as the components have been tested previously.

2.6 Container Execution Platforms

2.6.1 OpenShift 4.1

An initial description of Openshift Kubernetes platform was provided in D3.2. EVOLVED-5G CI/CD toolset includes a Jenkins slave to interact with Openshift infrastructure. Thus, Jenkins by using Helm charts can deploy, manage and destroy containers in Openshift execution platform.

Following the master-slave Jenkins design pattern, the architecture of this scenario with Openshift is the following:

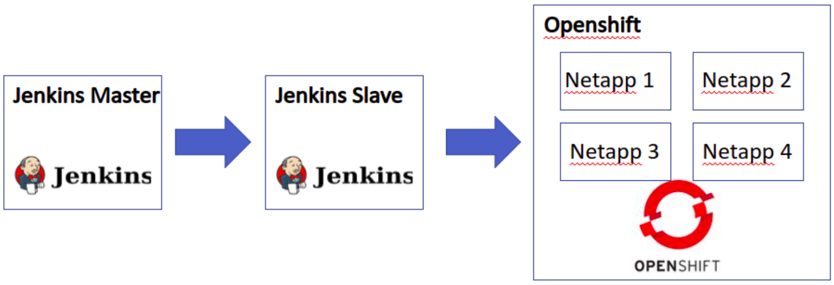
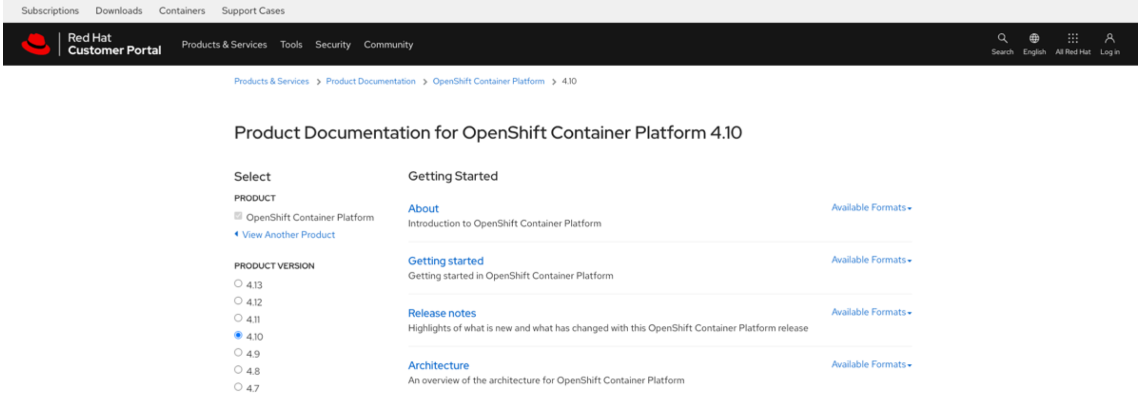


Figure 9: Jenkins slave for Openshift

Since the release of D3.2, Openshift platform has been updated to newer versions in Telefónica infrastructure. By the time this deliverable is being submitted, the version used for EVOLVED-5G is 4.10 as shown in Figure 10.



Products & Services > Product Documentation > OpenShift Container Platform > 4.10

Product Documentation for OpenShift Container Platform 4.10

Select

PRODUCT

OpenShift Container Platform

View Another Product

PRODUCT VERSION

○ 4.13
○ 4.12
○ 4.11
● 4.10
○ 4.9
○ 4.8
○ 4.7

Getting Started

About — Introduction to OpenShift Container Platform — Available Formats

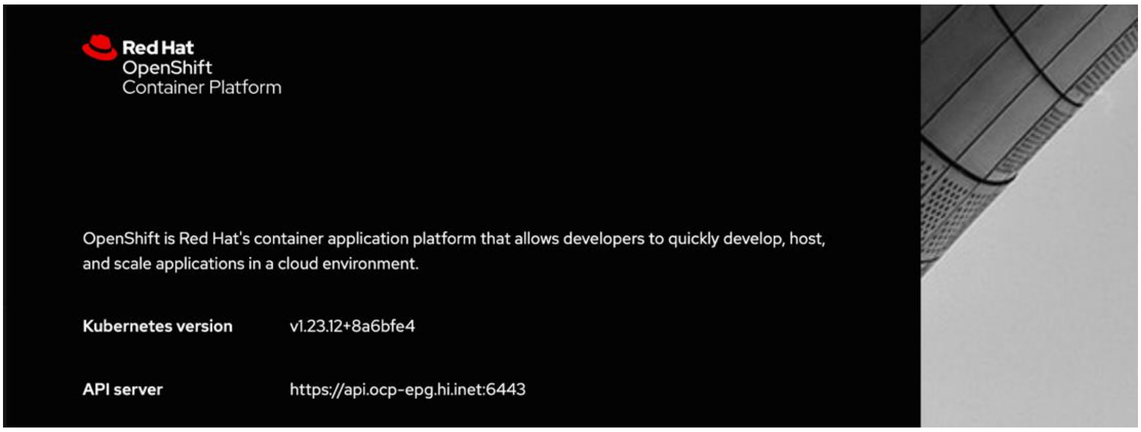
Getting started — Getting started in OpenShift Container Platform — Available Formats

Release notes — Highlights of what is new and what has changed with this OpenShift Container Platform release — Available Formats

Architecture — An overview of the architecture for OpenShift Container Platform — Available Formats

Figure 10: Openshift version in CICD Environment

This version of Openshift includes new Kubernetes versions as displayed in the following picture.



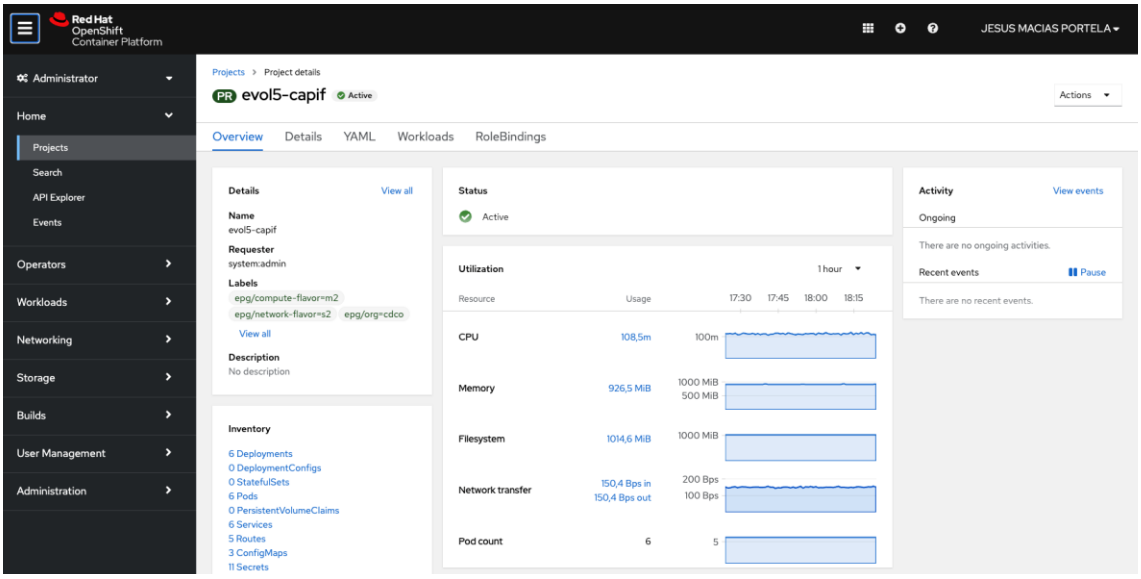
OpenShift is Red Hat's container application platform that allows developers to quickly develop, host, and scale applications in a cloud environment.

Kubernetes version v1.23.12+8a6bfe4

API server https://api.ocp-epg.hi.inet:6443

Figure 11: Kubernetes version in CICD Environment

Figure 12 displays the user interface that appears with all the information about the deployments in Openshift 4.10.



Projects > Project details

PR evol5-capif Active

Details

Name: evol5-capif

Requester: system:admin

Labels: epg/compute-flavor=m2, epg/network-flavor=s2, epg/org=cdco

Description: No description

Inventory

6 Deployments
0 DeploymentConfigs
0 StatefulSets
6 Pods
0 PersistentVolumeClaims
6 Services
5 Routes
3 ConfigMaps
11 Secrets

Status: Active

Utilization

Resource	Usage	
CPU	108,5m	100m
Memory	926,5 MiB	1000 MiB 500 MiB
Filesystem	1014,6 MiB	1000 MiB
Network transfer	150,4 Bps in 150,4 Bps out	200 Bps 100 Bps
Pod count	6	5

Activity

Ongoing

There are no ongoing activities.

Recent events

There are no recent events.

Figure 12: Openshift dashboard for EVOLVED-5G

2.6.2 Malaga Platform Kubernetes

The Kubernetes deployment in Málaga is composed of 4 worker nodes, with one dedicated to storage, and all of which are managed by 3 master nodes. All the nodes are distributed in three different physical servers. This multi-master deployment is installed on a bare-metal structure that makes use of MetalLB [11] (an open-source load balancer that is specifically tailored for bare-metal deployments) along with a Nginx Ingress controller that implements the network interfaces in the load balancer and controls the access to the system. Direct access and management of the resources is performed through the Kubectl.

Due to security considerations, the cluster makes use of role-based access scheme (RBAC), which provides isolation between different users and reduces access to the minimum required for each user.

2.6.3 Athens Platform Kubernetes

The description of the Athens K8s cluster can also be found in D3.3, however, for coherence and clarity, a brief description of thecluster is described below.

The K8s cluster deployed in NCSRD premises consists of three virtual machines, one master node and two worker nodes. Each node is equipped with 2 vCPUs and 4GB RAM. Access to the nodes is achieved through VPN connectivity and the overall setup of the cluster has been updated to K8s version 1.26. To enhance network connectivity, the cluster is equipped with Cilium, which provides flexibility through specific configurations and policies. Additionally, a NGINX Ingress Controller has been deployed, using the NodePort service exposure approach to route incoming requests to the services.

2.6.4 COSMOTE Platform Kubernetes

The Kubernetes cluster created for EVOLVED-5G project in COSMOTE is part of the Athens certification environment with the purpose of hosting various applications and services as necessary for the Network applications certification. The cluster architecture consists of a master node responsible for managing the cluster's control plane and two worker nodes that handle the execution of application workloads, the cluster's version is 1.25.6. Additionally, an extra virtual machine (VM) has been provisioned as a VPN server, allowing authorized personnel to securely access the cluster remotely.

In the current setup, the master node is equipped with 6 virtual CPUs (vCPUs), 12 GB of memory, and a 100 GB hard disk, ensuring sufficient resources for efficient orchestration of the cluster's operations. Each worker node has 16 GB of memory, enabling efficient execution and scaling of application workloads. Calico is utilized as the networking plugin, providing secure and scalable communication between pods and services within the cluster. The cluster was built using Rancher, a popular Kubernetes management platform that provides powerful tools and an intuitive user interface to simplify cluster creation, management, and monitoring. Rancher streamlines the deployment process and offers additional features for efficiently managing and operating Kubernetes clusters.

The cluster leverages Docker Engine as the container runtime, enabling the containerization and management of applications in Docker containers. Docker provides a consistent and isolated runtime environment, facilitating the packaging, distribution, and deployment of applications with ease and portability.

For monitoring the cluster's health and performance, Prometheus is implemented as the monitoring solution. Prometheus collects metrics and offers valuable insights into resource utilization, application performance, and cluster health, empowering proactive management and troubleshooting.

Future plans for the cluster involve scaling and expansion based on the evolving needs of the project. This can include adding more worker nodes to accommodate increased workload demands and considering the adoption of advanced Kubernetes features to enhance resilience and scalability.

3 CERTIFICATION TOOLS

3.1 SOFTWARE QUALITY TOOLS

3.1.1 NEF and CAPIF Services

During the certification process in the K8s cluster, the NEF Emulator component is deployed to enable Network Applications to undergo certification using the exposed services. One notable feature is the implementation of the logging service in the CAPIF Core Function (described later on). This API allows the NEF Emulator to push the consumed APIs by the Network Applications into the CCF's database, facilitating assessment and evaluation.
Details about the implementation aspects of the NEF emulator are described in D3.1 and D3.3. Additionally, a comprehensive description of the NEF services be found in D4.1

Respectively, CAPIF has been thoroughly described in Deliverable 3.3 (section 6.1.3). The latest release of CAPIF Core Function tool includes new methods that are fundamental for certifying API usage during certification testing. Those methods are CAPIF_Logging_API_Invocation_API and CAPIF_Auditing_API.

CAPIF_Logging_API_Invocation_API allows API Exposing Functions to "Log" API invocations from API Invokers. These invocations can be logged individually or in groups. By using the Logging API, AEFs can register in CAPIF Core Function API invocations received from API Invokers, leaving evidence that the API Invokers are using exposed APIs properly.

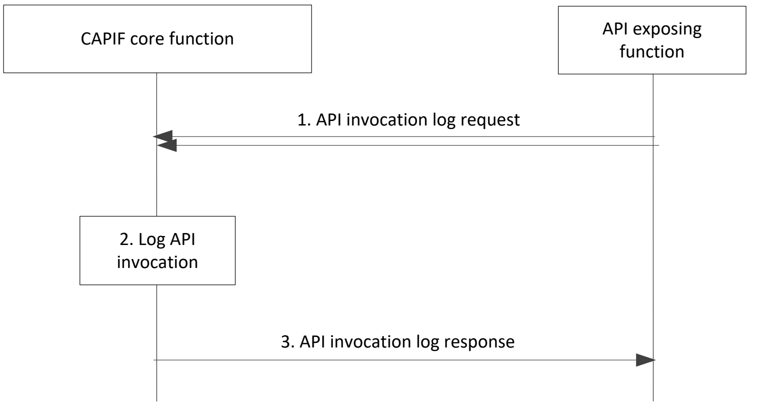


Figure 13: API Logging Procedure described in TS 23.222

EVOLVED-5G leverages these evidences for certifying the APIs exposed by NEF or TSN are being consumed by Network Applications being certified.

Both components, NEF Emulator and TSN Application Function have integrated CAPIF Logging capabilities, therefore, every time a Network Application uses any of the NEF Emulator or TSN AF exposed APIs in CAPIF, an API Invocation log is generated in CAPIF Core Function. These logs are used by EVOLVED-5G Certification pipeline as evidence proof of API consumption from the Network Applications.

Subsequently, for certifying the usage of network core APIs by a Network Application, an instance of the CAPIF [12] is deployed during the certification process. The network core APIs that are discovered through CAPIF are the ones exposed by the NEF emulator (NEF services) and TSN Application Function (TSN services). The certification pipeline assesses the CAPIF deployment process and certifies the usage of NEF and TSN APIs using the information provided by API Logging capabilities of CAPIF Core Function.

3.1.2 TSN Services

For certifying the usage of deterministic communication in the Network Application, an instance of the TSN FrontEnd [13] is deployed during the certification process.

Considering that during certification the actual network infrastructure is not available, the TSN FrontEnd is configured in backend-less mode, this is, the frontend is ready for accepting configuration requests, however, these are not sent to the actual infrastructure and only pre-defined responses are returned by the API.

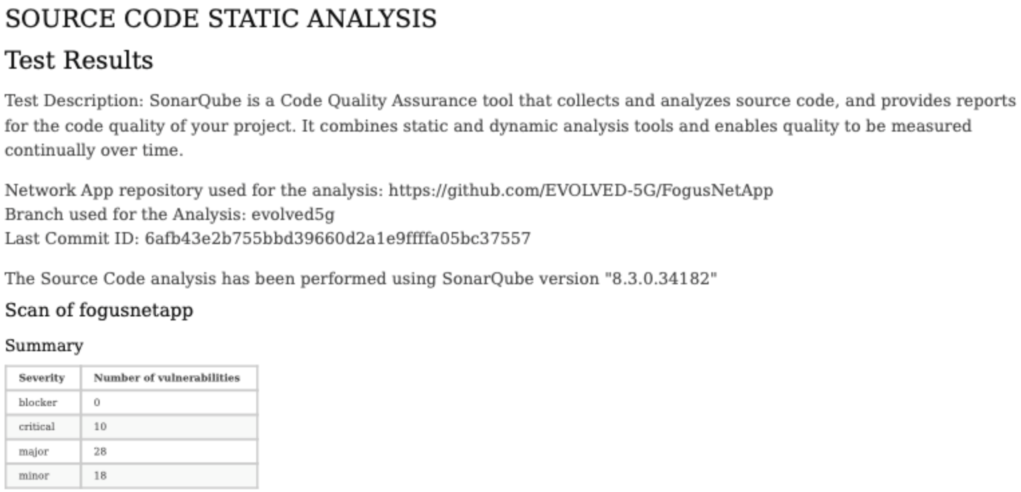
The TSN FrontEnd is integrated with the CAPIF framework in such way that information about every request received is recorded by using the logging APIs of CAPIF. This allows the certification pipeline to assess if the Network Application has sent the correct configuration requests to the TSN FrontEnd, ensuring the compatibility of the Network Application with the TSN Services.

3.2 Security Certification

As described in Deliverable 3.2 (section 3.2) several market tools have been selected to implement most of the tests required to assess the security certification. Security covers both the source of the Network Application and the binary images built of the Network Applications.

3.2.1 Static Code Analysis

The static code analysis is performed using SonarQube tools described in Deliverable 3.2, section 3.2.1. This tool provides a report of the issues detected that is integrated in the certification report:



SOURCE CODE STATIC ANALYSIS

Test Results

Test Description: SonarQube is a Code Quality Assurance tool that collects and analyzes source code, and provides reports for the code quality of your project. It combines static and dynamic analysis tools and enables quality to be measured continually over time.

Network App repository used for the analysis: https://github.com/EVOLVED-5G/FogusNetApp
Branch used for the Analysis: evolved5g
Last Commit ID: 6afb43e2b755bbd39660d2a1e9ffffa05bc37557

The Source Code analysis has been performed using SonarQube version "8.3.0.34182"

Scan of fogusnetapp

Summary

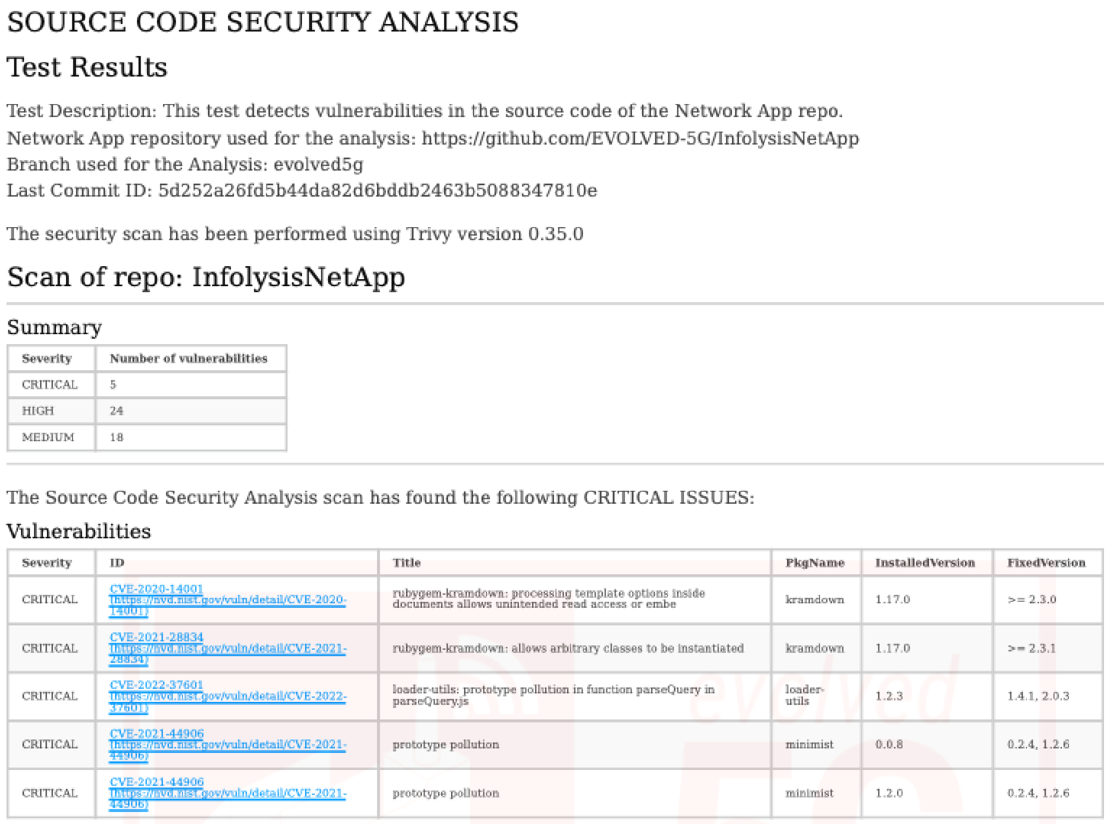
Severity	Number of vulnerabilities
blocker	0
critical	10
major	28
minor	18

Figure 14: Information reported by SonarQube tool

The report shows the repository URL under certification, the branch used for the analysis and the last commit ID to precisely define the Source Code version under certification. SonarQube version is displayed, considering that this can be evolved over time, and the scan results are summarized in the table, with four different severity levels: blocker issues, critical, major and minor.

