Deliverable D4.2

Test plan (draft): Draft descriptions of how to evaluate the selected security enablers

Project name	5G Enablers for Network and System Security and Resilience	
Short name	5G-ENSURE	
Grant agreement	671562	
Call	H2020-ICT-2014-2	
Delivery date	02.12.2016	
Dissemination Level:	Public	
Lead beneficiary	PARTNER	Sergio Morant, Sergio.morant@b-com.com
Authors	b<>com: Michel Corriou, Sergio Morant EAB : Mats Naslund Orange: José Sanchez, Jean-Philippe Wary SICS: Rosario Giustolisi TIIT: Madalina Baltatu TS: Edith Felix, Pascal Bisson	

Executive summary

5G-ENSURE aims at providing security proven enablers. In order to achieve this goal, a testbed has been designed within the scope of the project to host the enablers issued from the project. Their security claims will be tested against the security threats previously identified within the project. This will prove efficiency of the features developed.

This document version provides a draft containing the basis to build the complete test plan, the procedures to deliver and integrate the software, and the integration roadmap.

Other WP4 deliverables will arrive afterwards, to provide the complete test plan (D4.3 in M18), and analyse the results of the test plan execution (D4.4 in M24).

This document presents templates and examples of 5G-Ensure tests. Evaluation tests will be described in an add-on document due to the fact the inter Work Packages validation process regarding the Enabler claims of Threats coverage (see chapter 4) has not been fully defined nor endorsed.

The D4.2 Test Plan document embeds a draft version of the “Testbed Terms of Use”, which is under Partners’ legal review before final approval.

Foreword

5G-ENSURE belongs to the first group of EU-funded projects which collaboratively develop 5G under the umbrella of the 5G Infrastructure Public Private Partnership (5G-PPP) in the Horizon 2020 Programme. The overall goal of 5G-ENSURE is to deliver strategic impact across technology and business enablement, standardisation and vision for a secure, resilient and viable 5G network. The project covers research & innovation - from technical solutions (5G security architecture and testbed with 5G security enablers) to market validation and stakeholders' engagement - spanning various application domains.

This document provides the procedures to deliver and integrate the enablers issued from the 5G-ENSURE Software Releases. It also establishes the test plan structure that will allow the validation of the enabler's security claims against the identified security threats.

An important document, the **Testbed Terms of Use** (draft version)**,** is annexed as it provides the rules that are proposed to apply to the testbed usage and so will need to be respected by each partner in order to work with the testbed.

Disclaimer

Copyright notice

Contents

Abbreviations

5G-PPP	5G Infrastructure Public Private Partnership
AAA	Authentication Authorisation Accounting
Dx.y	Deliverable x.y
E.O	Enabler Owner
ETSI	European Telecommunications Standards Institute
ID	Identifier
NFV	Network Function Virtualisation
SDN	Software-Defined Networking
UC	Use Case
VNF	Virtualised Network Function
VPN	Virtual Private Network

1 Introduction

This deliverable covers the aspects of the enabler deployment and evaluation within the 5G-ENSURE testbed. This includes an analysis of the security requirements to cover the security use cases described in D2.1 [1]. This analysis consists in matching the Enabler's security claims described in D3.2 [2] of each enabler's feature, against the different use cases defined in D2.1 [1], and their associated security threats identified in D2.3 [3].

In order to achieve the enabler's integration on the testbed, an integration roadmap is provided in this document together with the procedures in order to deploy and evaluate the enablers.

A test plan for the evaluation of the enabler's security claims against the security use cases is provided as well. This includes the needed process between Work Packages 2, 3 and 4, a first structure regarding needed information to validate the enabler claim regarding threat coverage, and the test plan execution planning.

At this point of time, the target for this deliverable is to provide the required documentation to evaluate the features included on enabler's Release 1. By the time this deliverable is due, the enabler's integration on the testbed will be in an early stage, and the test plan would not be executed yet. Thus, this version of the deliverable provides the test plan structures and some test case examples. The deliverable will be extended and will take a final form in the D4.3 version "*Test plan (final): Final description of how to evaluate the selected security enablers*" (M18). The evaluation results from the test plan execution and the result analysis will be provided at the end of the project (M24) on the D4.4 "*Evaluation of the security enablers: Results and analysis of the Testbed runs*".

Annexed to this document is the provisional **Testbed Terms of Use (Annex A)** that needs to be signed by all the partners willing to use the testbed.

The first structure of descriptive information needed to run the evaluation process between WP2 and WP4 is in Annex B.

This document is based on outcomes from previous deliverables of the project, and then all naming or identification used is referring to precedent deliverables of the project. This includes preserving the structure and identification used to organize the enablers and their features (D3.2 [2]), the use cases (D2.1 [1]), and the threats (D2.3 [3]).

1.1 Definitions

Most of the definitions used in this document have been already defined in the Chapter 1 of D4.1 [4]. In this section only new definitions are added

1.1.1 Enabler evaluation Scenario

Description of (technical or theoretical) steps required to provide evidence for some claim. For instance, a Scenario (set of technical steps description) is used by an Enabler owner to demonstrate that its enabler features covers a specific threat.

Note: A Scenario never mitigates a threat, only an enabler feature mitigates a threat. A Scenario is used to validate and demonstrate that the enabler's threat coverage is more or less effective.

2 Security requirements to cover use case needs

2.1 Enabler's delivered software features

In Table 1 we gather all the enablers and their corresponding features to be integrated in the R1.

Note that those enablers and features are classified as a function of the security group they belong, namely AAA, Privacy, Trust, Security monitoring, and network management & virtualisation isolation.

This classification determines the indexing of the enablers and corresponding features, where the indexing of the features and enablers will be consistent throughout this deliverable.

The list of enablers and security features is taken from the deliverable D3.1 "5G-PPP security enablers technical road map" [D3.1REF], where enablers and features are defined to be integrated in the 5G testbed R1.

Table 1: Enabler's delivered software features

Id	Security Group	Owner	5G-ENSURE security enablers	Features for 1st sw release (R1)
1	AAA	SICS	1.1 Internet of things (IoT)	1.1.1 Group authentication by extending the LTE-AKA protocol
		TASE	1.2 Fine-grained Authorization	1.2.1: Basic Authorization in Satellite systems 1.2.2: Basic distributed authorization Enforcement for RCDs
2	Privacy	TIIT	2.1 Privacy Enhanced Identity Protection	2.1.1: Encryption of Long Term Identifiers **(**IMSI public-key based encryption**)**
		OXFORD	2.2 Device identifier(s) privacy	2.2.1: Enhanced privacy for network attachment protocols
3	Trust	TCS	3.1 VNF Certification	3.1.1: VNF Trustworthiness Evaluation
		VTT	3.2 Trust Metric	3.2.1: Trust metric based network domain security policy management
		IT-INNOV	3.3 Trust Builder	3.3.1: 5G Asset Model 3.3.2: 5G Threat knowledge base v1
4	Security Monitoring	ORANGE	4.1 Generic Collector Interface	4.1.1: Log and Event Processing
		VTT	4.2 Security Monitor for 5G Micro-Segments	4.2.1: Complex Event Processing Framework for Security Monitoring and Inferencing
		TASE	4.3 Satellite Network Monitoring	4.3.1: Pseudo real-time monitoring 4.3.2 : Threat detection
		TS	4.4 PulSAR: Proactive Security Analysis and Remediation	4.4.1: 5G specific vulnerability schema
		IT-INNOV	4.5 System security state repository	4.5.1 : Deployment model ontology
5	Network Management and Virtualization Isolation	NEC	5.1 Access Control Mechanisms	5.1.1: Southbound Reference Monitor
		NEC	5.2 Component-Interaction Audits	5.2.1: Basic OpenFlow Compliance Checker
		SICS	5.3 Bootstrapping Trust	5.3.1 Integrity Attestation of virtual network components
		VTT	5.4 Micro Segmentation	5.4.1: Dynamic Arrangement of Micro-Segments

2.2 Relevant use cases covered by each feature

In this section we present the relationships between each feature to be integrated in the testbed and the Use Cases (UCs). The indexing of the features is the same as shown in the previous section and will be consistent throughout this deliverable to ensure the coherence.

Each feature of this table is related to the different use cases as specified in D3.1 "5G-PPP security enablers technical road map" [5].