3.2.2 Source Code Security Analysis

After the Static Code Security analysis, a Source Code Security Analysis is performed using Trivy commercial tool. Trivy utilizes three severity levels to categorize issues detected: Critical, High and Medium. Critical issues are attached in the report with the information for the Developer to fix the issues if are to be identified and are available.



SOURCE CODE SECURITY ANALYSIS

Test Results

Test Description: This test detects vulnerabilities in the source code of the Network App repo.
Network App repository used for the analysis: https://github.com/EVOLVED-5G/InfolysisNetApp
Branch used for the Analysis: evolved5g
Last Commit ID: 5d252a26fd5b44da82d6bddb2463b5088347810e

The security scan has been performed using Trivy version 0.35.0

Scan of repo: InfolysisNetApp

Summary

Severity	Number of vulnerabilities
CRITICAL	5
HIGH	24
MEDIUM	18

The Source Code Security Analysis scan has found the following CRITICAL ISSUES:

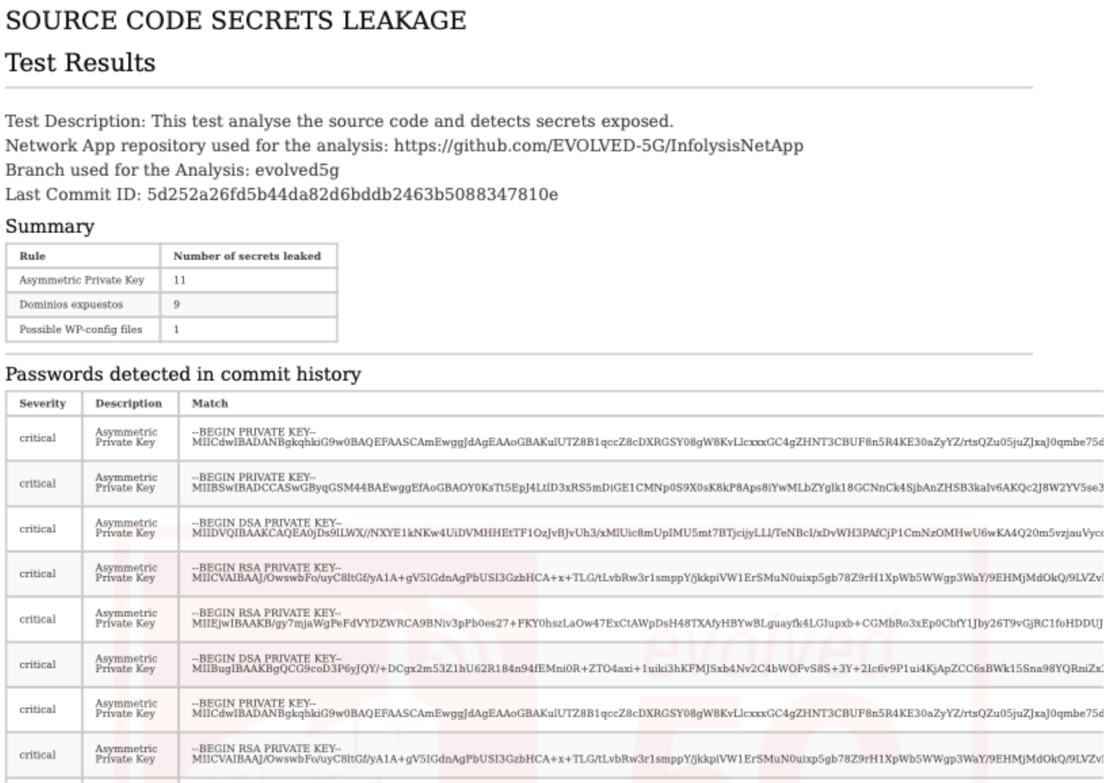
Vulnerabilities

Severity	ID	Title	PkgName	InstalledVersion	FixedVersion
CRITICAL	CVE-2020-14001 (https://nvd.nist.gov/vuln/detail/CVE-2020-14001)	rubygem-kramdown: processing template options inside documents allows unintended read access or embe	kramdown	1.17.0	>= 2.3.0
CRITICAL	CVE-2021-28834 (https://nvd.nist.gov/vuln/detail/CVE-2021-28834)	rubygem-kramdown: allows arbitrary classes to be instantiated	kramdown	1.17.0	>= 2.3.1
CRITICAL	CVE-2022-37601 (https://nvd.nist.gov/vuln/detail/CVE-2022-37601)	loader-utils: prototype pollution in function parseQuery in parseQuery.js	loader-utils	1.2.3	1.4.1, 2.0.3
CRITICAL	CVE-2021-44906 (https://nvd.nist.gov/vuln/detail/CVE-2021-44906)	prototype pollution	minimist	0.0.8	0.2.4, 1.2.6
CRITICAL	CVE-2021-44906 (https://nvd.nist.gov/vuln/detail/CVE-2021-44906)	prototype pollution	minimist	1.2.0	0.2.4, 1.2.6

Figure 15: Information reported by Trivy tool

3.2.3 Source Code Secrets Leakage

After the Source Code Security analysis, a Source Code Secrets Leakage analysis is performed using Trivy commercial tool. Trivy is able to detect different leakage problems and report them. Critical issues are attached in the report with the information for the Developer to fix them. An indicative paradigm regarding the source code secrets leakage is presented in Figure 16 below.



SOURCE CODE SECRETS LEAKAGE

Test Results

Test Description: This test analyse the source code and detects secrets exposed.
Network App repository used for the analysis: https://github.com/EVOLVED-5G/InfolysisNetApp
Branch used for the Analysis: evolved5g
Last Commit ID: 5d252a26fd5b44da82d6bddb2463b5088347810e

Summary

Rule	Number of secrets leaked
Asymmetric Private Key	11
Dominios expuestos	9
Possible WP-config files	1

Passwords detected in commit history

Severity	Description	Match
critical	Asymmetric Private Key	--BEGIN PRIVATE KEY-- MIICdwIBADANBgkqhkiG9w0BAQEFAASCAmEwggJdAgEAAoGBAKulUTZ8B1qccZ8cDXRGSY08gW8KvLlcxxxGC4gZHNT3CBUF8n5R4KE30aZyYZ/rtsQZu05juZJxaJ0qmbe75d
critical	Asymmetric Private Key	--BEGIN PRIVATE KEY-- MIIBSwIBADCCASwGByqGSM44BAEwggEfAoGBAOY0KsTt5EpJ4LtlD3xRS5mDiGE1CMNp0S9X0sK8kP8Aps8iYwMLbZYglk18GCNnCk4SjbAnZHSB3kaIv6AKQc2J8W2YV5se3
critical	Asymmetric Private Key	--BEGIN DSA PRIVATE KEY-- MIIDVQIBAAKCAQEA0jDs9lLWX//NXYE1kNKw4UiDVMHHEtTF1OzJvBJvUh3/xMlUic8mUpIMU5mt7BTjcijyLLl/TeNBcI/xDvWH3PAfCjP1CmNzOMHwU6wKA4Q20m5vzjauVyc
critical	Asymmetric Private Key	--BEGIN RSA PRIVATE KEY-- MIICVAIBAAJ/OwswbFo/uyC8ltGf/yA1A+gV5IGdnAgPbUSI3GzbHCA+x+TLG/tLvbRw3r1smppY/jkkpiVW1ErSMuN0uixp5gb78Z9rH1XpWb5WWgp3WaY/9EHMjMdOkQ/9LVZv
critical	Asymmetric Private Key	--BEGIN RSA PRIVATE KEY-- MIIEjwIBAAKB/gy7mjaWgPeFdVYDZWRCA9BNiv3pPb0es27+FKY0hszLaOw47ExCtAWpDsH48TXAfyHBYwBLguayfk4LGIupxb+CGMbRo3xEp0CbfY1Jby26T9vGjRC1foHDDUJ
critical	Asymmetric Private Key	--BEGIN DSA PRIVATE KEY-- MIIBugIBAAKBgQCG9coD3P6yJQY/+DCgx2m53Z1hU62R184n94fEMni0R+ZTO4axi+1uiki3hKFMJSxb4Nv2C4bWOFvS8S+3Y+2Ic6v9P1ui4KjApZCC6sBWk15Sna98YQRniZx
critical	Asymmetric Private Key	--BEGIN PRIVATE KEY-- MIICdwIBADANBgkqhkiG9w0BAQEFAASCAmEwggJdAgEAAoGBAKulUTZ8B1qccZ8cDXRGSY08gW8KvLlcxxxGC4gZHNT3CBUF8n5R4KE30aZyYZ/rtsQZu05juZJxaJ0qmbe75d
critical	Asymmetric Private Key	--BEGIN RSA PRIVATE KEY-- MIICVAIBAAJ/OwswbFo/uyC8ltGf/yA1A+gV5IGdnAgPbUSI3GzbHCA+x+TLG/tLvbRw3r1smppY/jkkpiVW1ErSMuN0uixp5gb78Z9rH1XpWb5WWgp3WaY/9EHMjMdOkQ/9LVZv

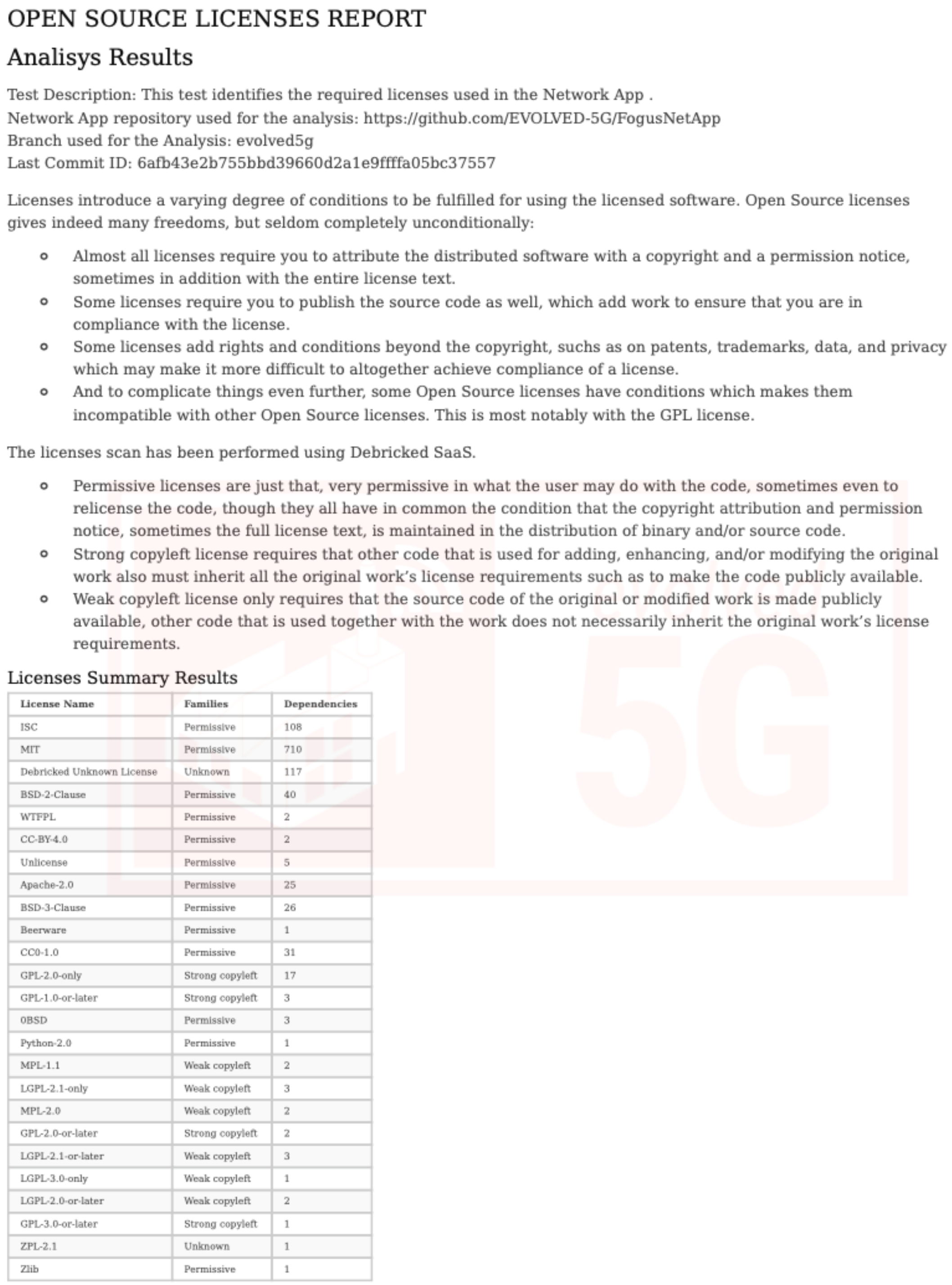
Figure 16: Secret leakage issues reported by Trivy tool

3.3 MARKETPLACE CERTIFICATION

3.3.1 Use Policy/Terms of service/ License files/ Open SourceScan Report

Before publishing a Network Application in the Marketplace, a License dependency scan is performed to identify all Licenses involved in the Product that will be commercialized in the Marketplace. For this purpose, Debricked tool has been selected to perform this analysis. Debricked tool was initially described in Deliverable 3.2 section 3.3.1.

Results of this analysis are integrated in the report for the Network Application with the purpose of informing the developer to attach the required license information to his product when publishing it in the Marketplace. A representative example of the aforementioned license dependency is illustrated in Figure 17.



OPEN SOURCE LICENSES REPORT

Analisys Results

Test Description: This test identifies the required licenses used in the Network App .
Network App repository used for the analysis: https://github.com/EVOLVED-5G/FogusNetApp
Branch used for the Analysis: evolved5g
Last Commit ID: 6afb43e2b755bbd39660d2a1e9ffffa05bc37557

Licenses introduce a varying degree of conditions to be fulfilled for using the licensed software. Open Source licenses gives indeed many freedoms, but seldom completely unconditionally:

- Almost all licenses require you to attribute the distributed software with a copyright and a permission notice, sometimes in addition with the entire license text.
- Some licenses require you to publish the source code as well, which add work to ensure that you are in compliance with the license.
- Some licenses add rights and conditions beyond the copyright, suchs as on patents, trademarks, data, and privacy which may make it more difficult to altogether achieve compliance of a license.
- And to complicate things even further, some Open Source licenses have conditions which makes them incompatible with other Open Source licenses. This is most notably with the GPL license.

The licenses scan has been performed using Debricked SaaS.

- Permissive licenses are just that, very permissive in what the user may do with the code, sometimes even to relicense the code, though they all have in common the condition that the copyright attribution and permission notice, sometimes the full license text, is maintained in the distribution of binary and/or source code.
- Strong copyleft license requires that other code that is used for adding, enhancing, and/or modifying the original work also must inherit all the original work's license requirements such as to make the code publicly available.
- Weak copyleft license only requires that the source code of the original or modified work is made publicly available, other code that is used together with the work does not necessarily inherit the original work's license requirements.

Licenses Summary Results

License Name	Families	Dependencies
ISC	Permissive	108
MIT	Permissive	710
Debricked Unknown License	Unknown	117
BSD-2-Clause	Permissive	40
WTFPL	Permissive	2
CC-BY-4.0	Permissive	2
Unlicense	Permissive	5
Apache-2.0	Permissive	25
BSD-3-Clause	Permissive	26
Beerware	Permissive	1
CC0-1.0	Permissive	31
GPL-2.0-only	Strong copyleft	17
GPL-1.0-or-later	Strong copyleft	3
0BSD	Permissive	3
Python-2.0	Permissive	1
MPL-1.1	Weak copyleft	2
LGPL-2.1-only	Weak copyleft	3
MPL-2.0	Weak copyleft	2
GPL-2.0-or-later	Strong copyleft	2
LGPL-2.1-or-later	Weak copyleft	3
LGPL-3.0-only	Weak copyleft	1
LGPL-2.0-or-later	Weak copyleft	2
GPL-3.0-or-later	Strong copyleft	1
ZPL-2.1	Unknown	1
Zlib	Permissive	1

Figure 17: License analysis result provided by Debriked tool

3.3.2 Valid Container Image and/or End-point details

After the Secret Leakage analysis and building the images of the Network Application that will undergo the certification process, the container images need to be analyzed. Source Code of the Network Application has been analyzed in the previous tests, but when building the container images, many other components are integrated into the image, and those components can be subject to also containing security vulnerabilities (libraries, Operative system…). The Image Security Analysis is performed using Trivy tool as well, and four categories of issues will be reported: Critical, High, Medium and Low issues. Again, for Critical issues, a full description is attached, the version of the affected component is listed, and versions of the same component with the issue fixed are displayed if available, as shown in Figure 18.

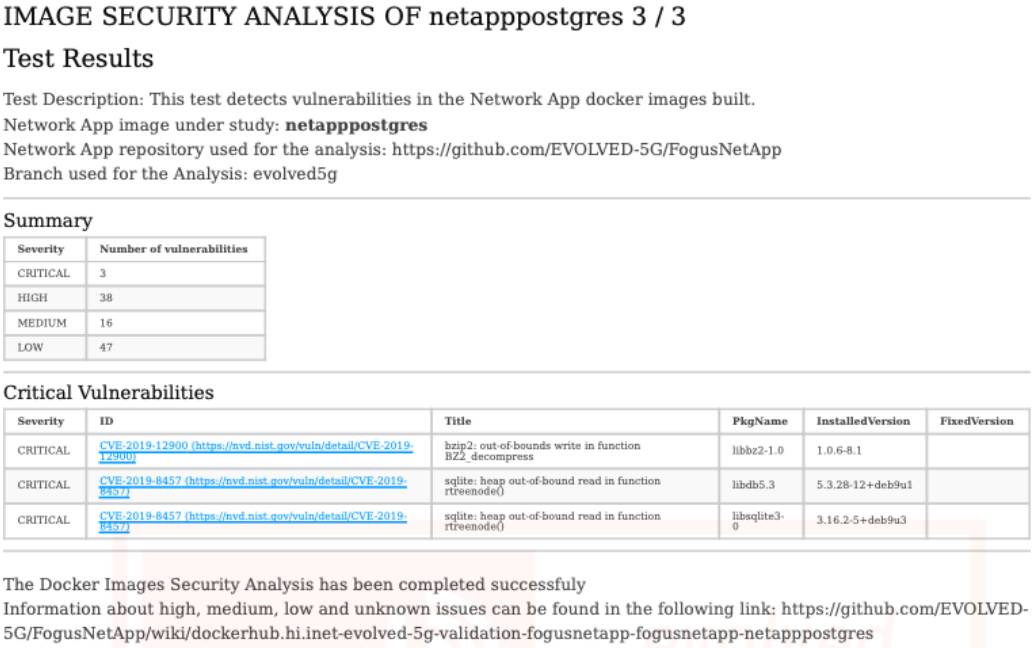


IMAGE SECURITY ANALYSIS OF netapppostgres 3 / 3

Test Results

Test Description: This test detects vulnerabilities in the Network App docker images built.
Network App image under study: **netapppostgres**
Network App repository used for the analysis: https://github.com/EVOLVED-5G/FogusNetApp
Branch used for the Analysis: evolved5g

Summary

Severity	Number of vulnerabilities
CRITICAL	3
HIGH	38
MEDIUM	16
LOW	47

Critical Vulnerabilities

Severity	ID	Title	PkgName	InstalledVersion	FixedVersion
CRITICAL	CVE-2019-12900 (https://nvd.nist.gov/vuln/detail/CVE-2019-12900)	bzip2: out-of-bounds write in function BZ2_decompress	libbz2-1.0	1.0.6-8.1	
CRITICAL	CVE-2019-8457 (https://nvd.nist.gov/vuln/detail/CVE-2019-8457)	sqlite: heap out-of-bound read in function rtreenode()	libdb5.3	5.3.28-12+deb9u1	
CRITICAL	CVE-2019-8457 (https://nvd.nist.gov/vuln/detail/CVE-2019-8457)	sqlite: heap out-of-bound read in function rtreenode()	libsqlite3-0	3.16.2-5+deb9u3	

The Docker Images Security Analysis has been completed successfuly
Information about high, medium, low and unknown issues can be found in the following link: https://github.com/EVOLVED-5G/FogusNetApp/wiki/dockerhub.hi.inet-evolved-5g-validation-fogusnetapp-fogusnetapp-netapppostgres

Figure 18: Container image issues reported by Trivy tool

4 CERTIFICATION PROCESS

4.1 CERTIFICATION PROCESS DESCRIPTION

Certification Process was outlined in Deliverable D3.2 section 4.1. At that point, EVOLVED-5G was not mature enough to provide all details on the implementation of this process. In this deliverable, the information provided is much more detailed and elaborated as it goes hand in hand with the implementation of the Certification pipeline being also developed in the context of WP5, using the tools described in this deliverable.

The certification process is responsible for executing the tests that certify that Network Applications are compliant with the definition of an EVOLVED-5G Network Application. Criteria for certifying a Network App were already described in Deliverable 2.2 [14], and it is repeated here for facilitating the reader comprehension:

- The Network Application is secure
- The Network Application can be deployed in Cloud infrastructure
- The Network Application is Cloud Native
- The Network Application uses CAPIF APIs to Discover and Consume APIs
- The Network Application uses NEF APIs to interact with 5G Infrastructure

By definition, the design principles for the Certification Process can be listed as *Repeatable, Deterministic, Traceable and Automated*. These four principles make the Certification Process a tool that Network Application developers can use as many times as they need to complete the certification.