Table 2: Relevant use cases covered by each feature

ID Feature	Relevant use cases
1.1.1	UC3.1 : Authentication of IoT Devices in 5G
1.2.1	UC1.3 : Satellite Identity Management for 5G Access
1.2.2	UC4.1 : Authorization in Resource-Constrained Devices Supported by 5G Network
2.1.1	UC2.2: Subscriber Identity Privacy UC2.3: Enhanced Communication Privacy
2.2.1	UC2.1: Device Identity Privacy UC2.2: Subscriber Identity Privacy
3.1.1	UC5.2: Adding a 5G Node to a Virtualized Core Network UC5.4: Verification of the Virtualized Node and the Virtualization Platform UC5.5: Control and Monitoring of Slice by Service Provider UC9.3: Authentication of New Network Elements
3.2.1	UC5.2: Adding a 5G Node to a Virtualized Core Network UC 5.5: Control and Monitoring of Slice by Service Provider UC7.1: Unprotected Mobility Management Exposes Network for Denial of Service UC9.1: Alternative Roaming in 5G
3.3.1	UC1.1 : Factory Device Identity Management for 5G Access UC3.1 : Authentication of IoT Devices in 5G
3.3.2	UC3.2 : Network-Based Key Management for End-to-End Security UC5.1 : Virtualized Core Networks, and Network Slicing UC9.3 : Authentication of New Network Elements UC11 .1: Lawful Interception in a Dynamic 5G Network UC11.2: End-to-end Encryption in LI-aware network
4.4.1	UC5.5: Control and Monitoring of Slice by Service Provider
4.3.1	UC5.6: Integrated Satellite and Terrestrial Systems Monitor UC8.1: Satellite-Capable eNB
4.3.2	
4.1.1	UC5.1: Virtualized Core Networks, and Network Slicing UC5.4: Verification of the Virtualized Node and the Virtualization Platform UC5.5: Control and Monitoring of Slice by Service Provider UC5.6: Integrated Satellite and Terrestrial Systems Monitor UC7.1: Unprotected Mobility Management Exposes Network for Denial of Service UC8.1: Satellite-Capable eNB UC8.2: Standalone EPC UC9.3: Authentication of New Network Elements UC10.1: Botnet Mitigation UC10.2: Privacy Violation Mitigation UC11.1: Lawful Interception in a Dynamic 5G Network UC8.2: Standalone EPC UC9.3: Authentication of New Network Elements UC10.1: Botnet Mitigation UC10.2: Privacy Violation Mitigation UC11.1: Lawful Interception in a Dynamic 5G Network
4.2.1	UC5.5: Control and Monitoring of Slice by Service Provider UC10.1: Botnet Mitigation

4.5.1	UC5.1: Virtualized Core Networks, and Network Slicing UC5.4: Verification of the Virtualized Node and the Virtualization Platform UC5.5: Control and Monitoring of Slice by Service Provider
5.1.1	UC4.2: Authorization for end-to-end IP connections UC5.2: Adding a 5G node to a virtualized core network UC9.3: Authentication of new network elements UC11.1: Lawful interception in a dynamic 5G network
5.2.1	UC5.2: Adding a 5G node to a virtualized core network UC5.4: Verification of the virtualized node and the virtualization platform UC9.3: Authentication of new network elements UC11.1: Lawful interception in a dynamic 5G network
5.3.1	UC5.1: Virtualized core networks, and network slicing UC5.2: Adding a 5G node to a virtualized core network UC5.4: Verification of the virtualized node and the virtualization platform UC9.3: Authentication of new network elements
5.4.1	UC5.1: Virtualized core networks and network slicing UC5.2: Adding a 5G node to a virtualized core network UC3.1: Authentication of IoT devices in 5G UC3.2: Network-based key management for end-to-end security UC1.3: Satellite identity management for 5G access

2.3 Enabler's security claims against use cases

This section contains all the features per enabler to be integrated in the ENSURE platform as well as the goal of each feature and a detailed description of those threats covered by the features. The threats listed in this table are with regard to the identified relevant use cases shown in the previous section.

Table 3: Enabler's security claims against use cases

ID Feature	Goal	Threats covered
1.1.1	*Enable 5G to support massive deployments of IoT devices by adding explicit support for group authentication of devices .*	• **T_UC3.1_1 :** Authentication traffic spikes • **T_UC1.4_1 :** Compromised Data. • **T_UC2.2_2 :** Mobile user interception and information interception
1.2.1	*To support access control of multiple users with different rights in satellite devices and services.*	• **T_UC1.3_1 :** Unauthorised activities related to satellite devices or (satellite) network resources • **T_UC1.3_2 :** Fake roaming from terrestrial network into satellite network • **T_UC5.6_1 : S**ecurity threats in a satellite network
1.2.2	*To support access control on RCDs based on existing http solutions using ABAC and adapted for these devices.*	• **T_UC3.1_1** : Authentication traffic spikes • **T_UC3.1_2** : Compromised authentication gateway • **T_UC4.1_1 :** Unauthorized data access
2.1.1 **2.2.1**	*Limit (preferably totally avoid) exposing user identities on (at least) the air interface* *Limit exposure of device identifiers and prior points of attachment, and therefore, limit the ability to track a device.*	• **T_UC2.2_1 : T**racking of device's (user's) location • **T_UC2.1_1 :** Mobile user interception and information interception. • **T_UC2.2_2 :** Mobile user interception and information interception
3.1.1	*Certify the trustworthy implementation of the VNF and to expose their characteristics through a Digital Trustworthiness Certificate.*	• **T_UC5.2_1 :** Add malicious nodes into core network • **T_UC5.2_2 :** Forwarding logic leakage • **T_UC5.2_3 :** Manipulation of forwarding logic • **T_UC5.5_1 :** Misuse of open control and monitoring interfaces • **T_UC5.5_2 :** Unauthorized access to a network slice • **T_UC5.5_3 :** Bogus monitoring data • **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers

		• **T_UC9.3_1 :** Hardening or patching of systems is not done • **T_UC9.3_2 :** Unauthentic device installed into the system
3.2.1	*Enable service providers to offer trust based services for customers in mass market and industry.*	• **T_UC3.1_1** : Authentication traffic spikes • **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers
3.3.1 **3.3.2**	*Allow the modelling of 5G networks using the information gathered.* *Allow the mapping of a limited subset of threats to the designed 5G system.*	• **T_UC3.1_1** : Authentication traffic spikes • **T_UC3.1_2** : Compromised authentication gateway • **T_UC3.2_1 :** Leaking keys • **T_UC5.1_1 :** Misbehaving control plane • **T_UC9.3_1 :** Hardening or patching of systems is not done • **T_UC9.3_2 :** Unauthentic device installed into the system • **T_UC11.1_1 :** Compromised / malicious LI (Lawful Interception) function • **T_UC11.2_1 :** Nefarious activities (manipulation of information, interception of information) over LI-aware network
4.4.1	*Extension of the Cyber Attack modelling.*	• **T_UC5.1_1 :** Misbehaving control plane • **T_UC5.5_1 :** Misuse of open control and monitoring interfaces
4.3.1 **4.3.2**	*Provide pseudo real-time monitoring of the satellite network* *Include rules in the monitoring system that correlate different incidents to detect specific threats and vulnerabilities in the satellite network.*	• **T_UC5.6_1 :** Security threats in a satellite network • **T_UC8.1_1 :** Service failure over satellite capable eNB • **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers • **T_UC5.5_3 :** Bogus monitoring data • **T_UC1.3_2 :** Fake roaming from terrestrial network into satellite network (and vice versa)
4.1.1	*Interoperability between events and logs format, in order to allow FastData technologies to be deployed inside the 5G Network*	• **T_UC1.4_1 :** Compromised Data • **T_UC5.1_1 :** Misbehaving control plane • **T_UC7.1_1 :** Denial of service due to Unprotected Mobility Management Exposes Network • **T_UC5.5_1 :** Misuse of open control and monitoring interfaces • **T_UC5.5_2 :** Unauthorized access to a network slice • **T_UC5.5_3 :** Bogus monitoring data • **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers • **T_UC9.3_1 :** Hardening or patching of systems is not done • **T_UC9.3_2 :** Unauthentic device installed into the system • **T_UC5.6_1 :** Security threats in a satellite network • **T_UC8.1_1 :** Service failure over satellite capable eNB • **T_UC10.2_1 :** Nefarious activities (malicious software, unauthorized activities, interception of information): privacy violations
4.2.1	*Enable distributed security monitoring and reactions to security incidents.*	• **T_UC5.1_1 :** Misbehaving control plane • **T_UC5.5_1 :** Misuse of open control and monitoring interfaces • **T_UC5.5_2 :** Unauthorized access to a network slice**T_UC9.3_2 :** Unauthentic device installed into the system
5.1.1	*Enforce access control policies that account for the southbound API of an SDN controller.*	• **T_UC5.2_2 :** Forwarding logic leakage • **T_UC5.2_1 :** Add malicious nodes into core network • **T_UC5.1_1 :** Misbehaving control plane • **T_UC3.2_1 :** Leaking keys • **T_UC5.1_1 :** Misbehaving control plane • **T_UC3.2_1 :** Leaking keys • **T_UC5.2_3 :** Manipulation of forwarding logic • **T_UC9.3_1 :** Hardening or patching of systems is not done • **T_UC9.3_2 :** Unauthentic device installed into the system • **T_UC11.1_1 :** Compromised / malicious LI (Lawful Interception) function
5.2.1	*Verification of the interaction between multiple network components with respect to simple policies about the components' exchanged OpenFlow messages.*	
5.3.1	*Implement the strictly minimal functionality of software components and protocols necessary to validate the concept of deploying SDN components in isolated execution environments with a hardware root of trust.*	• **T_UC5.2_1 :** Add malicious nodes into core network • **T_UC9.3_2 :** Unauthentic device installed into the system
5.4.1	*Enable dynamic arrangement (create, delete) of micro-segments in the network.*	• **T_UC5.2_1 :** Add malicious nodes into core network

3 Enabler integration roadmap (enablers R1) and current status

As stated in deliverable D4.1 [4], the enabler integration procedure is split in two: the R1 and the R1.1 (shown in Figure 1). The reason for this is that it is preferable to integrate the first set of enablers, which are easiest to integrate, and schedule the more complex enablers once the integration process is mature enough.

The first enabler to be integrated in 5G-ENSURE testbed was the Generic Collector Interface. Indeed, this enabler will collect information that will be sent to some other enablers, that is why its integration was one of the main priorities in the roadmap introduced in D4.1.

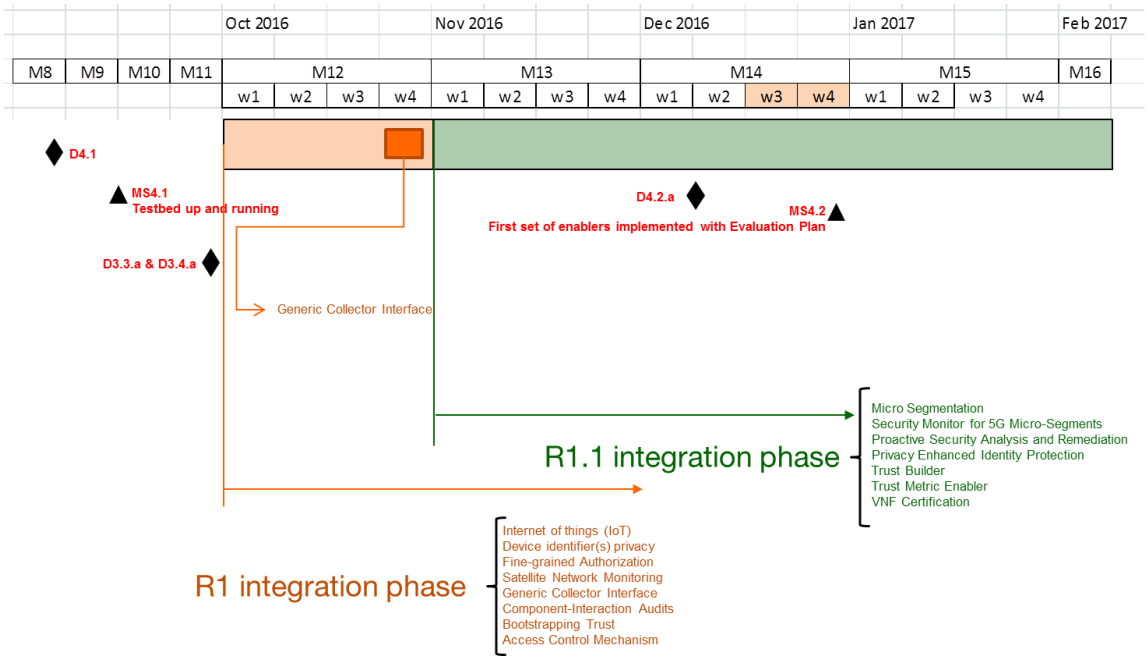

Figure 1: Testbed integration roadmap

The status of R1 Enablers integration at the date of D4.2 publication) is given hereafter. This status is based on the evidences collected from testbed tools (helpdesk, catalogue and test plan; see chapter 4):

Enabler	Feature	Hosting requirements provided	Terms of use Signed	#Packages	Packaging status	Deployment Request	UT description	Threats Claimed	Status
GCI (Orange)	Log and Event Processing	Yes	NO	3	100%	Yes	100% (3)	Yes	Integrated (bug pending)
IoT (SICS)	Group authentication by extending the LTE-AKA protocol	Yes	NO	7	100%	Yes	100% (4)	Yes	Almost completed (one UT KO)
Fine-grained Authorization	Basic Authorization in Satellite systems (TASE)	Yes	NO	?	0%	No	0%	No	(No inputs given)
	Basic distributed authorization Enforcement for RCDs (TS)	Yes	NO	2	100%	Yes	100% (6)	Yes	Integrated
Satellite Network Monitoring (TASE)	Pseudo real-time monitoring	Yes	NO	?	0%	No	0%	No	(No inputs)
	Threat detection	Yes	NO	?	0%	No	0%	No	(No inputs)
Component-Interaction audits (NEC)*	Basic OpenFlow Compliance Checker	Yes	NO	?	0%	No	0%	No	(No inputs, describe UT)
Device identifier(s) privacy	Enhanced privacy for network attachment protocols (OXFORD)	Yes	NO	?	0%	Yes	0% (0)	No	Pending (Packaging and architecture specification)
Bootstrappping trust (SICS)	Integrity Attestation of virtual network components	Yes	NO	1	50%	Yes	0% (0)	Yes	Pending (packaging and unitary tests)
Access control mechanism (NEC)*	Southbound Reference Monitor	Yes	NO	?	0%	No	0%	No	(No inputs, describe UT)
Microsegmentation (VTT)	Dynamic Arrangement of Micro-Segments	Yes	NO	1	80%	Yes	100% (4)	Yes	Pending (architecture proposal)
Security monitor for 5G microsegments (VTT)	Complex Event Processing Framework for Security Monitoring and Inferencing	Yes	NO	1	100%	Yes	100% (2)	Yes	Pending (Unitary tests and architecture proposal)
Pulsar: Proactive security analysis and remediation (TS)	5G specific vulnerability schema	No	NO	?	0%	No	100% (4)	No	To be done (docker orchestration ongoing)
Trust builder (IT-INNOV)	5G Asset Model	Yes	NO	1	100%	Yes	100% (2)	Yes	Integrated
	Graphical editor v1	Yes	NO	1	100%	Yes	100% (1)	Yes	Integrated

Trust metric enabler (VTT)	Trust metric based network domain security policy management	Yes	NO	1	100%	Yes	100% (3)	Yes	Pending (architecture proposal)
VNF certification (TCS)	VNF Trustworthiness Evaluation	Yes	NO	2	100%	Yes	100% (3)	No	Ongoing (architecture proposal)
Privacy Enhanced Identity Protection	Encryption of Long Term Identifiers (IMSI public-key based encryption)	Yes	NO	6	0%	No	20% (1)	Yes	Pending (packaging and unitary tests)

4 Testing procedures for the testbed

This section provides the procedures in support of the enabler testbed lifecycle. These procedures will be enhanced and provided with complementary details on the project's workspace wiki. In this way, the procedure will be able to evolve in time without compromising the concordance with the content described in this chapter.

Notice: the tools referred in this document (TestLink, Artifactory, Ansible, etc) have already been introduced in D4.1 [4]. Please refer to this document for more detailed information about the tool description and their use in the scope of the testbed

4.1 Enabler testbed lifecycle

The testbed lifecycle has been split in three main stages as shown on Figure 2

Figure 2: Enabler testbed lifecycle

- **Delivery** of the enabler to the testbed
- **Integration** of the enabler in the testbed allowing to the assertion of enabler's testbed acceptance
- **Evaluation** of the enabler against the security threats related to the security UCs

The first two stages (delivery and integration) constitute the **deployment process** of the enabler in the testbed, which ends up with the enabler acceptance. The last stage (evaluation) allows to evaluate and grade to which extent the security claims of the enabler are covered.

The web-based TestLink [6] system is used to describe each unitary test (or acceptance test) and evaluation tests over the testbed. Each project entity needing to access a specific test or Scenario description should refer to TestLink.

4.2 Enabler deployment strategy

This section proposes the workflows and procedures for the delivery and integration of an enabler over the testbed as opposed to the evaluation of the enabler, which takes place later in the process and checks the coherence of the enabler with respect to the expressed requirements. The process of delivery and integration of an enabler requires the collaboration and exchange of information among several actors for an optimal result.

As stated in D4.1, "*in order to provide the required degree of conformity for a telco grade platform, the deployment of the testbed instances will be handled by the Testbed Operator who will ensure that the required engineering rules are applied to all the instances running on the testbed.*"

Therefore, the process of delivery and acceptation of an enabler consists of several procedures whose goal is to ensure the good transfer of information between the Enabler Owner and the Testbed Operator.

4.2.1 Delivery process

The enabler delivery process is composed of three steps as depicted in Figure 3. This process is led by the Enabler Owner who is supported by the Testbed Operator.