Updating the information provided in Deliverable 2.2, TSN Services have been added to the Certification Process elements, as TSN application Function has been developed inside the project (see section 3.1.2).

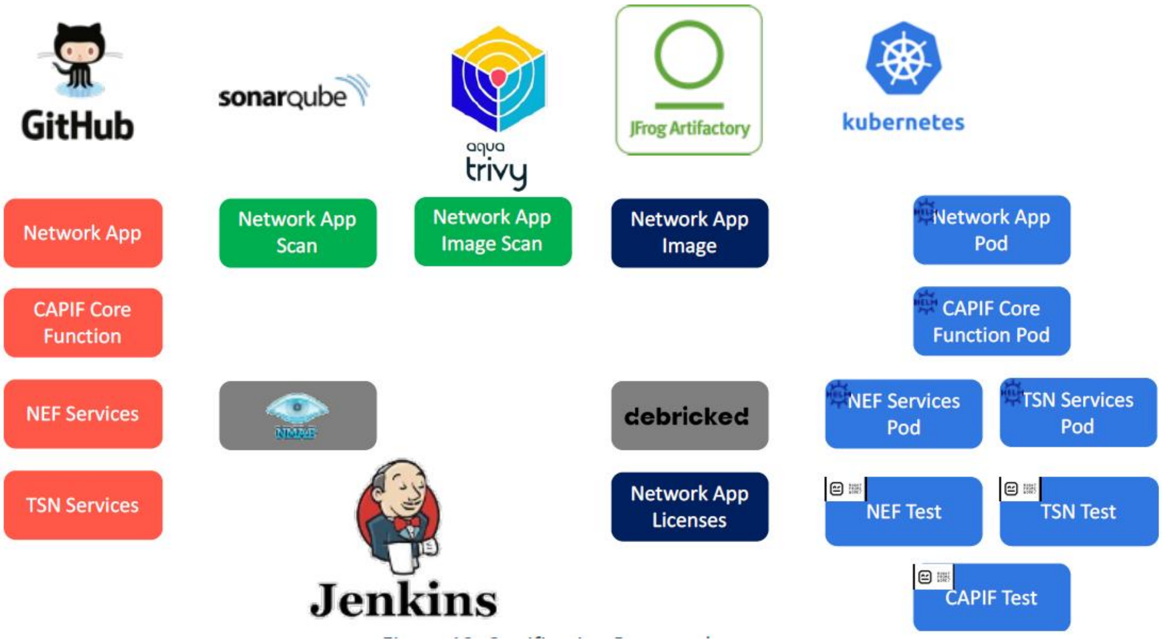


Figure 19: Certification Process elements

The starting point of the Certification Process is always a Validated Network Application. During Validation process, the Network Application image is uploaded to Open repository (Artifactory) in the Validation folder.

4.2 Certification Pipeline

The Certification Pipeline running in Jenkins is responsible to perform the Certification Process defined in EVOLVED-5G, as outlined in D3.2. It uses nested pipelines for some of the operations and both internal tools (Robot Framework) and external tools (Debricked, Trivy, Robot Framework) to perform some of the certification tests defined.

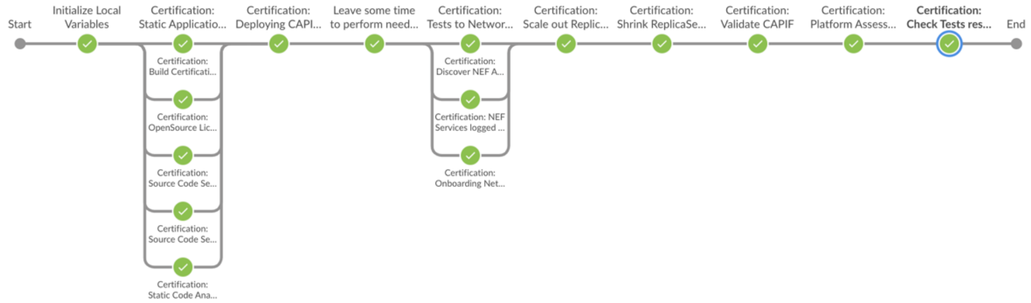
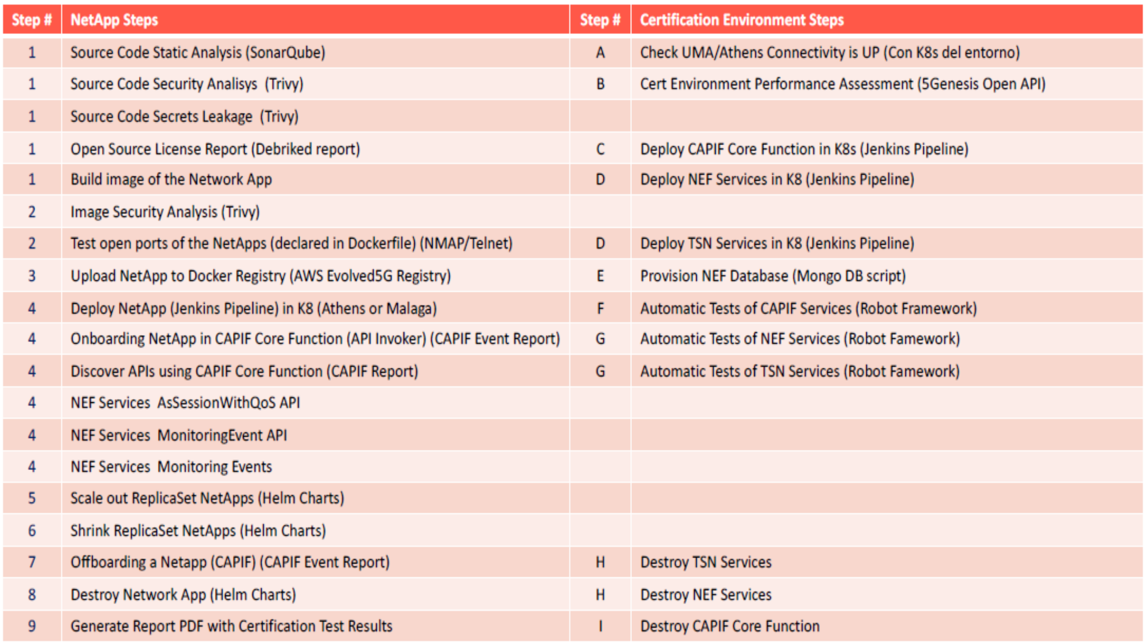

Start
Initialize Local Variables
Certification: Static Applicatio...
Certification: Build Certificati...
Certification: OpenSource Lic...
Certification: Source Code Se...
Certification: Source Code Se...
Certification: Static Code Ana...
Certification: Deploying CAPI...
Leave some time to perform need...
Certification: Tests to Networ...
Certification: Discover NEF A...
Certification: NEF Services logged ...
Certification: Onboarding Net...
Certification: Scale out Replic...
Certification: Shrink ReplicaSe...
Certification: Validate CAPIF
Certification: Platform Assess...
Certification: Check Tests res...
End

Figure 20: View of the Certification Pipeline in executed Jenkins

The Certification pipeline is summarized in Figure 20 where some of the steps have been updated in comparison with initial design in D3.2, due to the redefinition of some of them and the parallelization of some steps in the pipeline.



Step #	NetApp Steps	Step #	Certification Environment Steps
1	Source Code Static Analysis (SonarQube)	A	Check UMA/Athens Connectivity is UP (Con K8s del entorno)
1	Source Code Security Analisys (Trivy)	B	Cert Environment Performance Assessment (5Genesis Open API)
1	Source Code Secrets Leakage (Trivy)		
1	Open Source License Report (Debriked report)	C	Deploy CAPIF Core Function in K8s (Jenkins Pipeline)
1	Build image of the Network App	D	Deploy NEF Services in K8 (Jenkins Pipeline)
2	Image Security Analysis (Trivy)		
2	Test open ports of the NetApps (declared in Dockerfile) (NMAP/Telnet)	D	Deploy TSN Services in K8 (Jenkins Pipeline)
3	Upload NetApp to Docker Registry (AWS Evolved5G Registry)	E	Provision NEF Database (Mongo DB script)
4	Deploy NetApp (Jenkins Pipeline) in K8 (Athens or Malaga)	F	Automatic Tests of CAPIF Services (Robot Framework)
4	Onboarding NetApp in CAPIF Core Function (API Invoker) (CAPIF Event Report)	G	Automatic Tests of NEF Services (Robot Famework)
4	Discover APIs using CAPIF Core Function (CAPIF Report)	G	Automatic Tests of TSN Services (Robot Famework)
4	NEF Services AsSessionWithQoS API		
4	NEF Services MonitoringEvent API		
4	NEF Services Monitoring Events		
5	Scale out ReplicaSet NetApps (Helm Charts)		
6	Shrink ReplicaSet NetApps (Helm Charts)		
7	Offboarding a Netapp (CAPIF) (CAPIF Event Report)	H	Destroy TSN Services
8	Destroy Network App (Helm Charts)	H	Destroy NEF Services
9	Generate Report PDF with Certification Test Results	I	Destroy CAPIF Core Function

Figure 21: Certification Steps in the Pipeline to complete Network App Certification

In the next two subsections, the certification steps listed in the above figure are outlined. The full description of the implementation will be reflected in D5.6 by the end of the project.

4.2.1 Network App steps

This section outlines the steps to be implemented in the Certification Pipeline to complete the Network App Certification. Some of the steps will run simultaneously as there are no dependencies between them.

Step 1: The first step includes a series of tests taking the Network App repository as input. These tests include the Static Code Analysis (described in subsection 3.2.1), Source Code Security Analysis (3.2.2), Source Code Secrets Leakage (3.2.3), Open-Source License Report (3.3.1) and finally the Network App is generated.

Step 2: After testing the source code, and with the Network App generated, it is time to analyze the Image for security issues. The Network App image includes not only the Network App source code analyzed in Step 1, but also many libraries and OS components that need to be analyzed as well. The Image Security analysis (section 3.3.2) is performed, and the open ports declared by the Network app are tested.

Step 3: With the certainty of a secured image of the Network App, the image is then uploaded to the AWS Docker Registry to make it available for the deployment phase (Step 4).

Step 4: Jenkins deploys the Network App in the selected environment for Certification (see section 2.6) from the AWS Docker Registry used in Step 3. Once deployed, the Network App will bootstrap and will start using CAPIF, NEF and TSN APIs. The usage of those APIs is registered using CAPIF Core Function tool and Jenkins will generate evidences of the APIs used during this phase.

Step 5 and Step 6: These steps will be exclusive for Certification, and the goal is to certify that the Network App is Cloud Native by scaling out the number of containers of the Network App and shrinking them. Kubernetes API will be used to verify that the containers expand and shrink properly.

Step 7: will offboard the Network App from CAPIF to finalize the usage of 5G capabilities from the platforms.

Step 8: Once offboarded from CAPIF, it is time to destroy the Network App and release the allocated resources for next executions of the Certification pipeline.

Step 9: Once all the tests are completed, Jenkins will take the JSON result files to generate the markdown files that finally are converted to PDF to be included in the Certification Report and will send this report to the Network app developer.

4.2.2 Certification Environment Steps

During Certification, Jenkins needs to prepare the environment deploying some tools for the Certification tests to work properly. The main environment related steps (Env steps) are described in this section.

Env Step A: Before starting any tests, Jenkins needs to certify that connectivity to the selected platform (Málaga, Athens) for Certification is working as many tests depend on this connectivity.

Env Step B: Connectivity is required but is not enough. Jenkins will execute a Performance Assessment experiment in the platforms to characterize the performance in which the tests will be executed. Specific thresholds will be defined for bandwidth and latency to guarantee that tests will be executed in an environment that is not restricting or affecting the tests results.

Env Step C: One of the components required to perform Network App tests is CAPIF Core Function. A clean deployment of CAPIF Core function will guarantee that CAPIF Core Function is not affected by previous interactions or behavior of other Network Apps.

Env Step D: The other two components required to perform Network App tests are NEF and TSN Services. Again, a clean deployment of NEF and TSN Services will guarantee that those services are not affected by previous interactions or behavior of other Network Apps.

Env Step E: NEF Services might require deploying some data to prepare the scenario simulated by NEF Emulator. If this data is needed, it will be uploaded to the NEF Emulator in this step.

Env Step F: As the CAPIF Core Function has been deployed in clean mode for testing, in order to guarantee that CAPIF Core Function APIs work properly, a set of functional tests will be executed using Robot Framework to certify that CAPIF Core Function APIs works properly.

Env Step G: Same approach is taken to test NEF and TSN tools after the clean deployment. Functional tests are executed to verify that the tools work properly.

Env Step H and I: After the Network App has interacted with CAPIF, NEF and TSN, and the Network App has been destroyed, we need to release the resources from NEF, TSN and CAPIF as well to clean the environment for future certifications.

After cleaning the environment, the Certification pipeline will terminate, and a new Certification can be launched to re-start the process with the same or different Network App.

4.3 Certification Report

The Certification Report is a PDF file generated for every Certification of a Network App that contains the results of the Certification tests executed over the Network App being certified. Upon successful completion, the Certification Report will contain a Fingerprint Unique ID that the developer is using for publishing the Network App in EVOLVED-5G Marketplace.

This Fingerprint ID is a unique ID generated by the Certification Pipeline that is stored in Artifactory along with the binary images of the Network Application that have been certified. This way the Network App developer, when publishing its Network App, must introduce this unique ID to refer to the Network App version certified, as described later in section 5.4.

The Certification Report is generated using markdown-pdf tool that converts markdown files into PDF. These markdown documents are generated using Jinja tool. For each Certification test, a JSON file is generated and stored in the Open Repository containing the result of the test.

The full chain of tools is showed in the following picture:

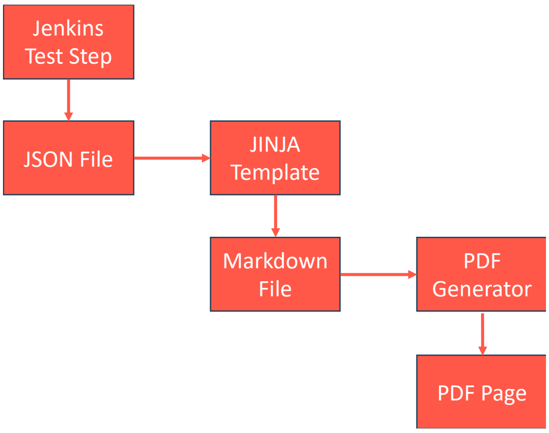
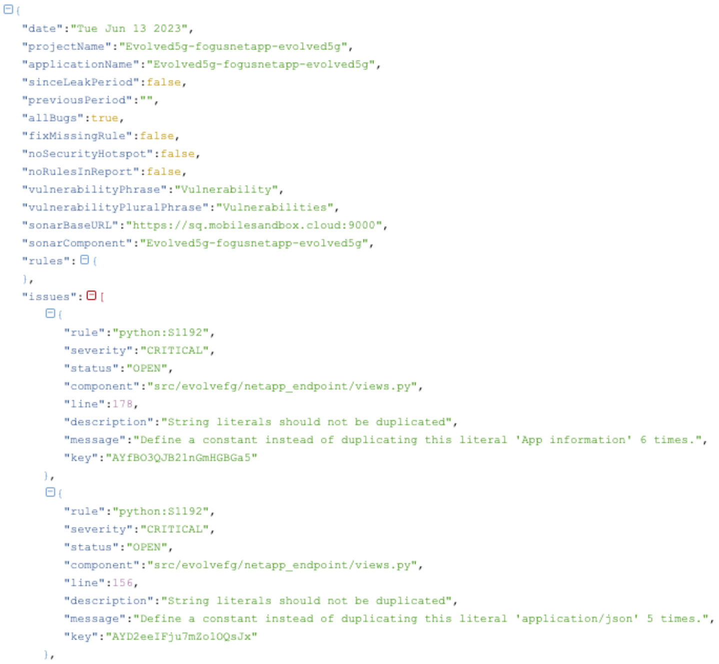


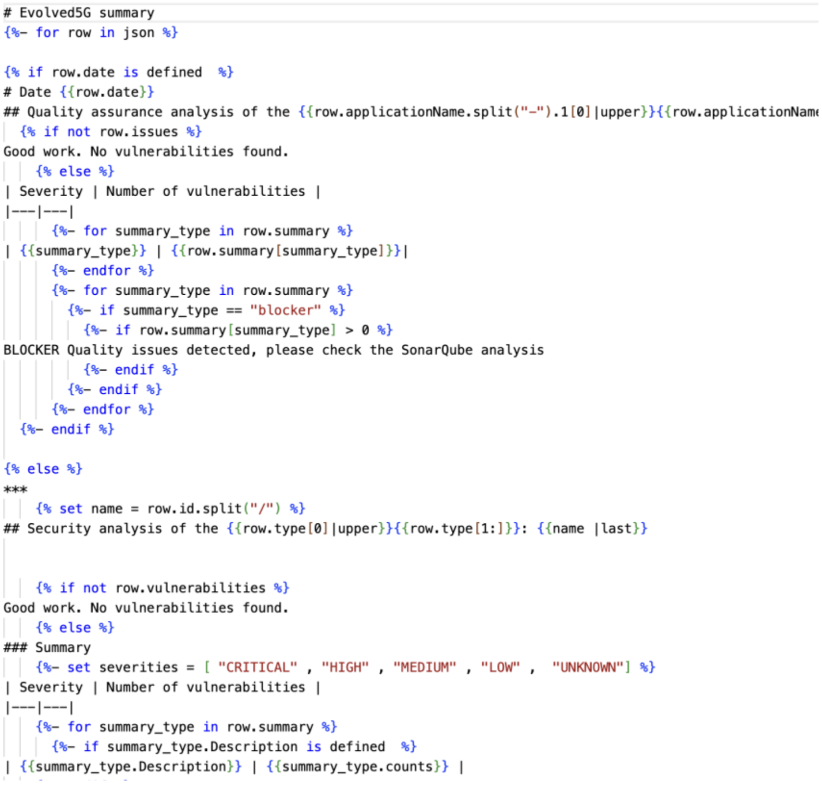
Figure 22: Tool chain to generate Certification PDF Report



```
⊟{
  "date":"Tue Jun 13 2023",
  "projectName":"Evolved5g-fogusnetapp-evolved5g",
  "applicationName":"Evolved5g-fogusnetapp-evolved5g",
  "sinceLeakPeriod":false,
  "previousPeriod":"",
  "allBugs":true,
  "fixMissingRule":false,
  "noSecurityHotspot":false,
  "noRulesInReport":false,
  "vulnerabilityPhrase":"Vulnerability",
  "vulnerabilityPluralPhrase":"Vulnerabilities",
  "sonarBaseURL":"https://sq.mobilesandbox.cloud:9000",
  "sonarComponent":"Evolved5g-fogusnetapp-evolved5g",
  "rules":⊟{
  },
  "issues":⊟[
     ⊟{
        "rule":"python:S1192",
        "severity":"CRITICAL",
        "status":"OPEN",
        "component":"src/evolvefg/netapp_endpoint/views.py",
        "line":178,
        "description":"String literals should not be duplicated",
        "message":"Define a constant instead of duplicating this literal 'App information' 6 times.",
        "key":"AYfBO3QJB21nGmHGBGa5"
     },
     ⊟{
        "rule":"python:S1192",
        "severity":"CRITICAL",
        "status":"OPEN",
        "component":"src/evolvefg/netapp_endpoint/views.py",
        "line":156,
        "description":"String literals should not be duplicated",
        "message":"Define a constant instead of duplicating this literal 'application/json' 5 times.",
        "key":"AYD2eeIFju7mZo1OQsJx"
     },
```