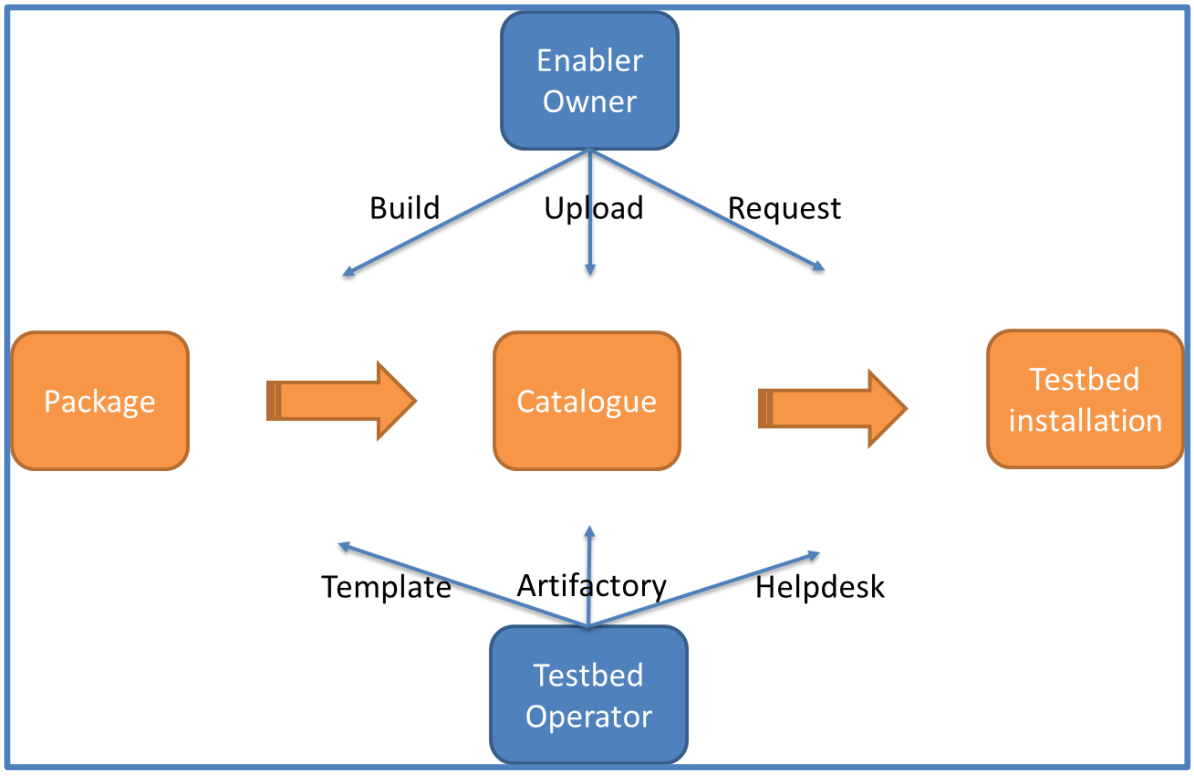

Figure 3: Delivery workflow

- The *package* build activity where the software, package and documentation is made ready.
- The *catalogue* upload activity where the enabler is being uploaded on the testbed.
- The *testbed installation* request permits to trigger the integration process by means of a deployment request through the helpdesk.

As concerns the Package build, the Enabler Owner builds a package containing:

- The dependencies.

- The enabler object code.
- The configuration file(s).
- The Ansible [7] configuration role (optional).

The Testbed Operator provides templates to simplify this task (the packaging and the Ansible [7] role definition).

Second, for the Catalogue upload, the Enabler Owner uploads the package to the 5G-ENSURE testbed catalogue. The catalogue is based on Artifactory [8] provided by the Testbed Operator. A dedicated repository is used for 5G-ENSURE enablers. The Enabler Owner provides the dependencies if they are not available as standard distribution packages. This procedure is detailed in section 4.3.

Third, for the testbed installation request, the Enabler Owner requests the enabler deployment through the helpdesk. A dedicated ticket template is available for this specific request. This communication channel is important for managing these requests and track resource allocation.

Figure 4 illustrates the helpdesk deployment request template:

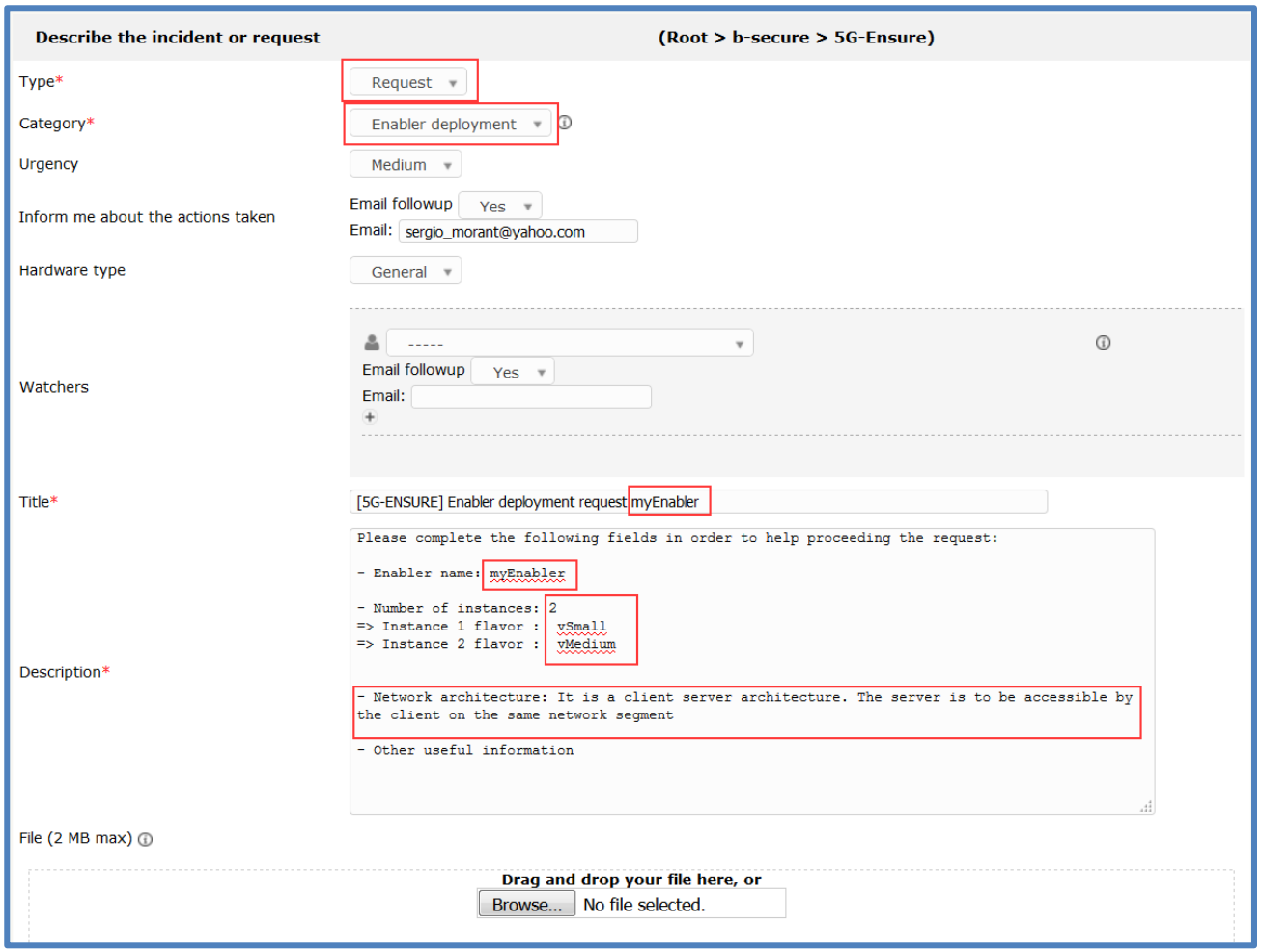

Figure 4: Helpdesk deployment request template

In order to trigger this template, the Enabler Owner needs to create a new **Request** ticket on the helpdesk and choose the **Enabler Deployment** category.

Then, the template will pre-set the required fields with the default information. The Enabler Owner should complete the ticket, before submitting it, with the following information:

Title: add the enabler name as defined in D3.2 [2].

Description: Provide as much information as possible to help preparing the deployment, namely:

- The number of instances to deploy and their flavours.
- If the enabler is composed of several packages, specify in which instance they should be deployed.
- The requested network architecture allowing the interconnection of all requested instances, and with any other required equipment. Architecture can be delivered as an attached document in the deployment request.
- Any other information that could help the Testbed Operator improve the understanding of the request.

Hereunder, Figure 5 resumes the request created for the Generic Collector Interface deployment as example:

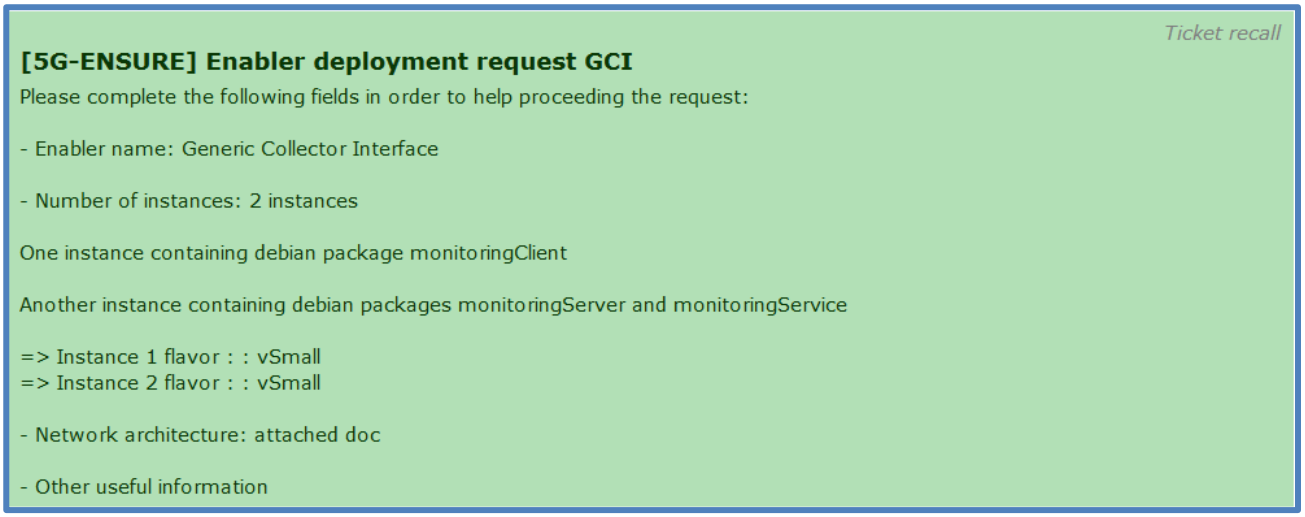

Figure 5: GCI helpdesk deployment request

4.2.2 Integration workflow

Figure 6 depicts the steps that need to be performed to complete the integration on the testbed. In this case, the process is driven by the Testbed Operator with the support of the Enabler Owner.