Figure 23: Fragment of JSON file generated for Sonarqube results

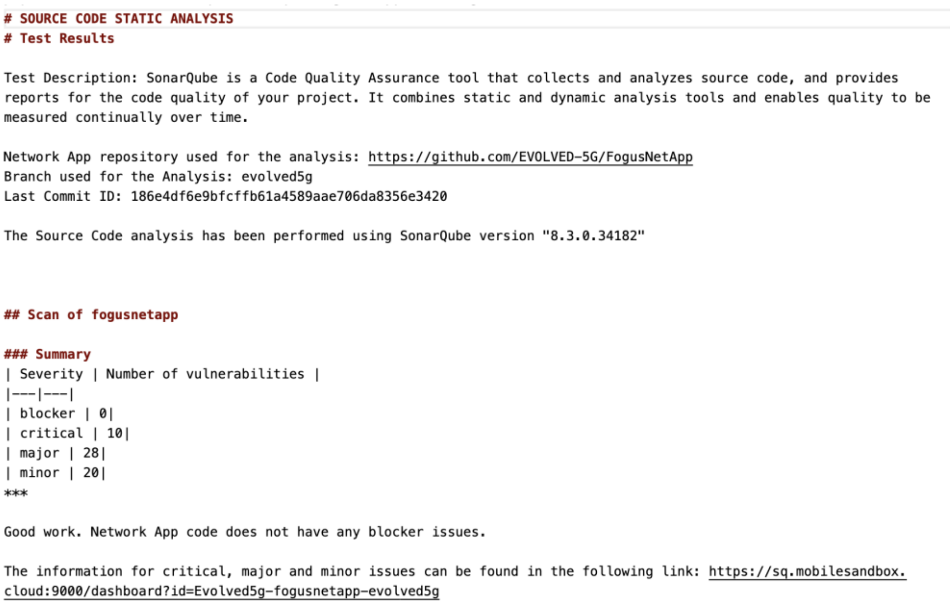
These JSON files are transformed into Markdown documents using JINJA templates. For example, for the JSON file from Figure 23, the following Jinja template generates the Markdown document for Sonarqube results:



```
# Evolved5G summary
{%- for row in json %}

{% if row.date is defined  %}
# Date {{row.date}}
## Quality assurance analysis of the {{row.applicationName.split("-").1[0]|upper}}{{row.applicationName
  {% if not row.issues %}
Good work. No vulnerabilities found.
    {% else %}
| Severity | Number of vulnerabilities |
|---|---|
      {%- for summary_type in row.summary %}
| {{summary_type}} | {{row.summary[summary_type]}}|
      {%- endfor %}
      {%- for summary_type in row.summary %}
        {%- if summary_type == "blocker" %}
          {%- if row.summary[summary_type] > 0 %}
BLOCKER Quality issues detected, please check the SonarQube analysis
          {%- endif %}
        {%- endif %}
      {%- endfor %}
  {%- endif %}

{% else %}
***
    {% set name = row.id.split("/") %}
## Security analysis of the {{row.type[0]|upper}}{{row.type[1:]}}: {{name |last}}


    {% if not row.vulnerabilities %}
Good work. No vulnerabilities found.
    {% else %}
### Summary
    {%- set severities = [ "CRITICAL" , "HIGH" , "MEDIUM" , "LOW" ,  "UNKNOWN"] %}
| Severity | Number of vulnerabilities |
|---|---|
    {%- for summary_type in row.summary %}
      {%- if summary_type.Description is defined  %}
| {{summary_type.Description}} | {{summary_type.counts}} |
```

Figure 24: Fragment of JINJA Template for Sonarqube JSON Files

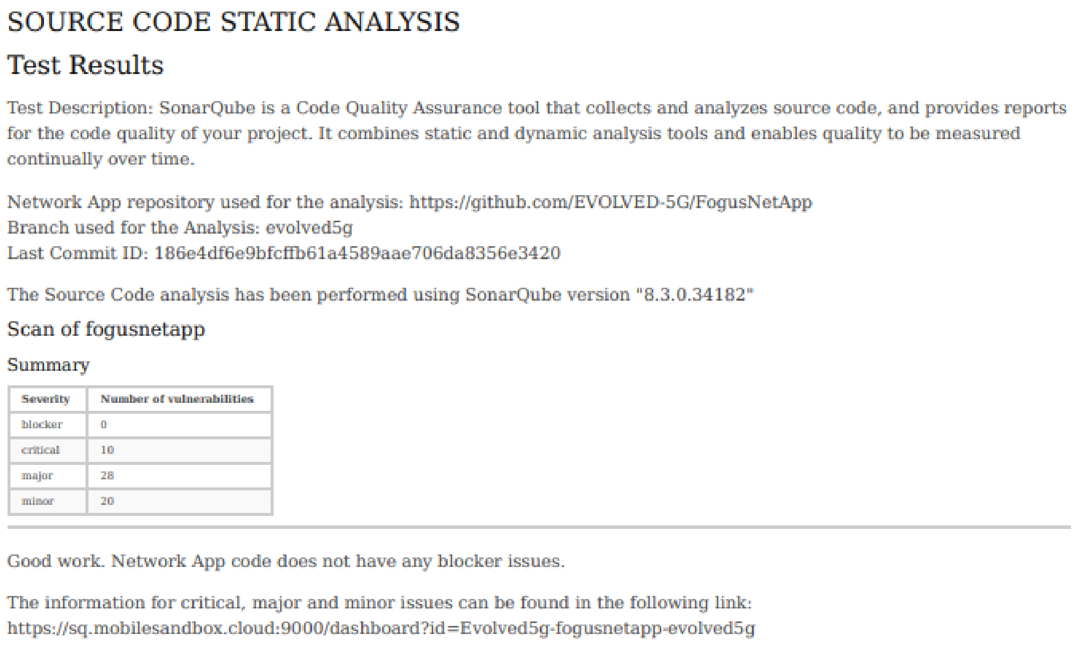
The result of this Jinja transformation is a Markdown document that shows the information to export into the PDF file of the Certification Report.



```
# SOURCE CODE STATIC ANALYSIS
# Test Results

Test Description: SonarQube is a Code Quality Assurance tool that collects and analyzes source code, and provides
reports for the code quality of your project. It combines static and dynamic analysis tools and enables quality to be
measured continually over time.

Network App repository used for the analysis: https://github.com/EVOLVED-5G/FogusNetApp
Branch used for the Analysis: evolved5g
Last Commit ID: 186e4df6e9bfcffb61a4589aae706da8356e3420

The Source Code analysis has been performed using SonarQube version "8.3.0.34182"



## Scan of fogusnetapp

### Summary
| Severity | Number of vulnerabilities |
|---|---|
| blocker | 0|
| critical | 10|
| major | 28|
| minor | 20|
***

Good work. Network App code does not have any blocker issues.

The information for critical, major and minor issues can be found in the following link: https://sq.mobilesandbox.
cloud:9000/dashboard?id=Evolved5g-fogusnetapp-evolved5g
```

Figure 25: Markdown Document for Sonarqube Test Results

Finally, this Markdown file is integrated into a PDF file using markdown-pdf tool [15].



SOURCE CODE STATIC ANALYSIS

Test Results

Test Description: SonarQube is a Code Quality Assurance tool that collects and analyzes source code, and provides reports for the code quality of your project. It combines static and dynamic analysis tools and enables quality to be measured continually over time.

Network App repository used for the analysis: https://github.com/EVOLVED-5G/FogusNetApp
Branch used for the Analysis: evolved5g
Last Commit ID: 186e4df6e9bfcffb61a4589aae706da8356e3420

The Source Code analysis has been performed using SonarQube version "8.3.0.34182"

Scan of fogusnetapp

Summary

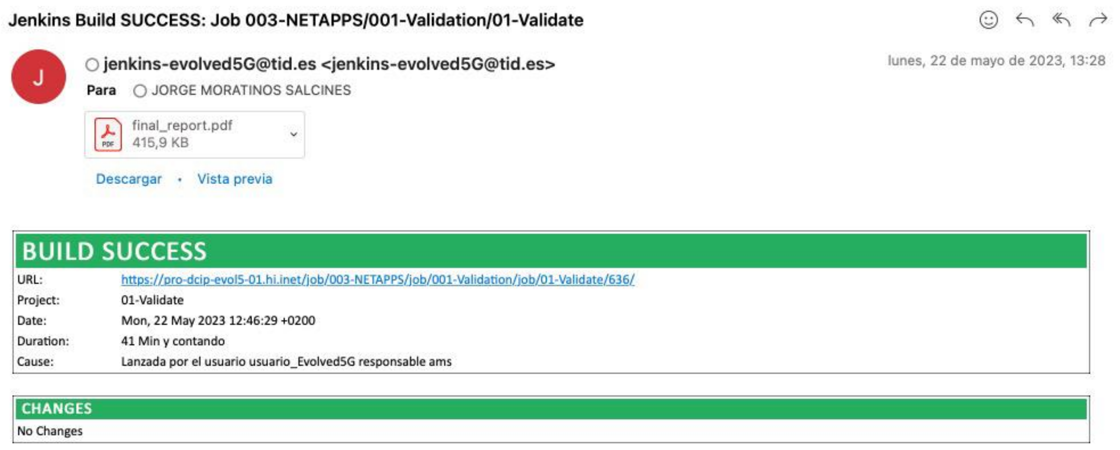
Severity	Number of vulnerabilities
blocker	0
critical	10
major	28
minor	20

Good work. Network App code does not have any blocker issues.

The information for critical, major and minor issues can be found in the following link:
https://sq.mobilesandbox.cloud:9000/dashboard?id=Evolved5g-fogusnetapp-evolved5g

Figure 26: PDF page for Sonarqube Test Results generated by markdown-pdf

When the Certification Process is completed, the Certification Report PDF is sent to the Developer via e-mail to the e-mail address configured when launching the Certification Pipeline.



Jenkins Build SUCCESS: Job 003-NETAPPS/001-Validation/01-Validate

J ○ jenkins-evolved5G@tid.es <jenkins-evolved5G@tid.es>
lunes, 22 de mayo de 2023, 13:28
Para ○ JORGE MORATINOS SALCINES

final_report.pdf
415,9 KB

BUILD SUCCESS

URL:	https://pro-dcip-evol5-01.hi.inet/job/003-NETAPPS/job/001-Validation/job/01-Validate/636/
Project:	01-Validate
Date:	Mon, 22 May 2023 12:46:29 +0200
Duration:	41 Min y contando
Cause:	Lanzada por el usuario usuario_Evolved5G responsable ams

CHANGES

No Changes

Figure 27: EMAIL sent to Network App Developers upon Successful Certification Result

Once the Developer gets the Certification Report with a Successful result, he or she can upload the Network App to the EVOLVED-5G Marketplace (using the Fingerprint Unique ID) that is described in next section.

5 Marketplace

5.1 Marketplace Overview/Intro

The EVOLVED-5G Marketplace is an online, cross-industry API Marketplace that enables developers, entrepreneurs and businesses to come together to create, discover and integrate services by consuming the Network Apps that have been created in the EVOLVED-5G ecosystem.

The Marketplace platform also provides support to its users by means of a forum. The forum allows Network App creators, Network App consumers and visitors to ask questions and exchange knowledge with other marketplace users and get support from marketplace administrators.

In the EVOLVED-5G context we have identified as Network App creators the SMEs or individual Network App developers that implement and release them in the Marketplace. Network App consumers are SMEs that want to use the released Network Apps in order to provide a solution to their end customers.
The Marketplace targets four primary profiles. These profiles represent an approximation of a segment of the marketplace users: “Visitors”, “Network App creators”, “Network App consumers” and "Marketplace Administrators".

- **“Visitors”** interact for the first time with the EVOLVED-5G ecosystem and are able to understand the EVOLVED-5G offering and its value propositions. They can use the marketplace to explore the product catalogue, view promotional material related with each Network App, request support through the marketplace’s forum or seek technical advice in order to participate in the marketplace as a Network App creator or as a Network App consumer.
- On the other hand, **“Network App creators”** use the Marketplace as the last step in the life cycle of their Network App development. By registering, they can connect their certified Network App with the marketplace, upload marketing and branding material, use wizards to provide CPQ (Configure, Price, Quote) functionality and provide technical support for their users through online tutorials or the forum.
- **“Network App consumers”** are able to use the marketplace to “purchase” Network Apps, gaining access to their APIs and utilize them. A smart contract is responsible for storing the digital print of the purchase in Ethereum's distributed database. This blockchain transaction acts as a “proof of purchase” and will publicly exist even if the marketplace is no longer in place.
- Finally, **“Marketplace Administrators”** are responsible for all the administration tasks of the platform.

The Marketplace has been developed taking into account best practices from the Software Development Life Cycle (SDLC [16]). SDLC is a set of steps used to create software applications. These steps divide the development process into tasks that can then be assigned, completed, and measured. In the upcoming sections we describe the design phase of the Marketplace (including planning, requirements collection, design and prototyping), the development phase of the Marketplace (including architecture definition and actual implementation) and the deployment phase of the Marketplace (including the dockerization of all the components) to be available in the EVOLVED-5G OpenShift platform. Finally, we conclude with some screenshots that demonstrate how a Network App can be released to the Marketplace.

5.2 Marketplace Design

The Marketplace design phase included 2 phases: 1) Finalization of the requirements, and 2) Mockups creation.

5.2.1 Phase 1 - Finalization of the requirements

The Marketplace is a complex web application with the goal to support a community of different users (I.e., profiles as described in the previous section). During the requirements collection phase, the following steps have been finalized:

a) Definition of the platform roles: the different user profiles that the platform will support as described in Table 1.

Table 1: List of Platform Roles

#	*Role name*	*Description*
1	Visitor	A user that visits Marketplace landing pages. (S)he can quickly understand how the Marketplace works and its value propositions
2	Network App creators (SMEs / Developers)	A Network App creator can register to the marketplace in order to: - manage the Network App that he implemented - release the Network App to the marketplace (make it public) - set pricing schemes for the Network App
3	Network App consumers	A Network App consumer is an entity that uses the marketplace to buy one or more Network Apps
4	Marketplace administrator	A Marketplace administrator has access to key performance indicators (KPIs) of the platform like: number of submitted Network Apps number of purchased Network Apps

b) Definition of high-level functional areas: that is high level categories of functionality that should exist in order to support the Marketplace's scenarios of usage

Table 2: List of Functional Areas

#	*Component / Functional Area*	*Description*
1	Landing pages	The goal of these pages is for a user to easily understand what the Marketplace offers and how it works, as well as to start interacting with the platform.
2	Authentication / Authorization	Allow the user to login / register to the platform
3	Public Product catalogue	Allow the user to search/ view Network Apps.
4	Network Apps management	Allow the user to manage (create/edit/delete) Network Apps
5	Network Apps purchases	Allow the user to purchase Network Apps and select from different available pricing packages
6	Email notifications	The platform should be able to notify users about events
8	Dashboards	Allow: - Network App creators to perform management and monitoring of Network Apps - Network App consumers to monitor Network Apps status- Marketplace administrators to view high level KPIs of the platform
9	Forum integration	The platform should provide support to its users by a forum. The forum will allow Network App creators, Network App consumers and Visitors ask questions and get support by other marketplace users and marketplace administrators.
10	Blockchain integration	The platform should store a digital receipt of a Network App purchase to a blockchain network. The user will have access to this digital receipt in the Ethereum network.

c) A set of user stories, for each of the roles above-mentioned. A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value for each of the aforementioned roles. For example, “a visitor should be able to search for Network Apps”, “a visitor should be able to view details of a specific Network App and understand its value propositions” etc.

d) A user interface flow diagram, which illustrates the interactions that users will have with the application. An example of the initial interface flow diagram that was created during this phase is provided below:

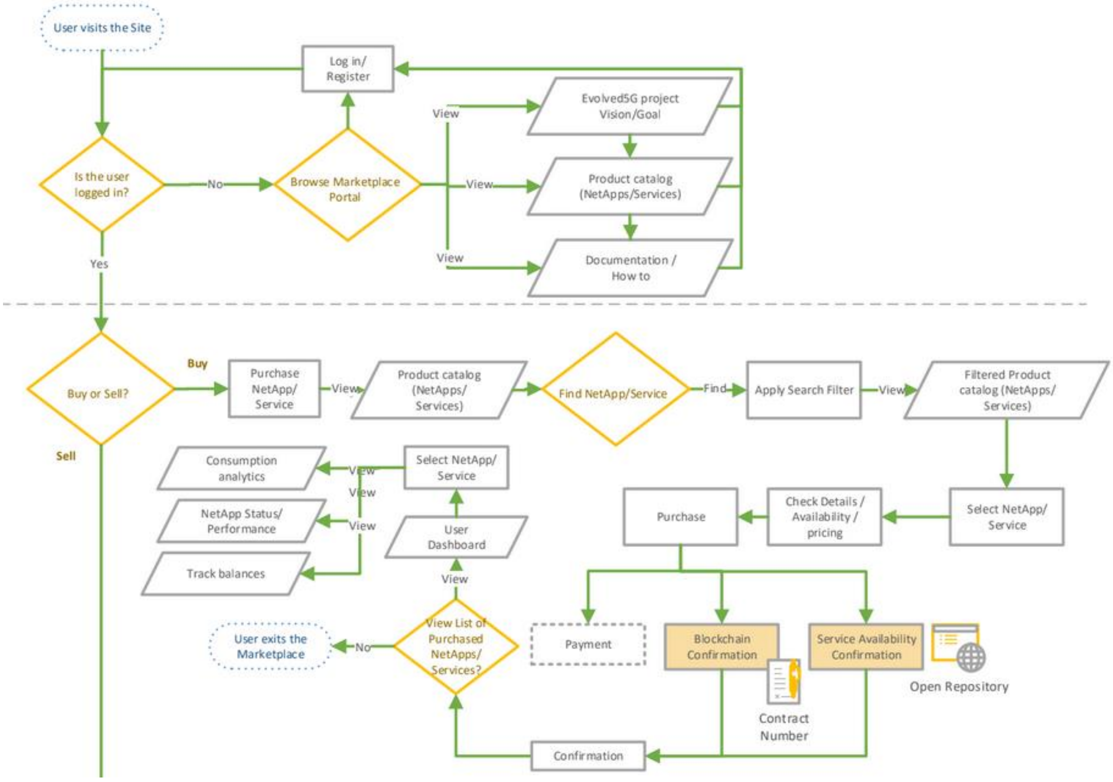


Figure 28: Evolved-5G user interface flow chart that drove the creation of the mockups

As depicted in the above figure when a user visits the EVOLVED-5G Marketplace portal and before registering or logging in he/she can browse through the contents of the portal and get some information about the EVOLVED-5G project vision and goal, read the Technical Documentation and How to’ s and even take a look in the product catalog to get some idea of the Network Apps already published in the Marketplace. After a visitor registers the Marketplace and logs-in he/she can selectively buy or sell a Network App.

5.2.2 Phase 2 - Mockups creation

Based on the requirements listed in the previous section, a set of mockups has been created. A mockup is a static design of a web page or application that features many of its final design elements but is not functional. The mockups have been created with the help of a popular online prototyping tool Adobe XD [17]. Adobe XD helps to craft prototypes that look and feel like the real thing, so that the user can communicate the design vision and maintain alignment across teams efficiently. The creation of the mockups allowed feedback collection about the Marketplace, early in the process, and facilitated fast improvements, since changing a UI Component in a static mockup is much more efficient compared to making the change in the implemented version of each screen.

The following figure provides examples of the mockups that have been created in the online prototyping tool. A snapshot of all the mockups that have been created using Adobe XD which can also be accessed in the following url: https://xd.adobe.com/view/6cffe34c-ceb6-4ec8-9d53-8d7bc36b3bab-39ac/grid

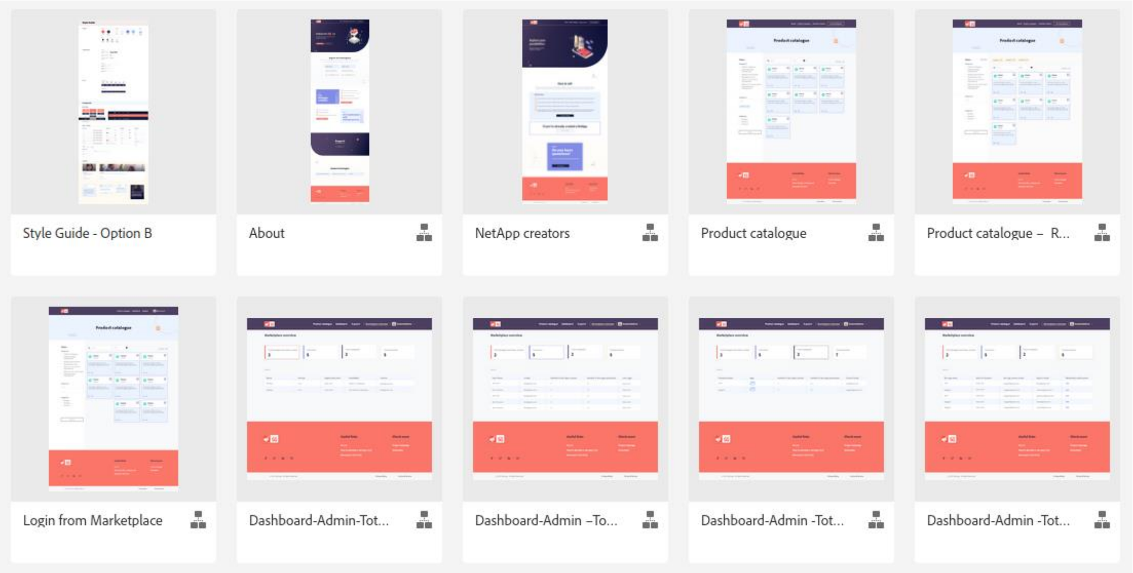
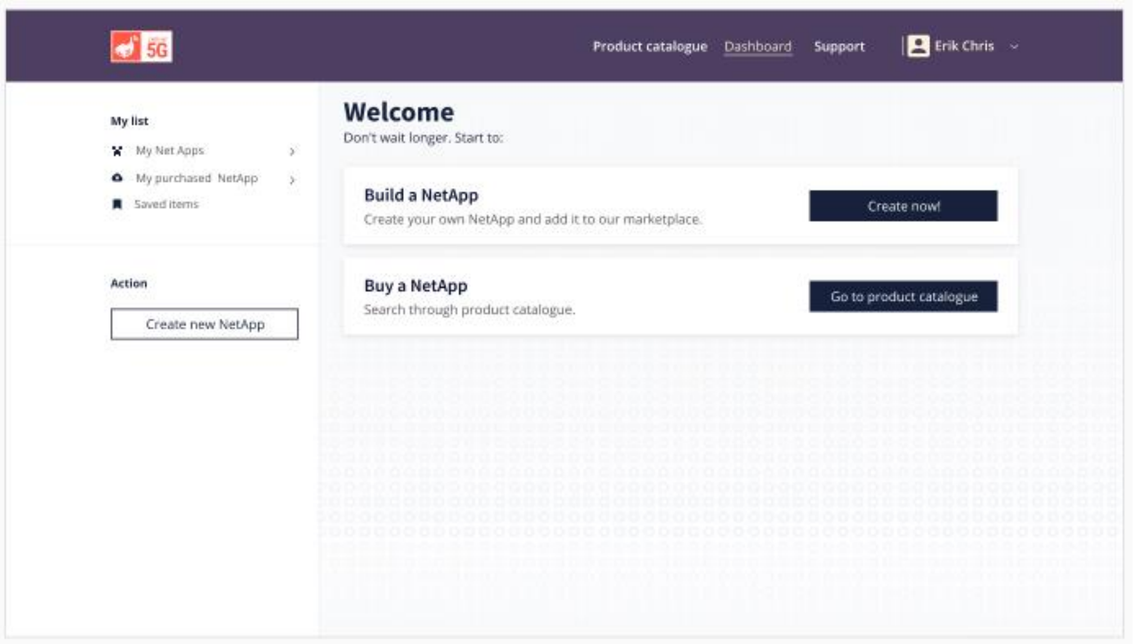


Figure 29: Examples of mockups that have been created in the online prototyping tool



My list

- My Net Apps
- My purchased NetApp
- Saved items

Action

Create new NetApp

Welcome

Don't wait longer. Start to:

Build a NetApp
Create your own NetApp and add it to our marketplace.
Create now!

Buy a NetApp
Search through product catalogue.
Go to product catalogue

Figure 30: Example of mockup of the Welcome

5.3 Marketplace Implementation

The functional areas defined in the previous development phases along with the user stories and the mockups allowed the technical team to define the overall architecture of the Marketplace and start the implementation.
In the following diagram the final high-level architecture of the Marketplace is shown.

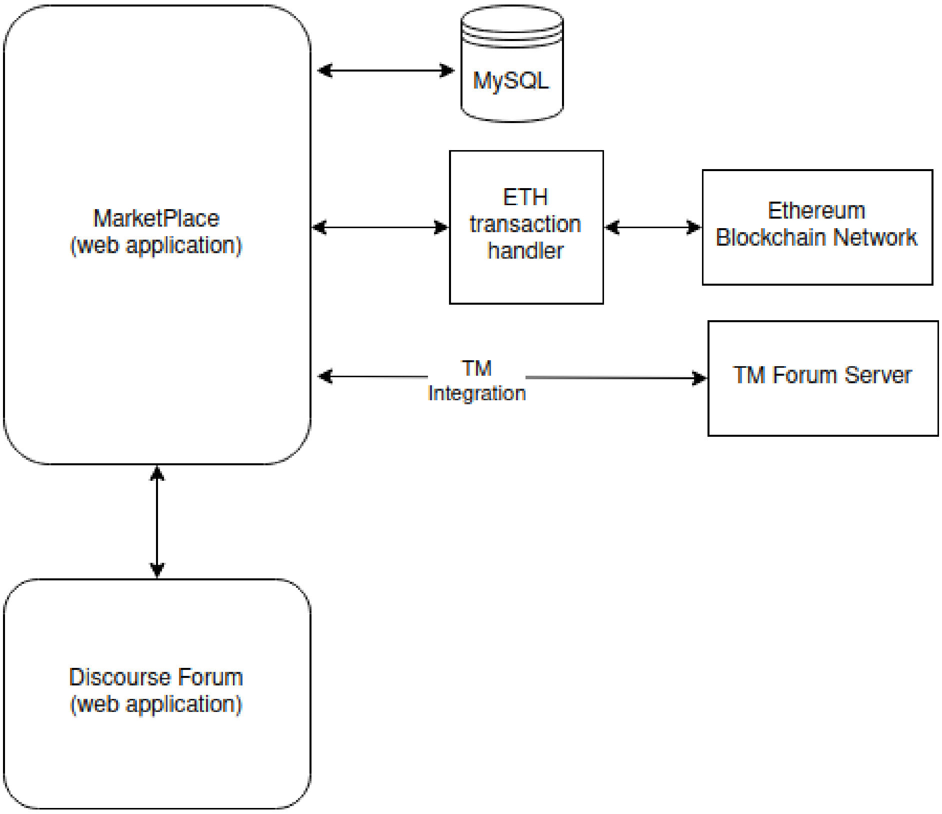
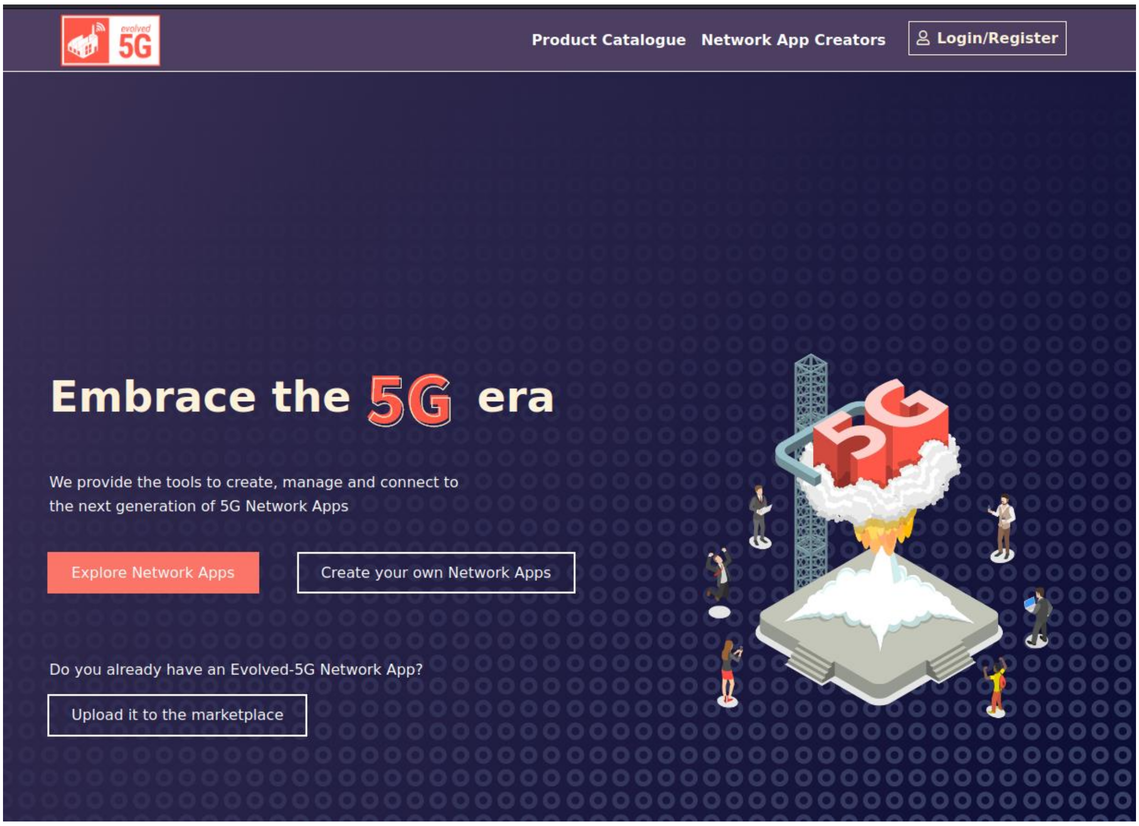


Figure 31: High level architecture of the Marketplace

5.3.1 Marketplace web application

The Marketplace web application supports all the user scenarios for Visitors, Network App creators, Network App consumers and Marketplace Administrators. Additionally, it acts as an entry point to the Marketplace's forum, where users can find support on various EVOLVED-5G topics. It is developed using Laravel [18], one of the most popular frameworks for building web applications, JavaScript Frameworks, VueJs [19] and CSS frameworks (Bootstrap [20] and Sass [21]). The storage layer was implemented in MySQL [22], a widely used relational database management system (RDBMS).



Embrace the 5G era

We provide the tools to create, manage and connect to the next generation of 5G Network Apps

Do you already have an Evolved-5G Network App?

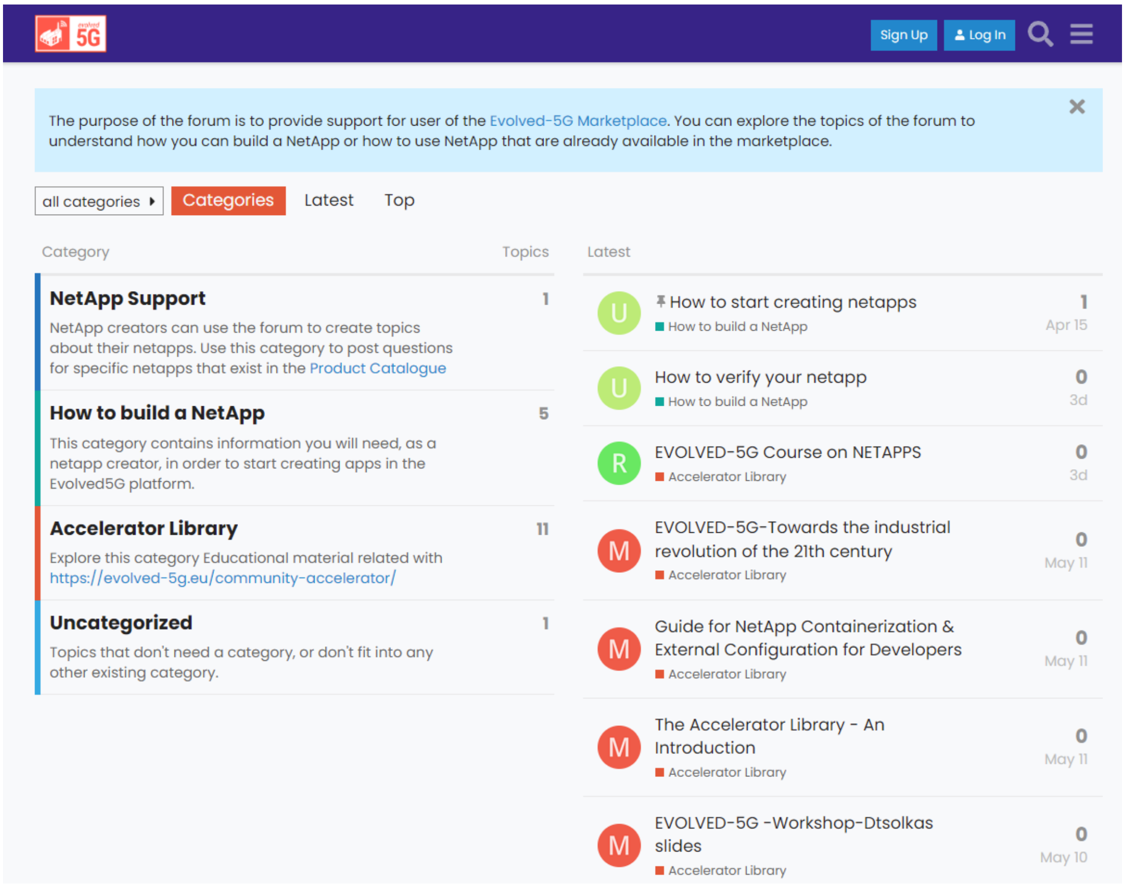
Figure 32: Marketplace entry page

The Marketplace backend implements the integration with other Marketplace components like a) the TM forum server and b) the Ethereum transaction handler. Both are explained later in the chapter.

The relevant repository can be found at https://github.com/EVOLVED-5G/marketplace, and the production environment can be found at https://marketplace.evolved-5g.eu/

5.3.2 Marketplace forum

The Marketplace forum has been implemented using Discourse [23], an open-source, powerful platform for communities. A Discourse server has been initialized and its theme has been configured in order to match the branding of the EVOLVED-5G communities.



The purpose of the forum is to provide support for user of the Evolved-5G Marketplace. You can explore the topics of the forum to understand how you can build a NetApp or how to use NetApp that are already available in the marketplace.

Category	Topics
NetApp Support NetApp creators can use the forum to create topics about their netapps. Use this category to post questions for specific netapps that exist in the Product Catalogue	1
How to build a NetApp This category contains information you will need, as a netapp creator, in order to start creating apps in the Evolved5G platform.	5
Accelerator Library Explore this category Educational material related with https://evolved-5g.eu/community-accelerator/	11
Uncategorized Topics that don't need a category, or don't fit into any other existing category.	1

Latest		
U	How to start creating netapps How to build a NetApp	1 Apr 15
U	How to verify your netapp How to build a NetApp	0 3d
R	EVOLVED-5G Course on NETAPPS Accelerator Library	0 3d
M	EVOLVED-5G-Towards the industrial revolution of the 21th century Accelerator Library	0 May 11
M	Guide for NetApp Containerization & External Configuration for Developers Accelerator Library	0 May 11
M	The Accelerator Library - An Introduction Accelerator Library	0 May 11
M	EVOLVED-5G -Workshop-Dtsolkas slides Accelerator Library	0 May 10

Figure 33: Forum and accelerator entry page

A Network App creator can use the forum in order to create topics about their Network Apps and to provide relevant support. Visitors of the EVOLVED-5G Marketplace can use the forum to understand how to build a Network App and to get relevant support. Finally, educational material can be found in the forum related with the "Community Accelerator"of EVOLVED-5G [24].
The online environment of the forum is available at https://forum.evolved-5g.eu/ and is integrated with the web application, that is, various screens in the web application point to relevant topics in the forum (e.g., when a user seeks for support for a specific Network App, or when a user needs to find out how to build a Network App).

5.3.3 ETH Transaction Sender / Blockchain integration

The blockchain integration in the Marketplace was driven by the need to create a digital signature of a purchase, when a Network App consumer purchases a Network App, and to store it in a public distributed database. This ledger acts as a "proof of purchase" and will exist publicly even if the Marketplace is no longer in place. The Ethereum Blockchain Network covers this need.

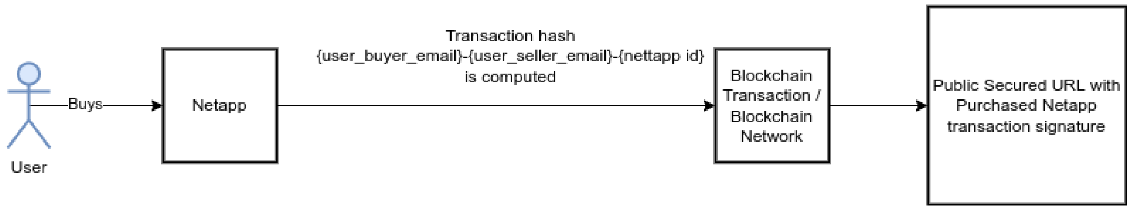


Figure 34: High level overview of the blockchain integration

Every time a user buys a Network App via the Marketplace, a digital signature (hash string) is created. For example, the user with email "*buyer@test.com*" buys a Network

App with id "*123*", that was created by the user "seller@test.com". The digital signature is a hashed version of the string: "buyer@test.com-buyer@*test.com-123". This signature, after being hashed, is posted to the Ethereum Network.

5.3.4 How it works

An Ethereum Wallet (which belongs to Evolved-5G), is responsible for creating a new Blockchain Transaction. This Transaction has the digital signature of the purchase, as its Input Data

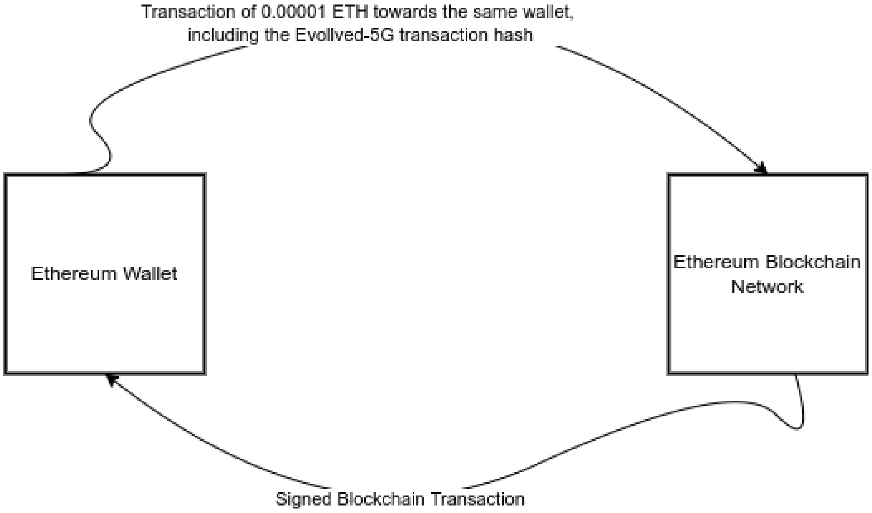
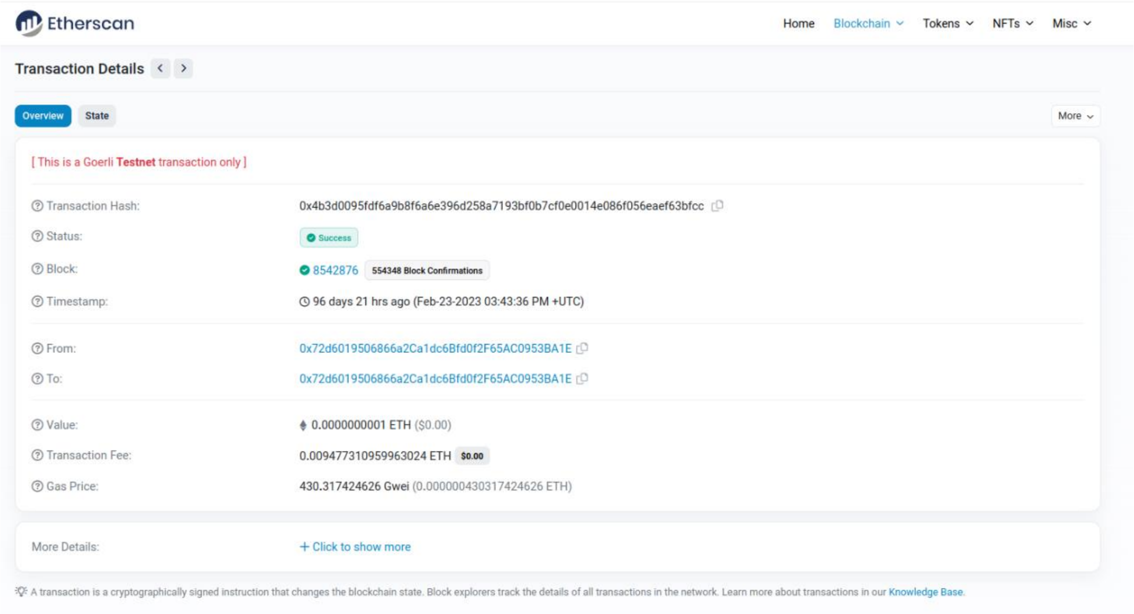

Transaction of 0.00001 ETH towards the same wallet,
including the Evollved-5G transaction hash
Ethereum Wallet
Ethereum Blockchain Network
Signed Blockchain Transaction

Figure 35: Blockchain transaction

The transaction is posted to the Ethereum Blockchain Network and is visible to all, containing also the digital signature: For example, in the image below, one can see a transaction from the ETH testing blockchain network https://goerli.etherscan.io/tx/0x4b3d0095fdf6a9b8f6a6e396d258a7193bf0b7cf0e0014e086f056eaef63bfcc
If one clicks the "click to see more" field, they can view the "Input Data" field that contains the hashed version of the digital signature.



[This is a Goerli **Testnet** transaction only]

Transaction Hash: 0x4b3d0095fdf6a9b8f6a6e396d258a7193bf0b7cf0e0014e086f056eaef63bfcc

Status: Success

Block: 8542876 554348 Block Confirmations

Timestamp: 96 days 21 hrs ago (Feb-23-2023 03:43:36 PM +UTC)

From: 0x72d6019506866a2Ca1dc6Bfd0f2F65AC0953BA1E

To: 0x72d6019506866a2Ca1dc6Bfd0f2F65AC0953BA1E

Value: 0.0000000001 ETH ($0.00)

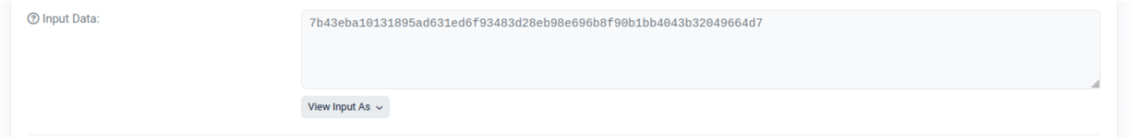
Transaction Fee: 0.009477310959963024 ETH $0.00

Gas Price: 430.317424626 Gwei (0.000000430317424626 ETH)

More Details: + Click to show more

A transaction is a cryptographically signed instruction that changes the blockchain state. Block explorers track the details of all transactions in the network. Learn more about transactions in our Knowledge Base.