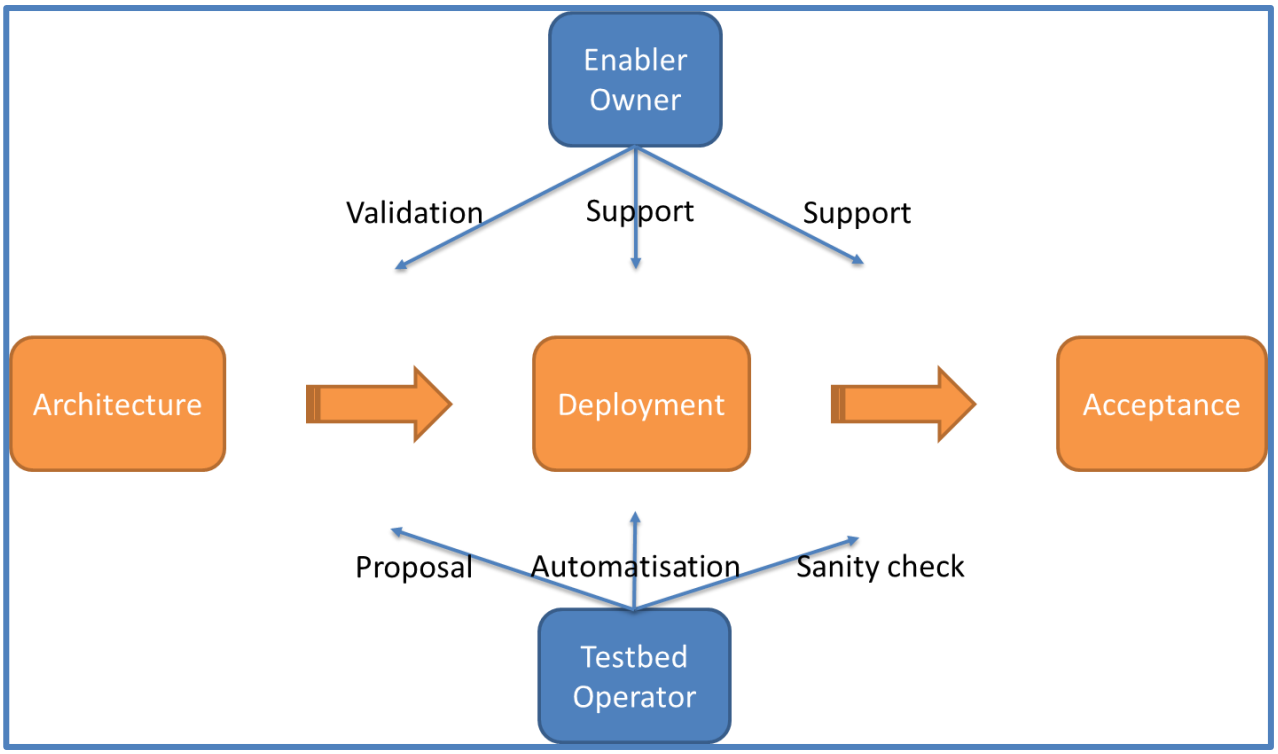

Figure 6: Integration workflow

The architecture proposal allows the specification of a target deployment architecture for the enabler and its associated components. The proposal will be based on the following inputs:

- The enabler User Manual present in D3.4 [9].
- The content of the deployment request generated by the Enabler Owner through the helpdesk.

The Testbed Operator will provide, by answering the helpdesk request, a deployment architecture proposal containing the information required by the Enabler Owner to validate the correctness of the deployment.

The following example (Figure 7 and Figure 8) contains the proposal for the hosting of the Generic Collector Interface.

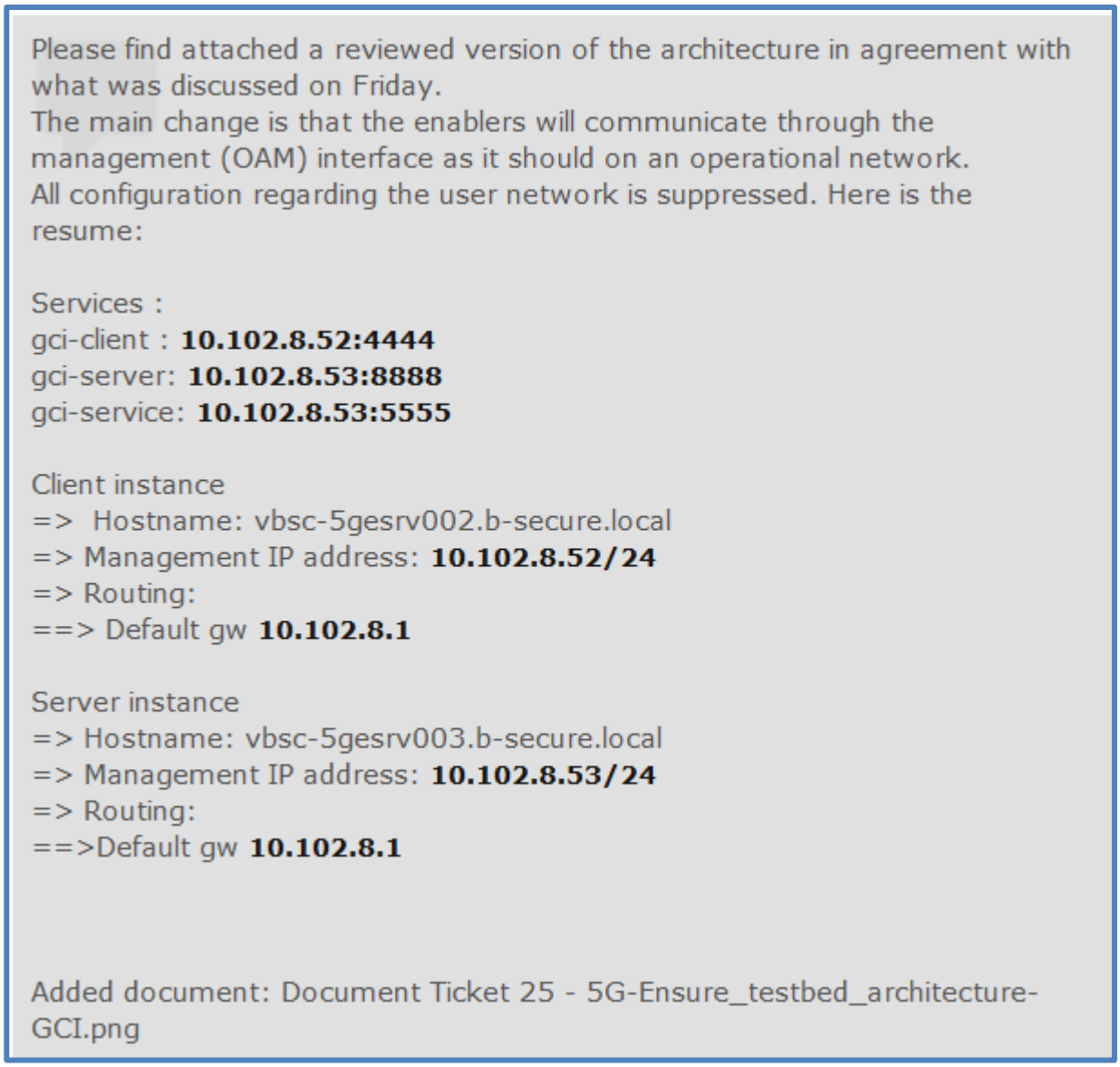

Figure 7: GCI Network configuration proposal

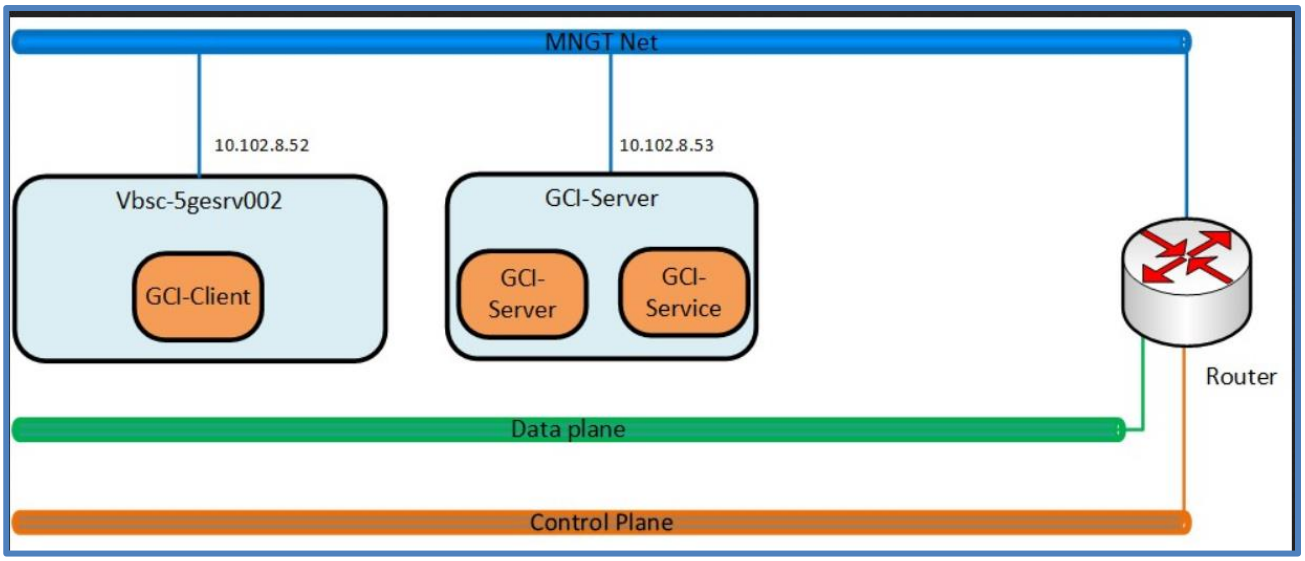

Figure 8: GCI network architecture proposal

Upon validation by the Enabler Owner, which is to be done through the ongoing helpdesk request, the deployment can be triggered.