Figure 36: Digital signature of the Evolved-5G purchase transaction



Input Data:

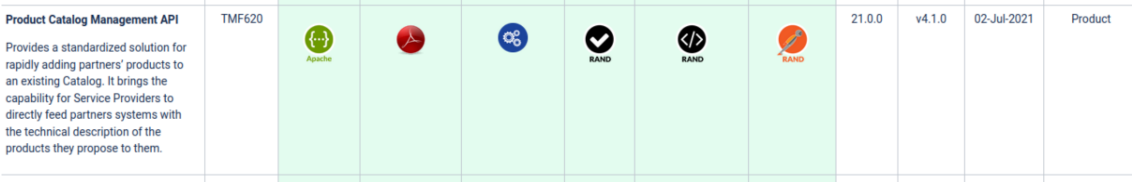
```
7b43eba10131895ad631ed6f93483d28eb98e696b8f90b1bb4043b32049664d7
```

Figure 37: Input Data in the Etherscan and the related hashed signature

The Blockchain transaction consists of a transfer of Ethereum (0.00000001 ETH => 0.00001 EUR), using the same wallet as origin and destination. These wallets belong to and are controlled by the EVOLVED-5G Marketplace.
At the end of the procedure, the wallet pays only the transaction costs, for creating and storing the transaction on the Blockchain Network. The pilot currently runs on the Goerli Test Ethereum Network, which is free. For the Smart Contract implementation, infura.io has been used, which is a 3rd party service that allows free requests to the Ethereum network for up to 100K requests/day.

5.3.5 TM forum integration

The purpose of the TM forum integration is to pilot how an integration can be achieved with TM Forum Open APIs. For this reason, the Product Catalogue Management API, TMF620-API [25] has been integrated to the Marketplace, which provides a standardized solution for adding products to a catalogue.



Product Catalog Management API Provides a standardized solution for rapidly adding partners' products to an existing Catalog. It brings the capability for Service Providers to directly feed partners systems with the technical description of the products they propose to them.	TMF620	Apache			RAND	RAND	RAND	21.0.0	v4.1.0	02-Jul-2021	Product

Figure 38: TMF620 API - Product Catalogue Management API

A TM Forum server [26] has been initialized and the Marketplace backend utilizes the forum server in order to a) store the categories of the Network Apps, and b) store pricing info about the Network Apps.

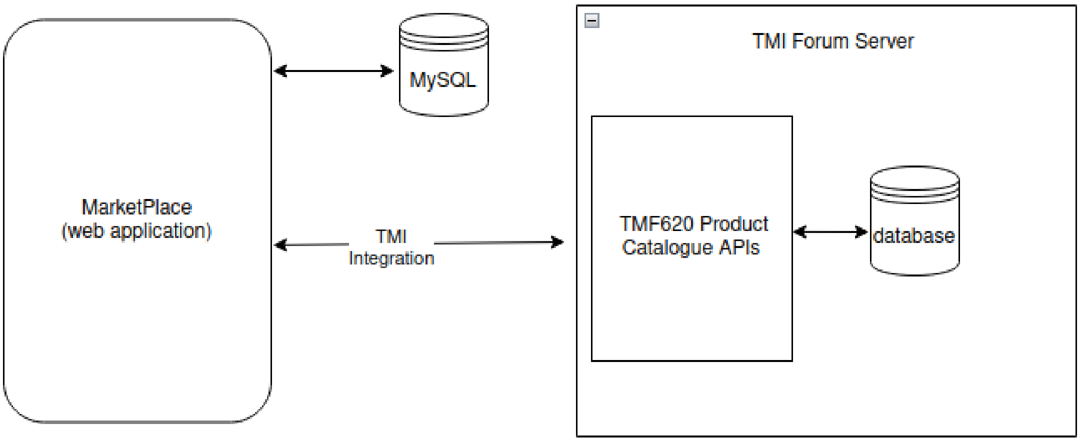
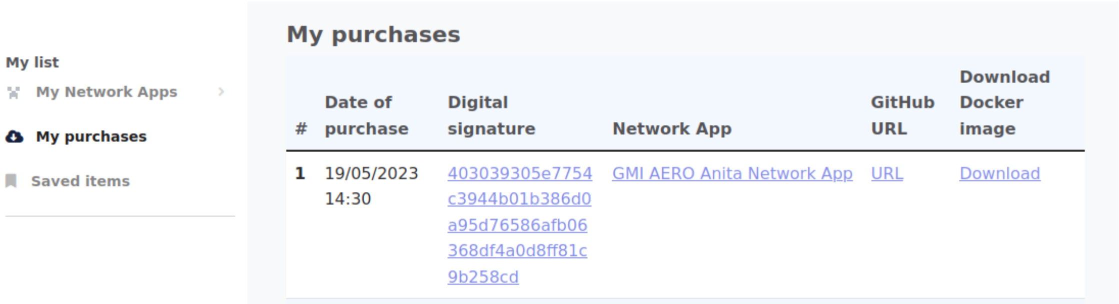


Figure 39: High level architecture of the TM Forum integration

5.4 Marketplace and Open Repository Integration

The EVOLVED-5G Marketplace is the final destination of a Network App, for which the development lifecycle has ended. The Marketplace exposes a dockerized image of the Network App ensuring also that it has previously been certified. A Marketplace user is able to access this docker images, through the "My purchases section" by clicking the option "Download Docker Image" as demonstrated in the figure below:



My purchases

#	Date of purchase	Digital signature	Network App	GitHub URL	Download Docker image
1	19/05/2023 14:30	403039305e7754c3944b01b386d0a95d76586afb06368df4a0d8ff81c9b258cd	GMI AERO Anita Network App	URL	Download

Figure 40: Users are able to download the Docker Images that they have purchased

The distribution of a Network App through the Marketplace is realized through the implementation of the EVOLVED-5G Open Repository. The EVOLVED-5G Open Repository is based on the JFrog platform. JFrog facilitates the creation of software repositories hosting artifacts of various forms, including files, packages, containers etc. Thus, the integration of the Open Repository with the EVOLVED-5G framework is designed in two points: (i) certification process – Open Repository and (ii) Open Repository – Marketplace.

As described in Chapter 4, the certification process ends up with a certified Network App being uploaded to the Open Repository, in the specific catalogue of certified artifacts. At the same time, a certification report is produced by the process, which is also uploaded to the Open Repository. A Fingerprint unique ID of the Network App produced by hashing the certified Network App image is shared in the certification report. This way, a detailed certification report is ensured to refer to a specific Network App, and in particular to a specific version of the Network App. Any update of the Network App requires a fresh certification process that yields a new certification report. The diagram depicted in the following figure illustrates the mapping process of the Network App with its certificate.

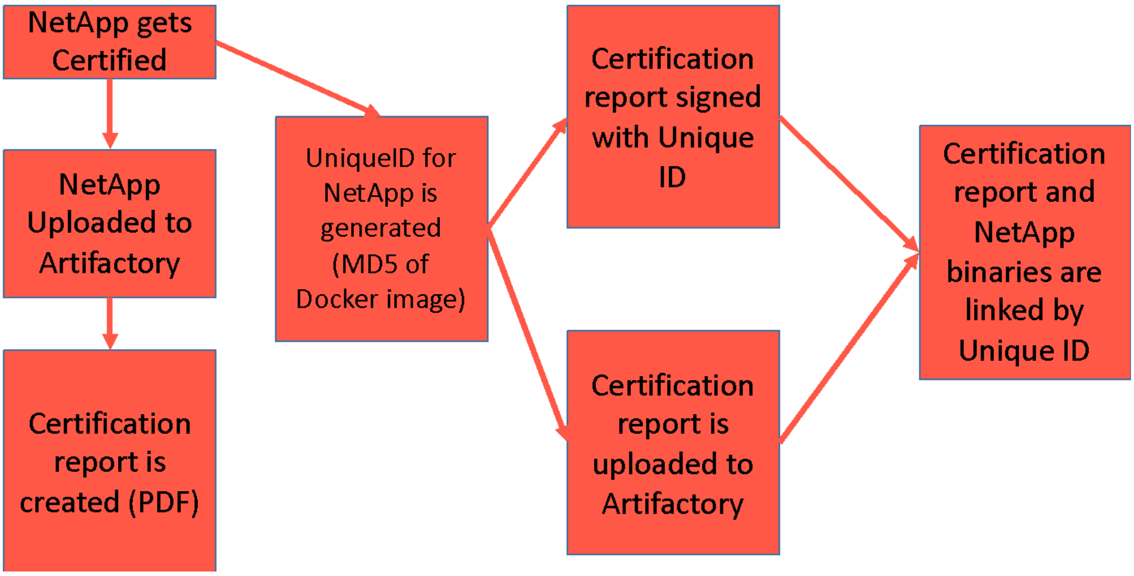

NetApp gets Certified
NetApp Uploaded to Artifactory
Certification report is created (PDF)
UniqueID for NetApp is generated (MD5 of Docker image)
Certification report signed with Unique ID
Certification report is uploaded to Artifactory
Certification report and NetApp binaries are linked by Unique ID

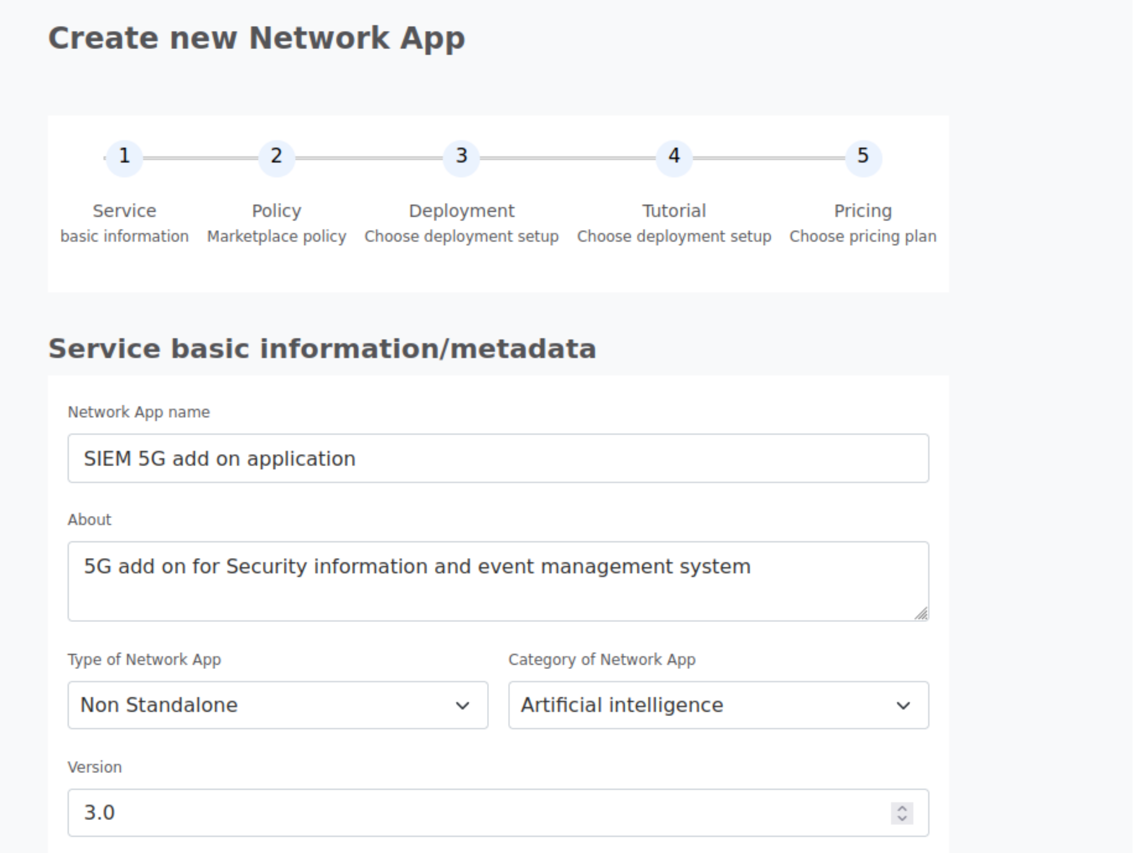
Figure 41: Mapping a Network App with its certificate

The integration of the Open Repository with the Marketplace is illustrated in the above Figure 41, Mapping a Network App with its certificate. The product catalogue of the Marketplace lists all the Network Apps that have been certified and uploaded to the Marketplace. Each Network App is presented with additional descriptive information along with the relevant certification report. A Network App in the product catalogue is linked with the Open Repository with a URL corresponding to the Network App image. Both are provided by the Network App developer during the Network App creation wizard execution in the Marketplace platform.

5.4.1 Implementation

In order to support the Open Repository integration, the “Network App creation” wizard requires the following information to be entered by the Network App developer:

a) The Network App version, as depicted in the Figure below, during the first step of the wizard:



Create new Network App

Service basic information/metadata

Network App name: SIEM 5G add on application

About: 5G add on for Security information and event management system

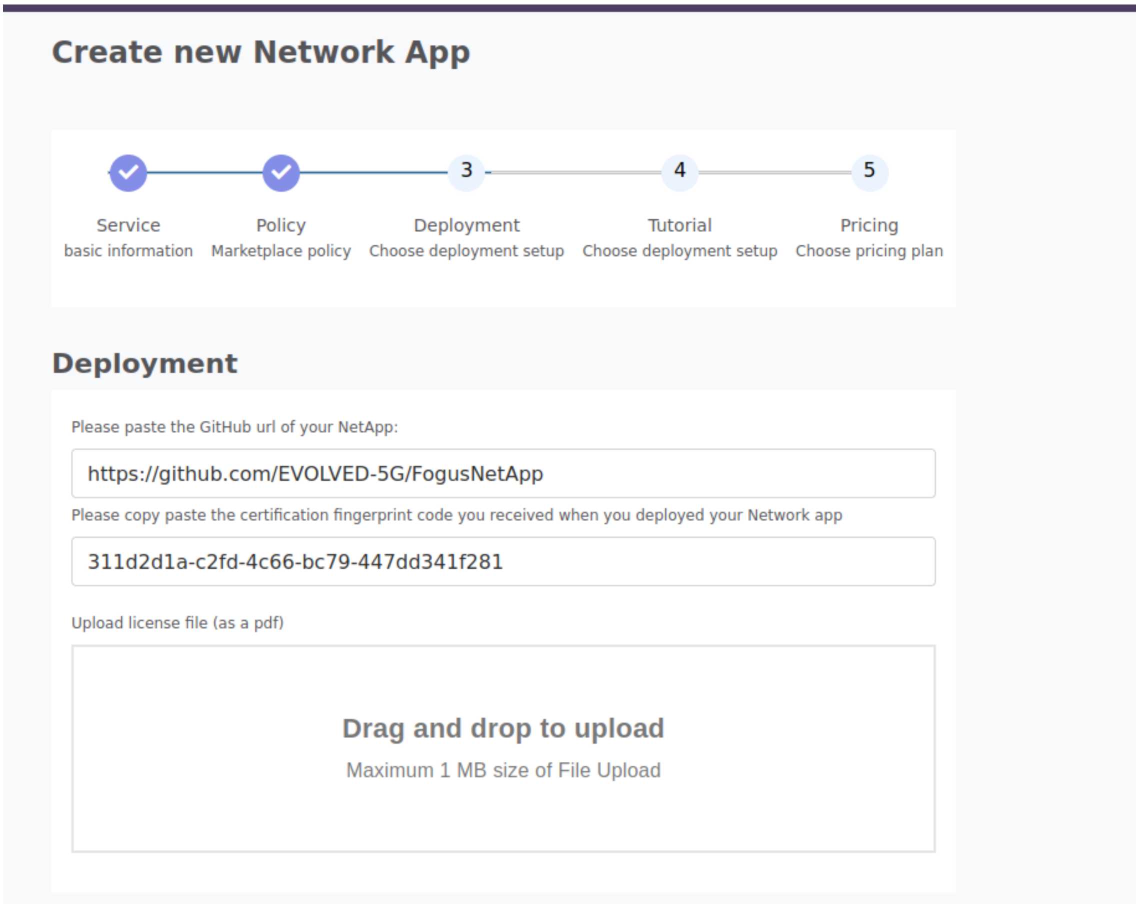
Type of Network App: Non Standalone

Category of Network App: Artificial intelligence

Version: 3.0

Figure 42: Version of the deployed Network App

b) The GitHub URL of the Network App.
c) The Unique ID (fingerprint code) the developer has received from the Certification report.



Create new Network App

Deployment

Please paste the GitHub url of your NetApp:

https://github.com/EVOLVED-5G/FogusNetApp

Please copy paste the certification fingerprint code you received when you deployed your Network app

311d2d1a-c2fd-4c66-bc79-447dd341f281

Upload license file (as a pdf)

Figure 43: A Network App developer provides the GitHub url of the Network app along with the Unique ID (fingerpint code) (s)he has received from the certification report

During this step the Marketplace will perform a request to the Open Repository in the background in order to make sure that the provided Fingerprint Unique ID matches the version of the deployed docker image.

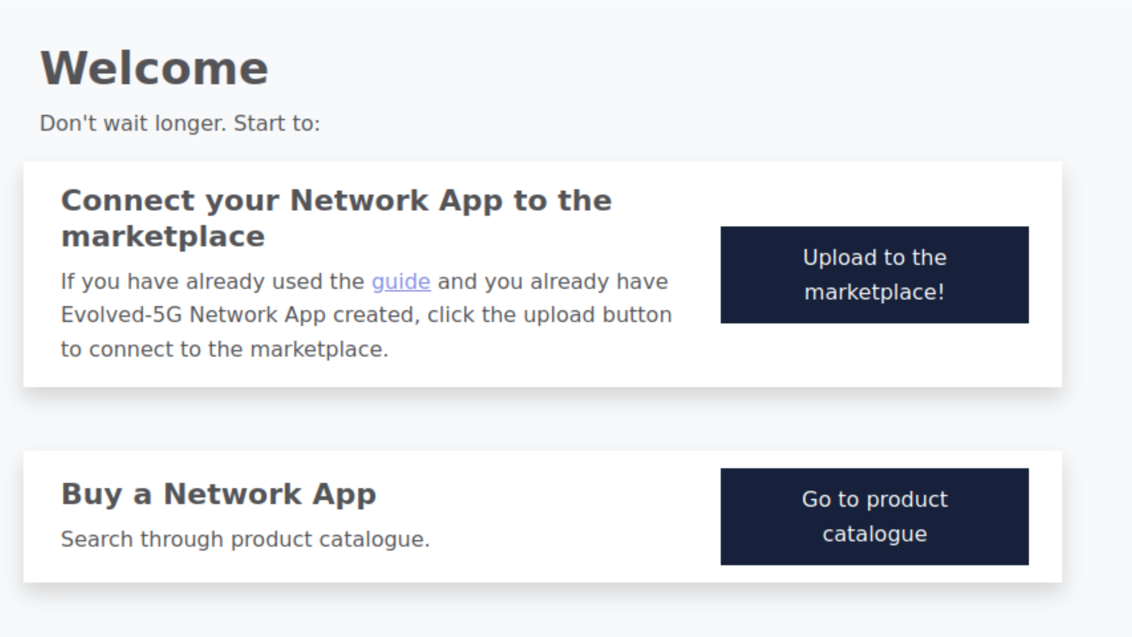
5.5 MARKETPLACE DEPLOYMENT

The Marketplace is deployed in the OpenShift platform. For this purpose, all Marketplace components have been dockerized:

a) The web application (related repo: https://github.com/EVOLVED-5G/marketplace) available at https://marketplace.evolved-5g.eu/
b) The blockchain component responsible for sending transactions to the Ethereum network (related repo: https://github.com/EVOLVED-5G/marketplace-blockchain-integration)
c) The TM forum server that contains the TMF60-API, Product Catalogue Management API endpoints (related repo: https://github.com/EVOLVED-5G/marketplace-tmf620-api
d) The Discourse Forum (related repo: https://github.com/discourse/discourse_docker) available at https://forum.evolved-5g.eu/

5.6 Network App Release to Marketplace

"Network App creators" use the Marketplace as the last step in the life cycle of their Network App. Once registered to the platform, they can initialize a wizard to release the Network App to the Marketplace.



Welcome

Don't wait longer. Start to:

Connect your Network App to the marketplace

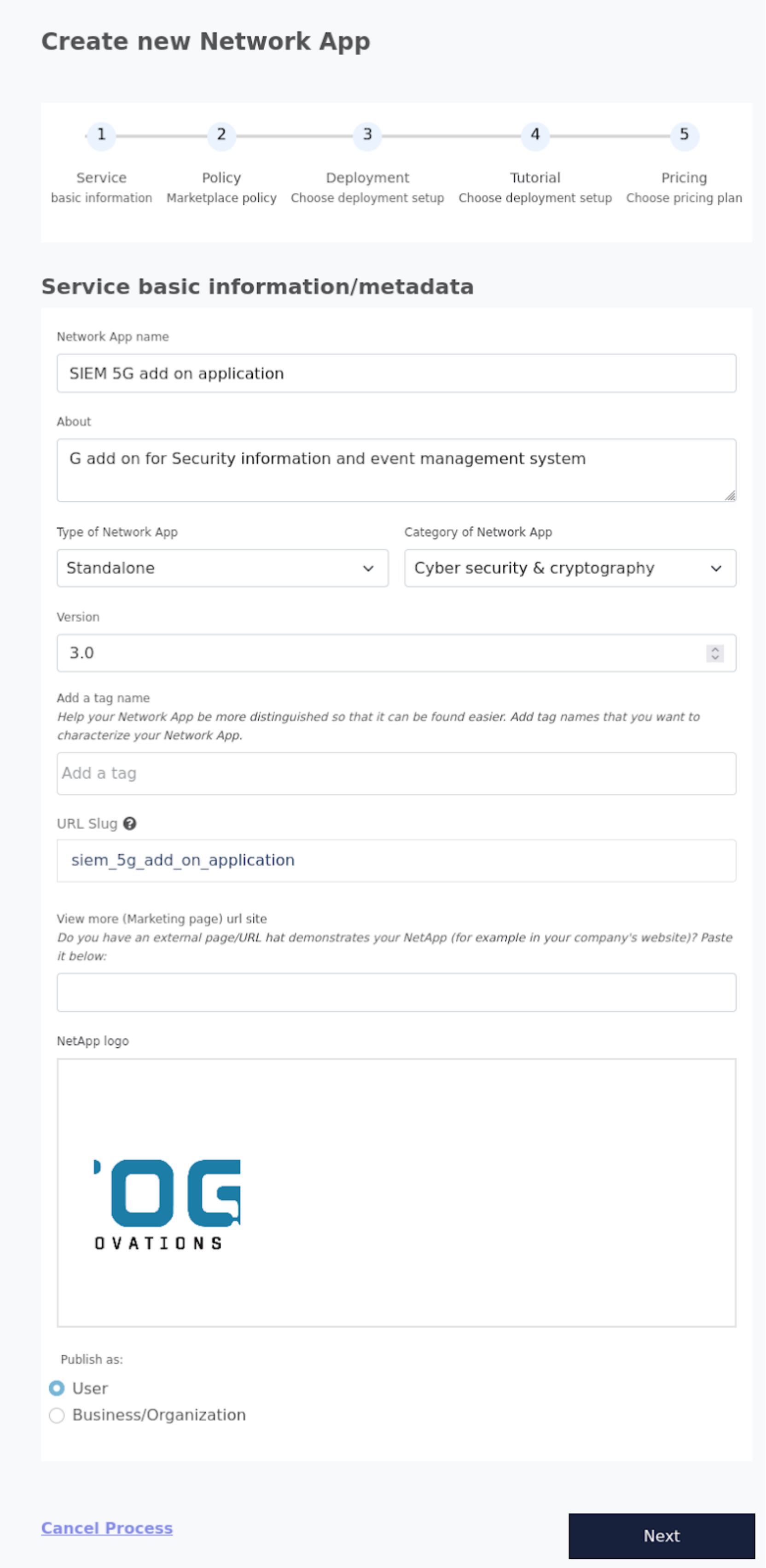
If you have already used the guide and you already have Evolved-5G Network App created, click the upload button to connect to the marketplace.

Buy a Network App

Search through product catalogue.

Figure 44: Welcome screen - The Network App creator is invited to start the wizard

During the first step, the Network App's basic details are provided, e.g., the Network App name, a short description, the category of the Network App etc.



Create new Network App

1 Service basic information
2 Policy Marketplace policy
3 Deployment Choose deployment setup
4 Tutorial Choose deployment setup
5 Pricing Choose pricing plan

Service basic information/metadata

Network App name: SIEM 5G add on application

About: G add on for Security information and event management system

Type of Network App: Standalone

Category of Network App: Cyber security & cryptography

Version: 3.0

Add a tag name
Help your Network App be more distinguished so that it can be found easier. Add tag names that you want to characterize your Network App.

Add a tag

URL Slug: siem_5g_add_on_application

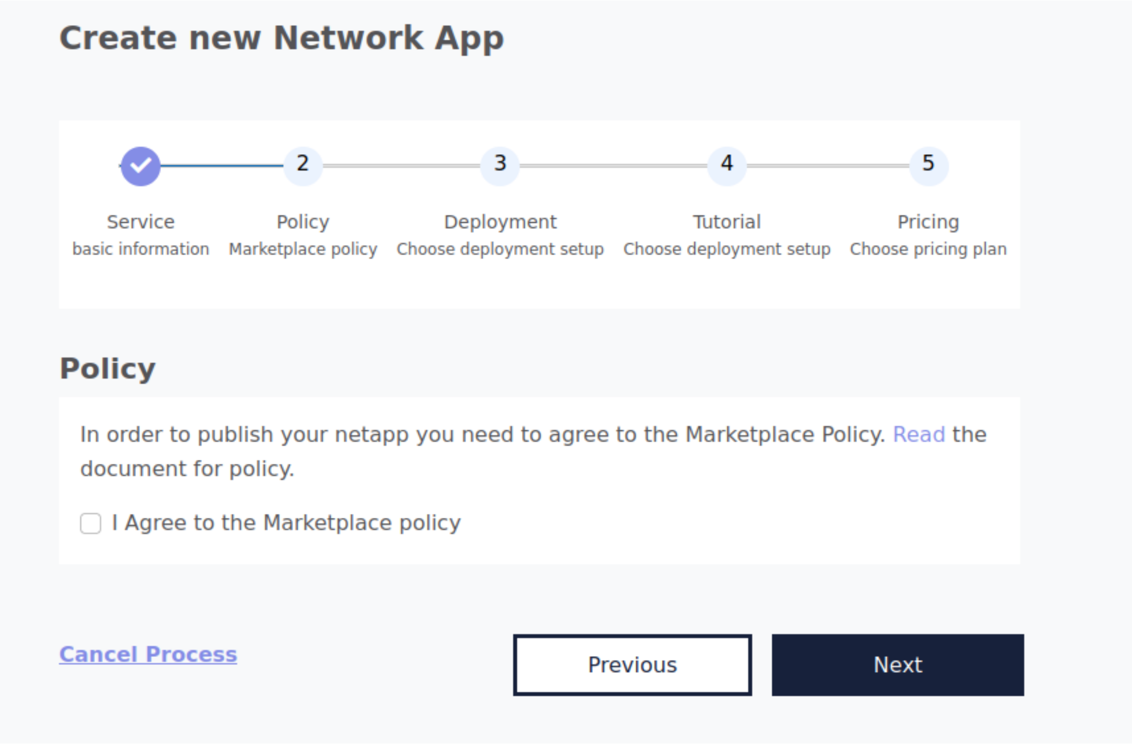
View more (Marketing page) url site
Do you have an external page/URL hat demonstrates your NetApp (for example in your company's website)? Paste it below:

NetApp logo

Publish as:
● User
○ Business/Organization

Figure 45: Step 1: Network App creator adds basic information

During the second step, the user must agree to the Marketplace policy/terms and conditions.



Create new Network App

✓ Service basic information — 2 Policy Marketplace policy — 3 Deployment Choose deployment setup — 4 Tutorial Choose deployment setup — 5 Pricing Choose pricing plan

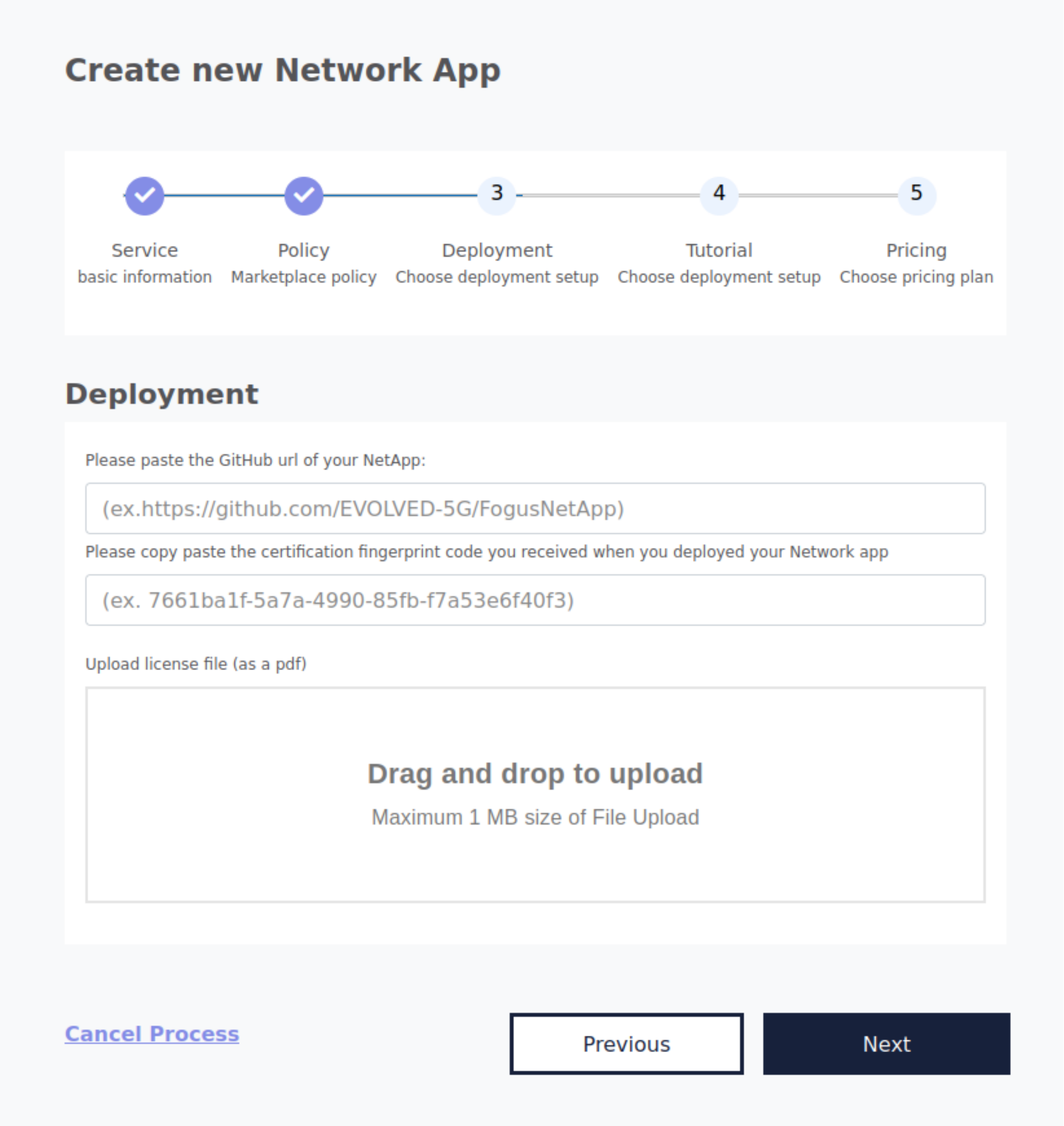
Policy

In order to publish your netapp you need to agree to the Marketplace Policy. Read the document for policy.

□ I Agree to the Marketplace policy

Figure 46: Step 2: Network app creator agrees to the privacy policy

During the 3rd step, the Network App developer provides all the information required for the Marketplace - Open repository integration, as explained in chapter 5.4. Additionally, there is an option to upload a file that contains the Network App license information.



Create new Network App

Deployment

Please paste the GitHub url of your NetApp:

(ex.https://github.com/EVOLVED-5G/FogusNetApp)

Please copy paste the certification fingerprint code you received when you deployed your Network app

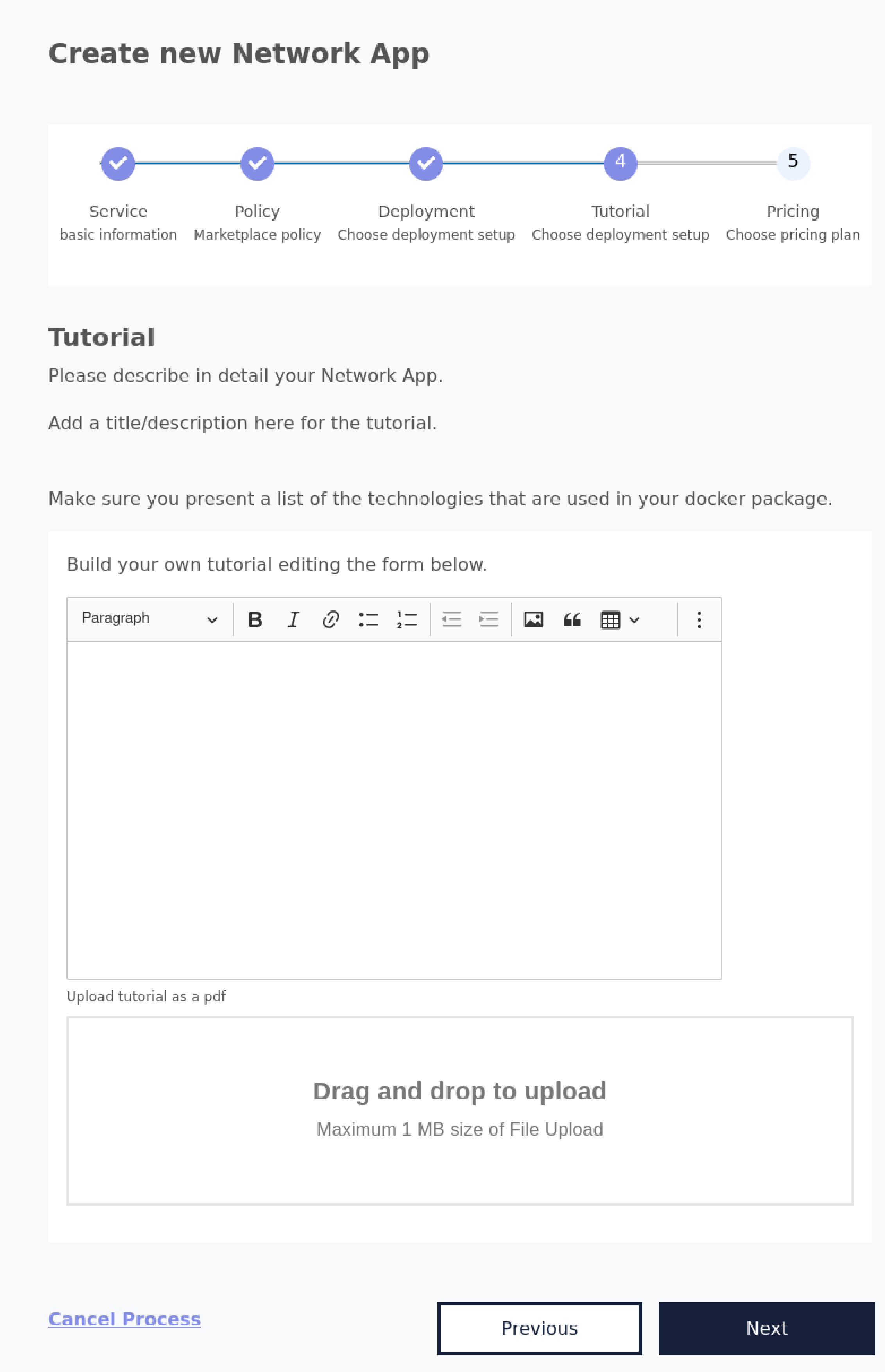
(ex. 7661ba1f-5a7a-4990-85fb-f7a53e6f40f3)

Upload license file (as a pdf)

Drag and drop to upload
Maximum 1 MB size of File Upload

Figure 47: Step 3 - GitHub URL and the certification related unique-id (fingerprint code)

During the fourth step, the user provides information about the Network App. The goal is to provide a tutorial of usage, describing how someone can use the Network App.



Create new Network App

Tutorial

Please describe in detail your Network App.

Add a title/description here for the tutorial.

Make sure you present a list of the technologies that are used in your docker package.

Build your own tutorial editing the form below.

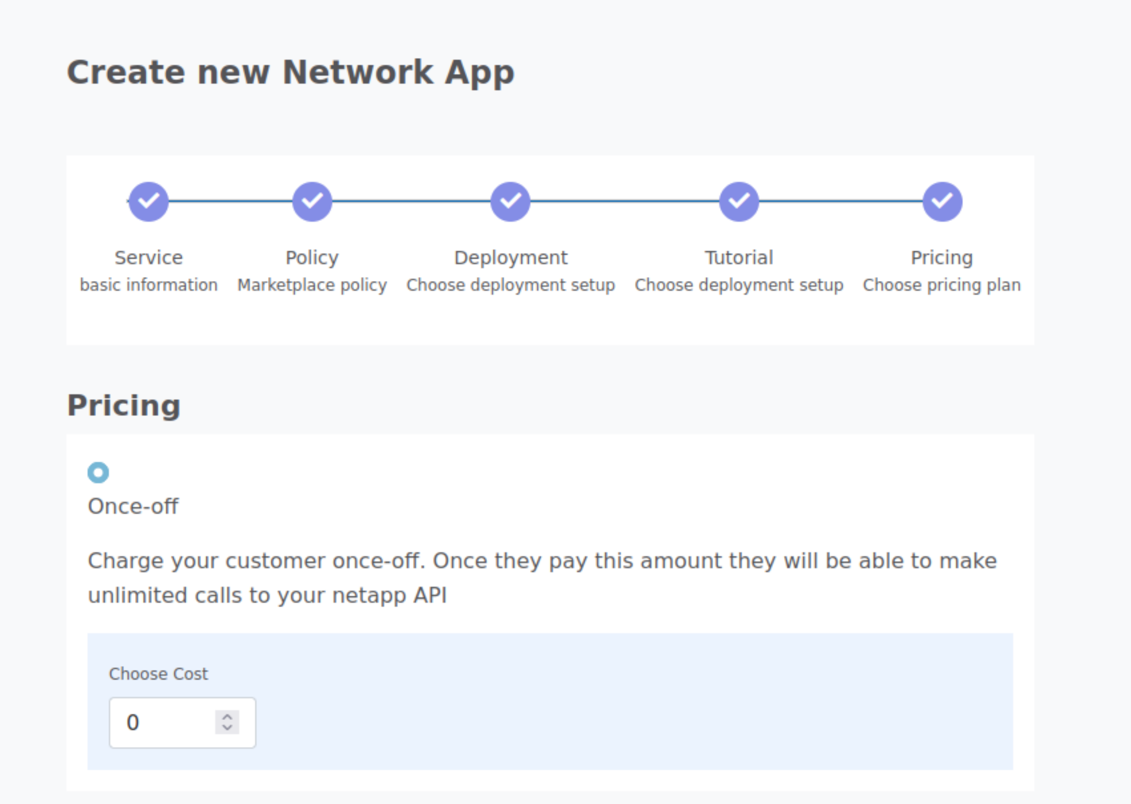
Upload tutorial as a pdf

Drag and drop to upload

Maximum 1 MB size of File Upload

Figure 48: Step 4 - Network App creator adds a tutorial of how one can use the Network App

During the fifth step, the user provides pricing information about the Network App. Two options are available. In the "Once-off pricing" option, a customer pays a fixed amount in order to start using the Network App.



Create new Network App

Pricing

○ Once-off

Charge your customer once-off. Once they pay this amount they will be able to make unlimited calls to your netapp API

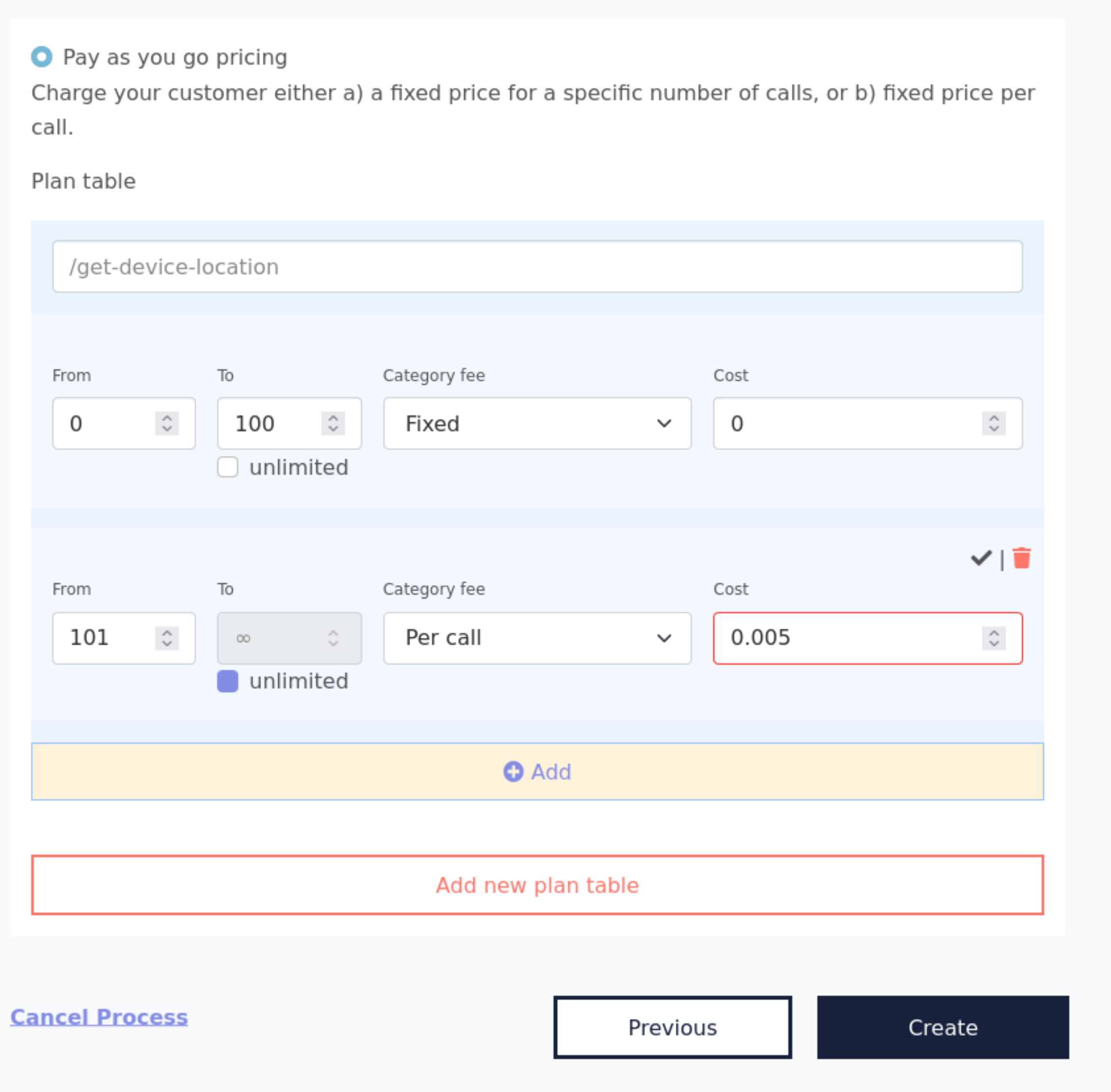
Choose Cost: 0

Figure 49: Step 5 - Network App creator add pricing information - Once off scenario

In the "pay as go" pricing scenario, a Network App creator can define a more flexible and detailed pricing plan.