At this stage the Testbed Operator will map all the collected information to the Orchestration and configuration management tools. Once this step is done the deployment process will be held automatically.

At the end of the process the systems will be deployed with the identified enabler components and the requested configuration. If for any reason there are issues to deploy the target architecture, the Enabler Owner will support the Testbed Operator to identify a solution. The main communication channel to support this action is the helpdesk.

Once the enabler and its associated components are deployed, the acceptance procedure can take place. The goal at this stage is just to run the enabler's unitary tests described in D3.4 [9], which have been integrated as part of the test plan (see section 5.2.1). Running these tests in the testbed, functions as enabler's sanity checks. If the enabler passes the tests, it can be considered as integrated in the testbed. The testbed acceptance of the enabler is announced by means of an official mail to the Enabler Owner and the project Technical Manager.

4.3 Delivering an enabler on the catalogue

This procedure was described in a high level in D4.1 [4]. This section aims at describing the procedure in more details now that the testbed and the catalogue tool are fully operational.

A catalogue tool (Artifactory [8]) is provided within the testbed. It centralizes and manages the delivery and deployment of the enablers within the testbed. Enabler packaging is an operational requirement for the enablers to be deployed on the testbed.

Hereunder the complete procedure to deliver an enabler on the catalogue is specified:

- Connect to the catalogue repository: https://artifact.b-com.com
- Login using the personal testbed credentials. A web page looking like the following should appear:

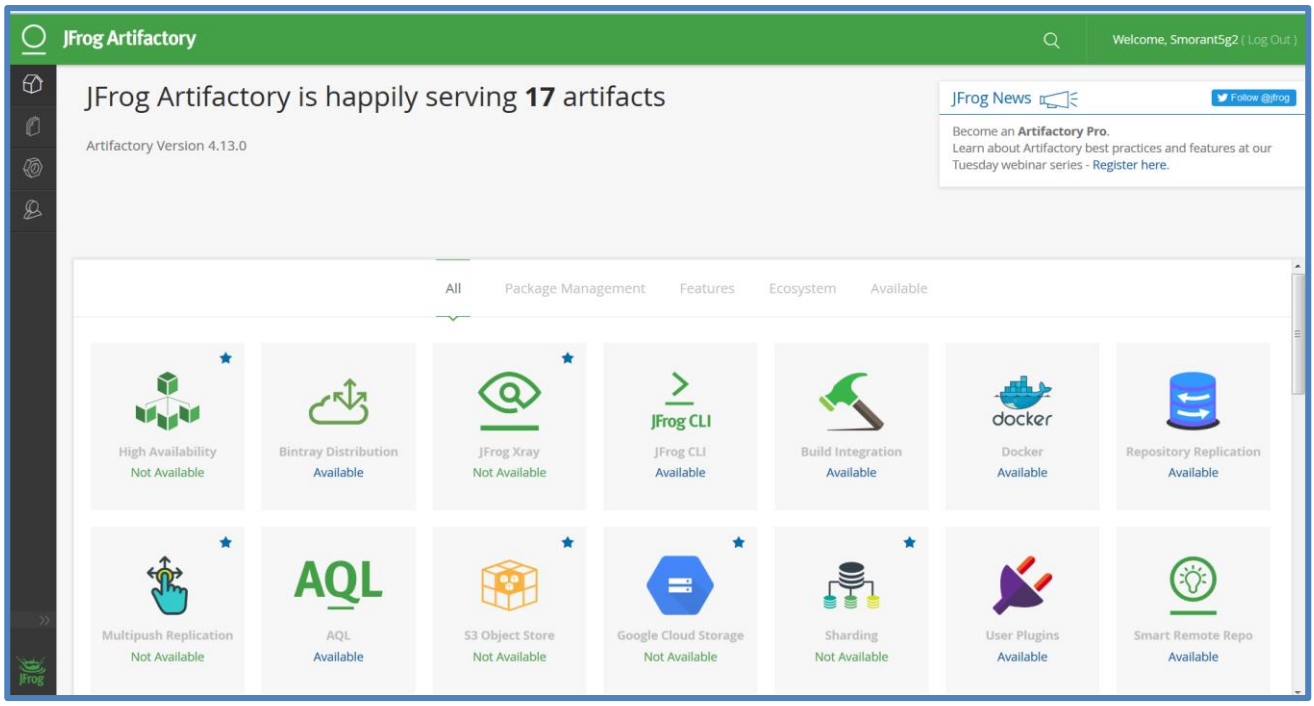

Figure 9: Catalogue home page

- Go to the Artifacts menu. A screen looking like Figure 10 should appear

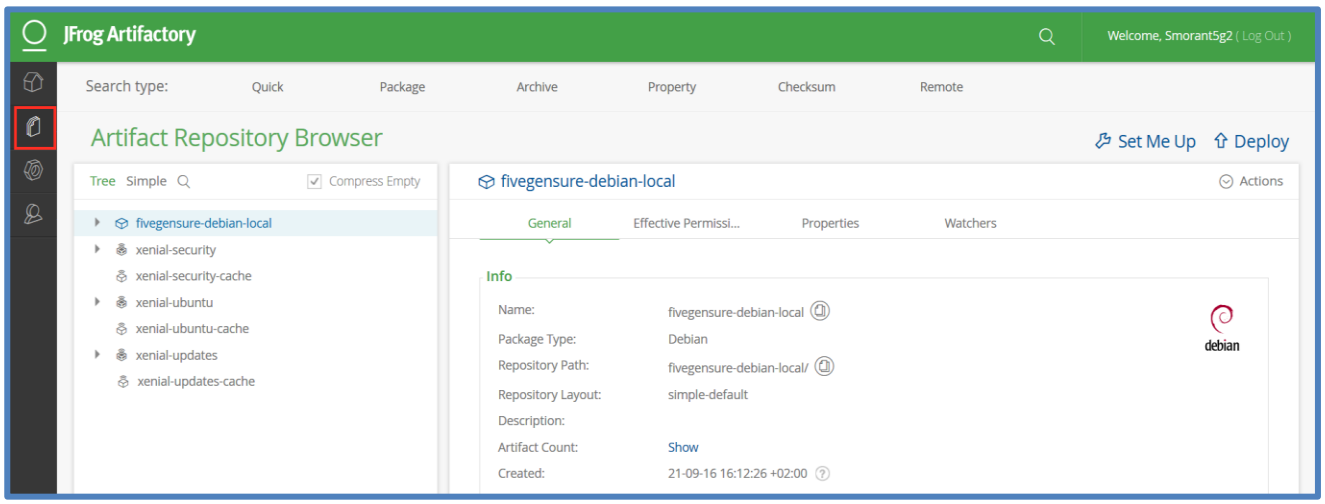

Figure 10: Catalogue repository page

- The following repositories are available at the time of the writing of this deliverable (see Figure 11):
  - **Fivegensure-debian-local**: Repository dedicated for the 5G-ENSURE project enablers for Debian / Ubuntu distributions
  - **Xenial-xxxxx**: Repositories used to cache Ubuntu Xenial distribution packages. This allows to install the system packages on the testbed from a local repository
- Choose the target path on the left hand side of the webpage, taking into account the considerations regarding the operating system (Ubuntu Xenial), the architecture (amd64) and the nature of the enablers regarding their Intellectual property (restricted). This would provide the following target path "*fivegensure-debian-local/dists/xenial/restricted/binary-amd64/*"