For example, API endpoints of the Network App can have a relevant cost each time they are consumed.

In the screenshot below the Network App creator has created a pricing plan, where the first 100 calls to endpoint "/get-device-location" are free, and all of the upcoming calls have a fixed price of 0.005€ per call.



● Pay as you go pricing

Charge your customer either a) a fixed price for a specific number of calls, or b) fixed price per call.

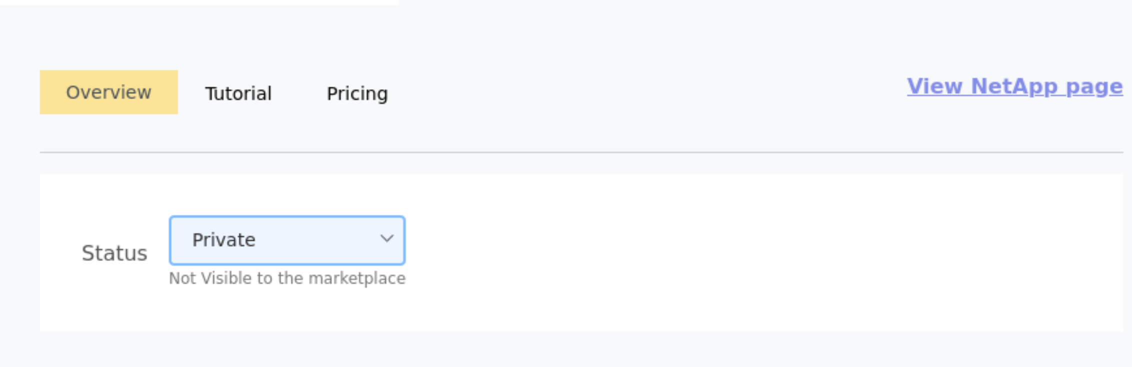
Plan table

/get-device-location

From	To	Category fee	Cost
0	100 □ unlimited	Fixed	0
101	∞ ☑ unlimited	Per call	0.005

Figure 50: Step 5 - Network App creator add pricing information - Flexible pricing scenario

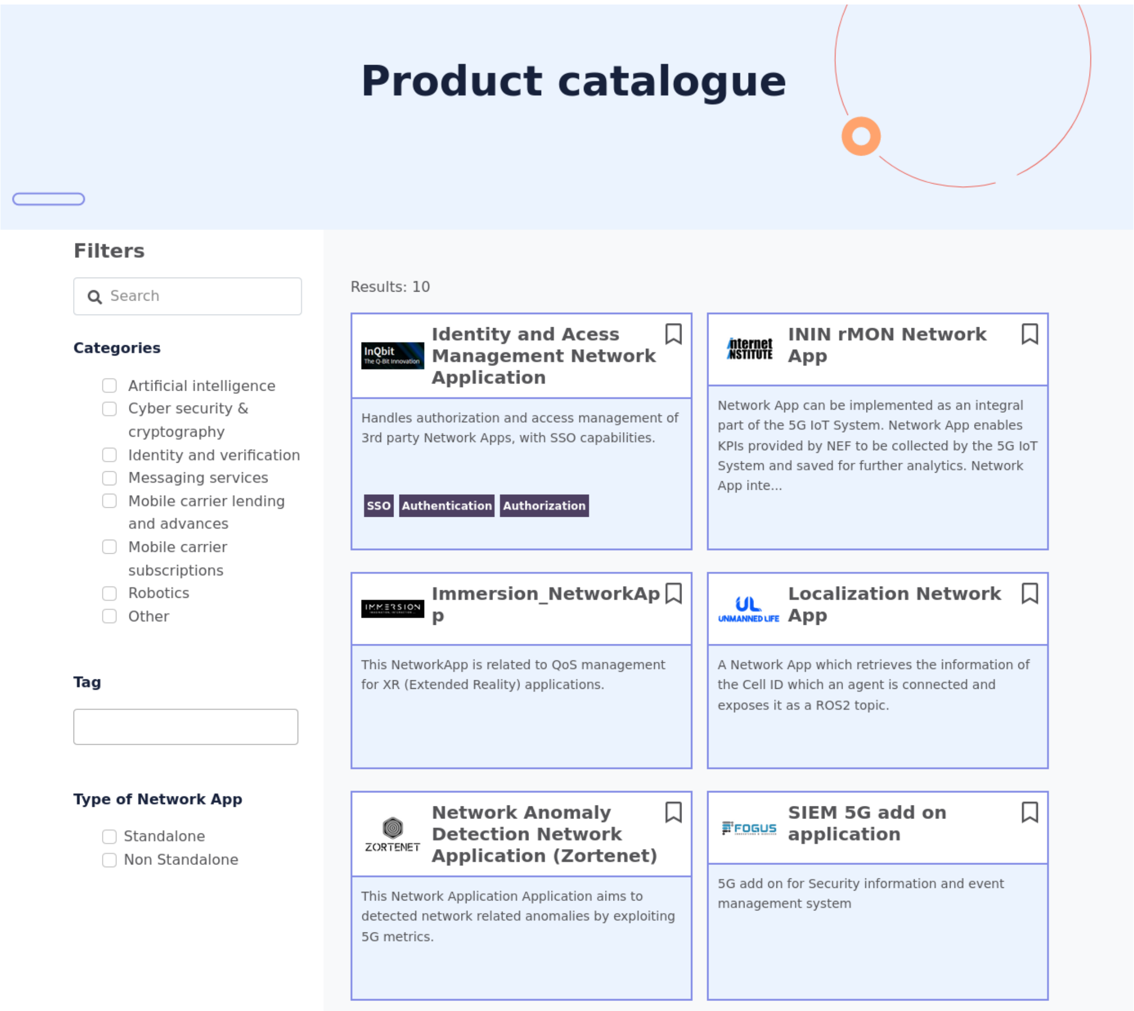
Finally, a Network App creator can save the Network App and switch its status from "Private" to "Public". This action will make the Network App visible to the product catalogue of the Marketplace. A landing page of the Network App, displaying the Network App information (basic details, tutorials and pricing info), will also be available.



Status: Private

Not Visible to the marketplace

Figure 51: Step 5 - Change status to release to the Marketplace



Product catalogue

Filters

Search

Categories

- □ Artificial intelligence
- □ Cyber security & cryptography
- □ Identity and verification
- □ Messaging services
- □ Mobile carrier lending and advances
- □ Mobile carrier subscriptions
- □ Robotics
- □ Other

Tag

Type of Network App

- □ Standalone
- □ Non Standalone

Results: 10

Identity and Acess Management Network Application

Handles authorization and access management of 3rd party Network Apps, with SSO capabilities.

SSO Authentication Authorization

ININ rMON Network App

Network App can be implemented as an integral part of the 5G IoT System. Network App enables KPIs provided by NEF to be collected by the 5G IoT System and saved for further analytics. Network App inte...

Immersion_NetworkApp

This NetworkApp is related to QoS management for XR (Extended Reality) applications.

Localization Network App

A Network App which retrieves the information of the Cell ID which an agent is connected and exposes it as a ROS2 topic.

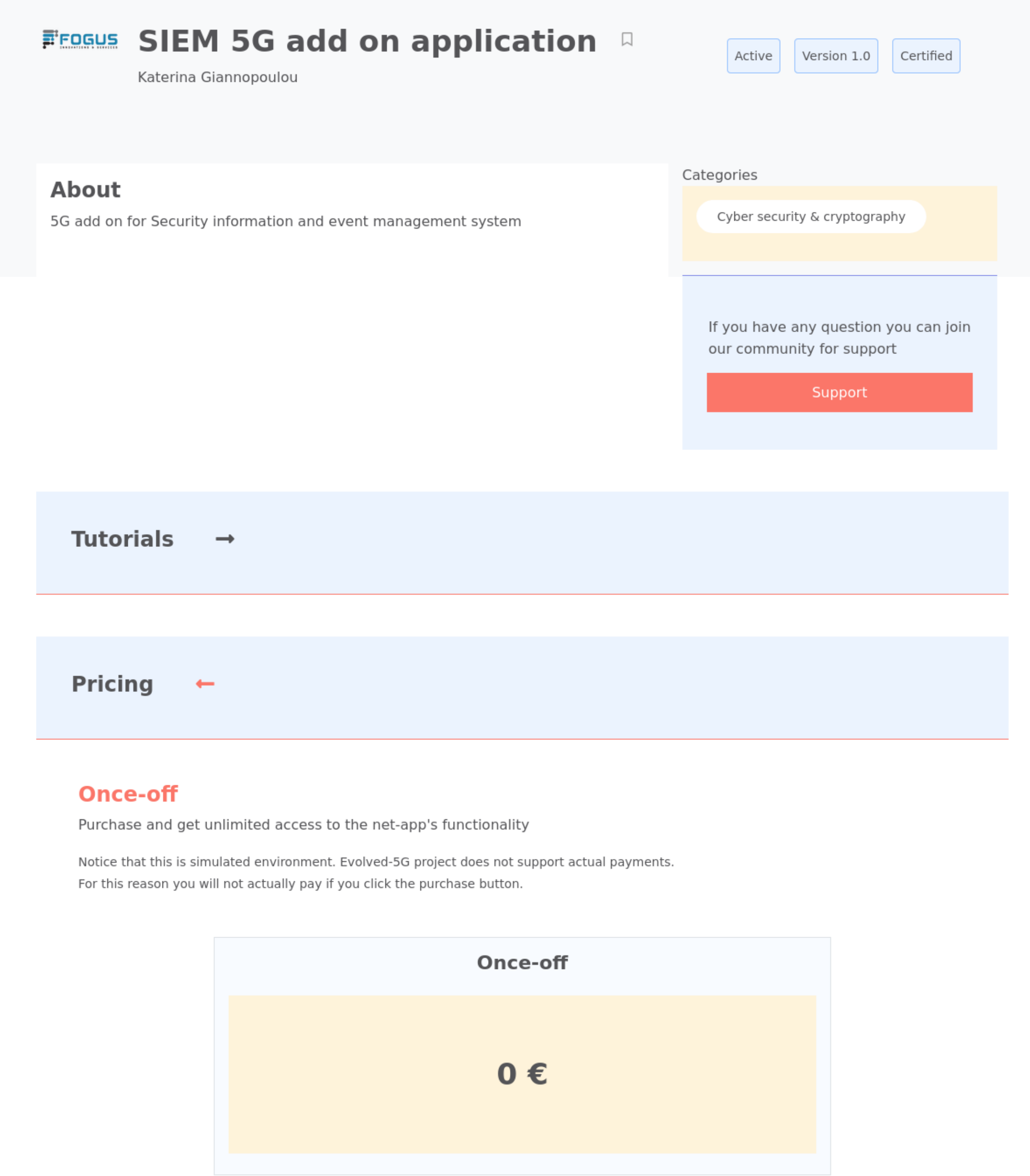
Network Anomaly Detection Network Application (Zortenet)

This Network Application Application aims to detected network related anomalies by exploiting 5G metrics.

SIEM 5G add on application

5G add on for Security information and event management system

Figure 52: Network App is now available in the Product Catalogue



SIEM 5G add on application

Katerina Giannopoulou

About

5G add on for Security information and event management system

Categories

If you have any question you can join our community for support

Once-off

Purchase and get unlimited access to the net-app's functionality

Notice that this is simulated environment. Evolved-5G project does not support actual payments.
For this reason you will not actually pay if you click the purchase button.

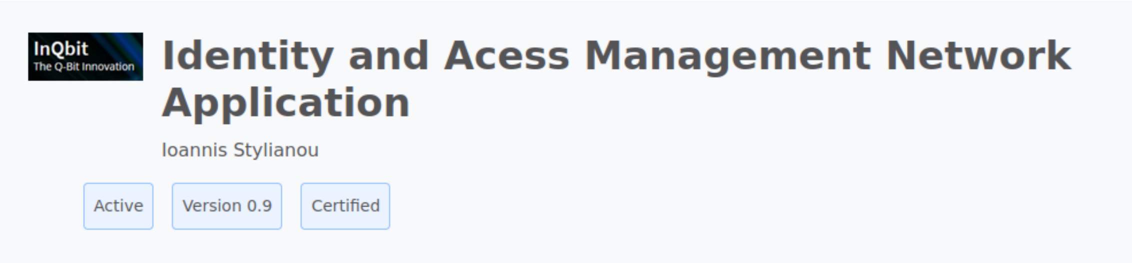
Once-off

0 €

Figure 53: A single public page exists for the Network App

5.7 Network App Versioning

A Network App developer may release different versions of their Network App. These can exist as separate entries in the product catalogue of the Marketplace, each version having its own separate URL address. The version number is displayed in the Network app public page as depicted below.



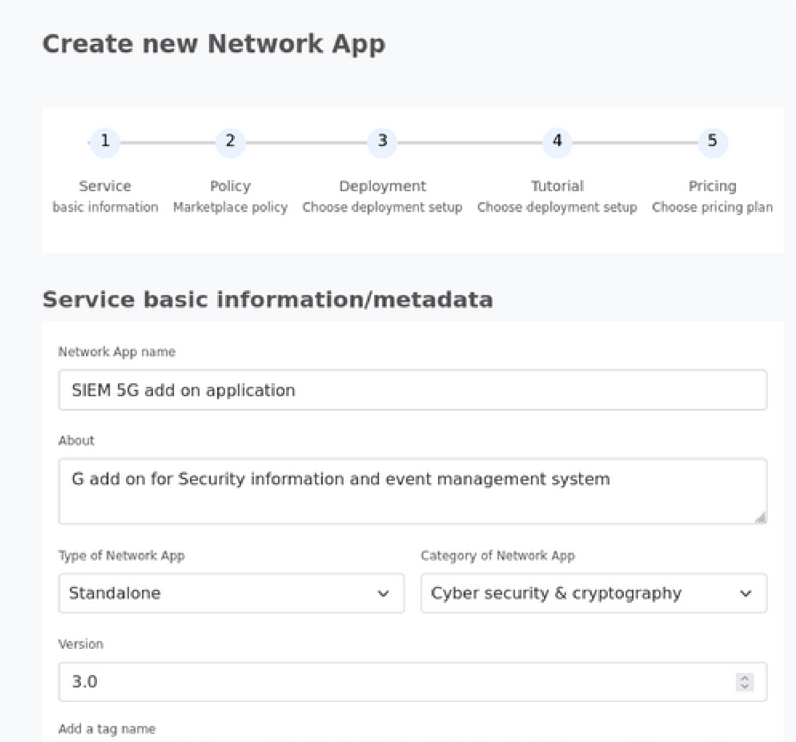
InQbit The Q-Bit Innovation

Identity and Acess Management Network Application

Ioannis Stylianou

Figure 54: A single public page exists for the Network App that displays the version

To upload a new version, the developer has to launch the "Creation Wizard", as explained in chapter 5.6, set the corresponding version and complete all the steps.



Create new Network App

1 Service basic information
2 Policy Marketplace policy
3 Deployment Choose deployment setup
4 Tutorial Choose deployment setup
5 Pricing Choose pricing plan

Service basic information/metadata

Network App name: SIEM 5G add on application

About: G add on for Security information and event management system

Type of Network App: Standalone

Category of Network App: Cyber security & cryptography

Version: 3.0

Add a tag name

Figure 55: Create new Network App Wizard

Different versions can co-exist in the product catalog or older ones can be deactivated by setting their status to "Not visible to the Marketplace" a
This deliverable has presented the work performed in the context of WP3, and more specifically, as part of task T3.4, during the full duration of WP3.

T3.4 designed and started the development of the marketplace and the specification of certification process that Network Apps will have to go through in order to be certified. A detailed description of the tools used to build this certification process and the development of the Marketplace has been provided.

EVOLVED-5G has selected best-in-class tools in each category for processing the Network Apps, scan the source code (Sonarqube and Trivy), scan the Network Apps images (Trivy, NMAP), analyse and identify Licenses dependencies (Debricked), etc

Additionally, the project has developed specific tools for 5G integration such as NEF Emulator and TSN and the API framework to manage those tools like CAPIF Core Function tool.

Regarding next steps, these tools are being integrated in the context of WP5 tools, with special focus in Task 5.2 for Validation and Task 5.3 for Certification, where the Validation and Certification Pipelines are being implemented.

The results of using Validation and Certification pipelines with the Network Apps will be reported in deliverables from WP5 Network Apps D5.5 “Validation and onboarding to Open Repository (Final)” and D5.6 “Network Apps Certification and Release to Marketplace (Final)”.

6 CONCLUSION AND NEXT STEPS

This deliverable has presented the work performed in the context of WP3, and more specifically, as part of task T3.4, during the full duration of WP3.

T3.4 designed and started the development of the marketplace and the specification of certification process that Network Apps will have to go through in order to be certified. A detailed description of the tools used to build this certification process and the development of the marketplace has been provided.

EVOLVED-5G has selected best-in-class tools in each category for processing the Network Apps, scan the source code (Sonarqube and Trivy), scan the Network Apps images (Trivy, NMAP), analyse and identify Licenses dependencies (Debriked), etc

Additionally, the project has developed specific tools for 5G integration such as NEF Emulator and TSN and the API framework to manage those tools like CAPIF Core Function tool.

Regarding next steps, these tools are being integrated in the context of WP5 tools, with special focus in Task 5.2 for Validation and Task 5.3 for Certification, where the Validation and Certification Pipelines are being implemented.

The results of using Validation and Certification pipelines with the Network Apps will be reported in deliverables from WP5 Network Apps D5.5 “Validation and onboarding to Open Repository (Final)” and D5.6 “Network Apps Certification and Release to Marketplace (Final)”.

REFERENCES

[1] EVOLVED-5G, Deliverable 2.1 “Overall Framework Design and Industry 4.0 Requirements”

[2] EVOLVED-5G, Deliverable 3.2 “Network App Certification Tools and Marketplace development (intermediate)”

[3] CICD perspective best practices, from https://en.wikipedia.org/wiki/CI/CD.

[4] GitHub, from .

[5] JFrog Artifactory, from https://www.jfrog.com/confluence/display/JFROG/JFrog+Documentation

[6] AWS Elastic Container Registry, from https://docs.aws.amazon.com/AmazonECR/latest/userguide/Registries.html

[7] Open Container Initiative: https://opencontainers.org

[8] Jenkins https://www.jenkins.io/doc/

[9] Helm Charts: https://helm.sh/docs/topics/charts/

[10] Robot Framework: https://robotframework.org

[11] MetalLB: https://metallb.universe.tf

[12] CAPIF implementation, from https://github.com/EVOLVED-5G/CAPIF_API_Services

[13] TSN FrontEnd https://github.com/EVOLVED-5G/TSN_FrontEnd

[14] EVOLVED-5G, Deliverable 2.2 “Design of the Network Apps development and evaluation environments”

[15] Markdown PDF Converter https://www.npmjs.com/package/markdown-pdf

[16] SDLC. S, Shylesh, A Study of Software Development Life Cycle Process Models (June 10, 2017). Available at http://dx.doi.org/10.2139/ssrn.2988291

[17] Adobe XD, from https://www.adobe.com/products/xd.html

[18] Laravel, from https://laravel.com/

[19] VueJS, from https://vuejs.org/

[20] Bootstrap, from https://getbootstrap.com/docs/5.0/getting-started/introduction/

[21] Sass, from https://sass-lang.com/

[22] MySQL, from https://www.mysql.com/

[23] Discourse, from https://www.discourse.org/

[24] Community Accelerator, from https://evolved-5g.eu/community-accelerator/

[25] TMF620-API https://www.tmforum.org/resources/specification/tmf620-product-catalog-management-api-rest-specification-r17-5-0/

[26] TM Forum server https://github.com/EVOLVED-5G/marketplace-tmf620-api