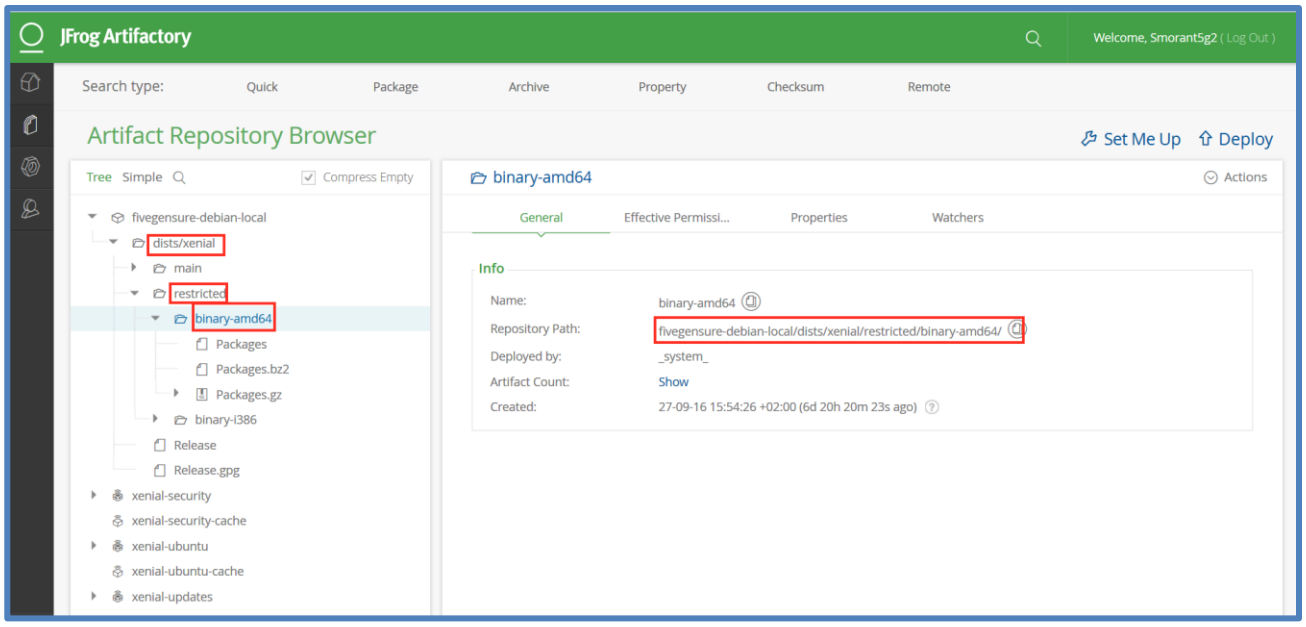

Figure 11: Catalogue 5G-ENSURE Debian / Ubuntu repository

- In order to upload a new package on the catalogue, click on the Deploy button. The following menu will appear (see Figure 12):

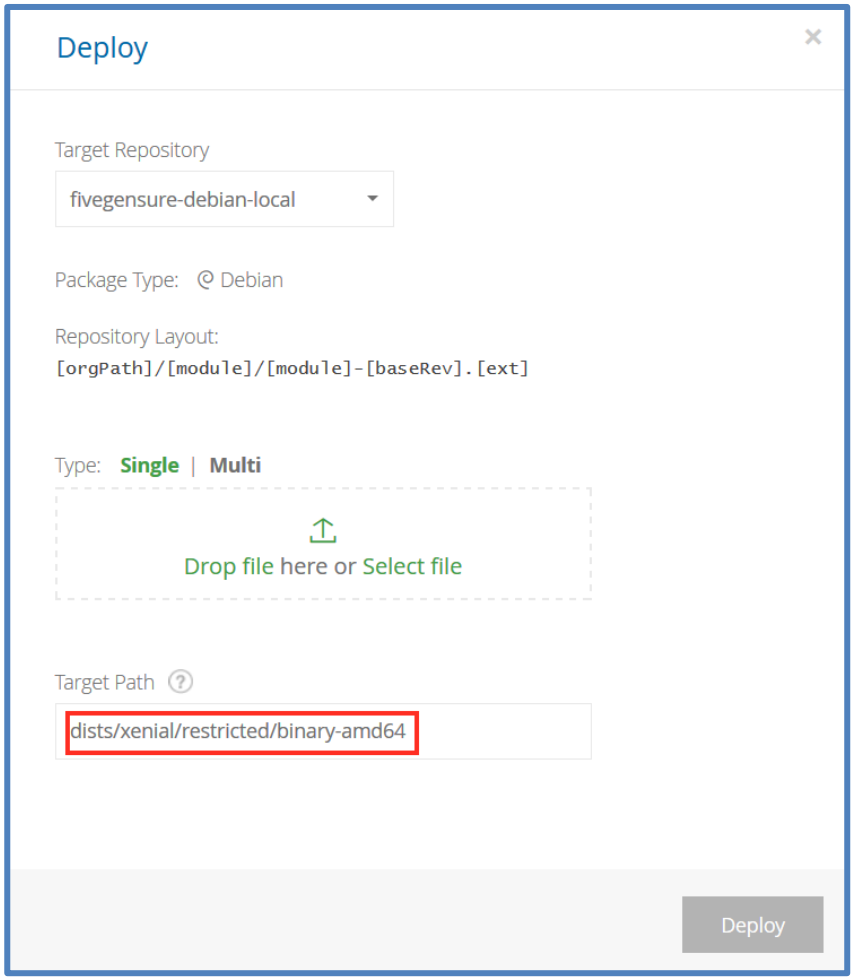

Figure 12: Catalogue deploy menu (1/2)

- Select the file containing the enabler and check that the target path is set as expected. Then click on the “Deploy” button (see Figure 13).

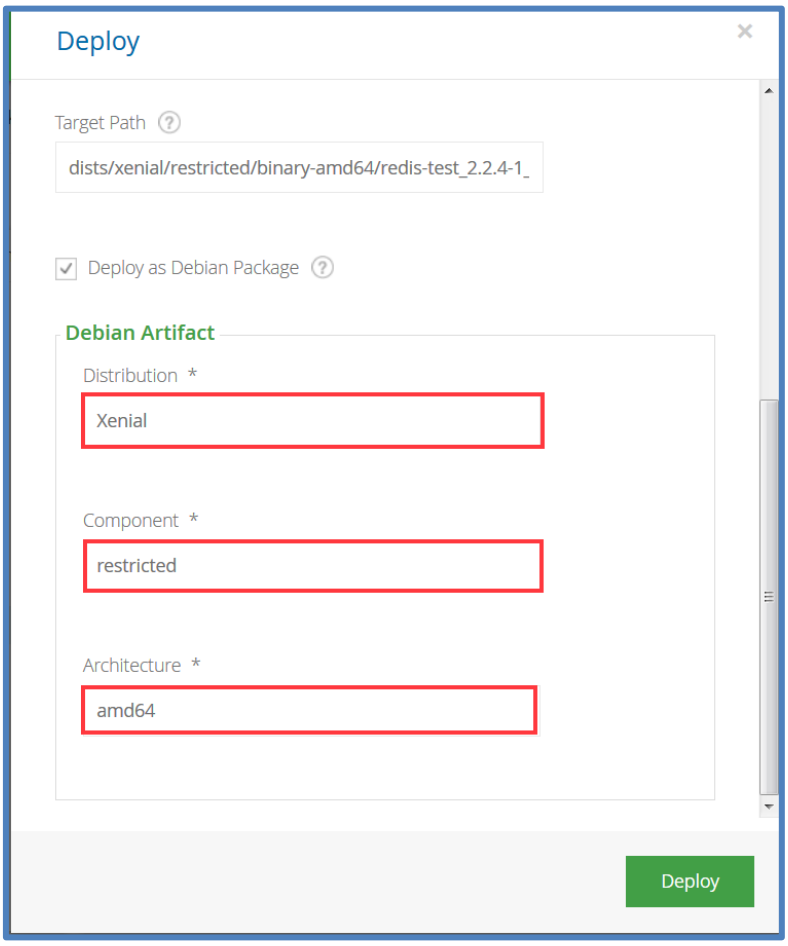

Figure 13: Catalogue deploy menu (2/2)

Alternatively, it is also possible to perform the action using the Artifactory Rest API. In order to get the right format, use the Set Me Up button on the Artifacts menu. This will provide the curl command template. The output should look like as in Figure 14.

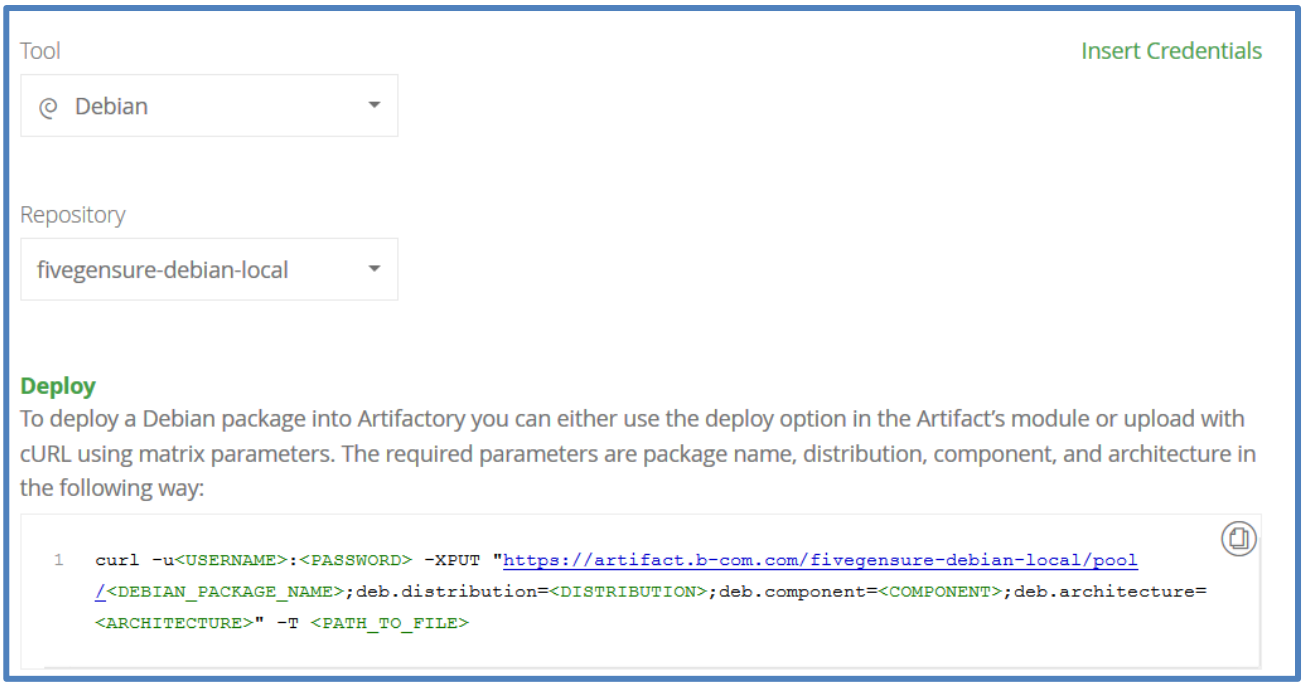

Figure 14: Catalogue Curl template for package upload

At this point, it should be possible to install / update the enabler from any testbed host system by using the repository management tool, or deploy a new Docker container.

For the instances deployed on the testbed, the configuration management tool will configure their repository to point to the catalogue.

4.4 Running an enabler security evaluation

Note: to enter in a **security evaluation stage**, an enabler should have finalized its **integration stage** as described in the chapter 4.1.

The evaluation stage will be performed for a specific pair of two elements defined as (enabler feature, threat), as it was stated in (D3.1 [5], D3.2 [2] and D2.3 [3]).

The Enabler Owner has to describe how its enabler may mitigate the identified threat. This description will be based on the enabler technical specification, threat and uses case, and the testbed's available nodes and resources (see D4.1 [4] testbed architecture description).

In particular, it is not feasible to generate traffic spikes against a simulated network or against the enabler itself over the testbed.

Hereafter is shown the Evaluation Scenario validation:

- **WP2(Task 2.3) is responsible** for validating if the proposed Scenario, delivered by E.O., is sufficient to demonstrate that the enabler addresses and mitigates the identified threat. It is not the WP2 responsibility to look neither at the enabler implementation details, nor penetration test, nor configuration / software security evaluation of the proposed enabler feature.

Note: the structure of the document related to the test evaluation Scenario is proposed for information, as a draft, in Annex B. This structure will evolve based on the information collected during the R1 evaluation stage and will be finalized in the D4.3 deliverable.

- **WP4(Task 4.2) is responsible** for validating if the proposed Scenario (after WP2 validation) is technically compatible with the testbed architecture (see D4.1 [4]).

  In case the proposed Scenario is not technically feasible in the testbed, WP4 will ask the E.O. to specify a new evaluation Scenario, otherwise WP4 may decide that the current testbed is not able to support the proposed evaluation Scenario and then may recommend the E.O. to proceed with a theoretical evidence of coverage.
  This means that the evaluation metric associated to this pair (enabler feature, threat) will be set at "**theoretical or paper-based evidence**" level (see chapter 4.5). In this specific case of theoretical validation, published scientific papers will be accepted as evidence of coverage, as the scientific community support the results if and only if the paper is accepted for publication.

Evaluation Scenario process:

Step 1: To deliver evidence and facts of a threat coverage, the E.O. delivers a description of the Scenario allowing to demonstrate the coverage of the identified threat.

Step 2: This Scenario proposal will be then reviewed by WP2.

Step 3: WP2 notifies to WP4 on the potential demonstration of Scenario proposed by the E.O. to cover the threat for the specific enabler feature.

Step 4: WP4 validates the technical feasibility of the proposed tests and Scenario (in case of issue, we go back to step 1 or finalize the evaluation procedure based on theoretical evidences)

Step 5: WP4 runs tests based on the description in TestLink (under the E.O. responsibility) and performs the evaluation of test result.

Step 6: E.O., WP2 and WP4 validate the achieved results as a project result.

Note: the evaluation performed on the testbed for a specific pair (enabler feature, threat) will be based on the proposed Scenario (under E.O. responsibility), but nothing prevents extra Scenarios from being defined and run after evaluation phase of one enabler feature, regarding the acquired information inside the whole project.

4.5 Project's evaluation metric definitions

We provide hereafter with a set of elementary metrics to evaluate the coverage of the different threats:

0: no evidence of coverage of the threat is delivered

1: theoretical evidence (scientific article) of coverage of the threat is delivered

2: implementation delivered (integration phase on the testbed achieved and evaluation test described inside TestLink have been validated by WP2 without performing it, see 4.4 Running an enabler security evaluation)

3: Evaluation Tests performed on the testbed, based on simulated environment, achieved and positive.

4: Evaluation Tests performed on the testbed have been done over the real testbed flows as described in the evaluation Scenario validated by WP2 and corresponding test description (TestLink).

5: Scientific paper (formal proof) and verification that the tested code and Scenario conform to the scientific paper. This could only happen once level 4 is achieved for the specific (enabler feature, threat).

Some evaluation examples:

- The following pair (enabler feature, threat), where only unitary tests are performed (integration phase) but there is not any theoretical, nor technical, nor scientific evidence on how it covers the claimed threats will be scored with the value "0".
- The following pair (enabler feature, threat), where only theoretical, technical or scientific paper based evidence on how it covers the claimed threats will be scored with the value "1".
- The following pair (enabler feature, threat), where theoretical or scientific evidence is delivered and unitary test(s) are performed (integration phase) will be scored with the value "2".

Those metrics are delivered for each pair (enabler feature, threat).

These metrics are an elementary set of KPIs that allow us to calculate three additional metrics to be delivered as results in the project:

1) an average per identified threat and per enabler (set of features),
2) which is the most efficient enabler feature for a given threat, and
3) the most efficiently covered threat by a given enabler feature

To illustrate the use of these metrics, we propose an example of one enabler feature (A.b.c), which claims to cover the 4 following threats: T_UC3.x_z, T_UC3.2_5, T_UC9.2_2, T_UC10.2_1. They performed the evaluation of the 4 different threats with the following scoring results:

- T_UC3.x_z → 3 (simulation based evidence)
- T_UC3.2_5 → 0 (no evidence)
- T_UC9.2_2 → 4 (real-test bed based evidence)
- T_UC10.2_1→ 1 (theoretical based evidence)

Those scoring results mean that:

- The average result for this enabler feature A.b.c is equal to "2", what gives an approximated idea on its threats coverage.
- The enabler feature A.b.c covers better the threat T_UC9.2_2 than other features claiming to cover it.

Based on the evaluation performed, several metrics are collected which that allows an evaluation per threat at global project level.

For instance, threat T_UC9.2_2 is claimed to be covered by 2 different enabler features, with an average metrics at project level of '3.5' (average between '3' for (D.e.f, T_UC9.2_2) and '4' for (A.b.c, T_UC9.2_2)). The conclusion is that enabler feature (A.b.c) is the one which best covers the given threat.

Example of metrics reporting

ID Feature	Threats claimed (Id)	efficiency of coverage per threat	Enabler efficiency level	Most efficiently covered threat
A.b.c	T_UC3.x_z	3	2	T_UC9.2_2 (4)
	T_UC3.2_5	0		
	T_UC9.2_2	4		
	T_UC10.2_1	1		
D.e.f	T_UC3.2_5	3	2,33	T_UC9.2_2 T_UC3.2_5 (3)
	T_UC9.2_2	3		
	T_UC7.2_1	1		
G.h.i	T_UC3.2_5	3	3	T_UC3.2_5 (3)

Threats claimed (Id)	#enablers claiming to cover the threat	Threat coverage level in the project	More efficient Threat coverage level (enabler)
T_UC3.x_z	1	3	3 (A.b.c)
T_UC3.2_5	3	2	3 (A.b.c , D.e.f)
T_UC7.2_1	1	1	1 (D.e.f)
T_UC9.2_2	2	3,5	4 (A.b.c)
T_UC10.2_1	1	1	1 (A.b.c)

Based on this collection of KPI, D4.3 will be able to deliver a general conclusion of the project 5G-ENSURE.

5 Test plan

This chapter covers the way the test plan has been structured and how this structure is matched against the TestLink [6] web tool, which is provided by the testbed to build the test plan, drive the tests, and collect the results. The chapter also provides a preliminary definition of test cases (the full version will be available in D4.3).

The complete user manual of TestLink is available at [10] and a screencast is available at [11] also.

5.1 Roles

In D4.1 [4] the following roles related to the test plan are defined:

Test plan Editor

It is a (testbed) user that contributes to the edition of the test plan for the project's enabler security validation.

Test plan Executor

It is a (testbed) user that participates to the execution of the test plan and the collection of the results.

In this section, these definitions will be extended in two directions:

- Provide the relationship between these roles and those existing on TestLink.
- Identify the partner's role endorsement

5.1.1 Role matching

TestLink is bundled with 6 different default permission levels built in, as described in [10]. These permission levels are the following:

•**Guest:** *A guest only has permission to view test cases, reports and metrics. He cannot modify anything.*

•**Test Executor:** *A tester has permissions to see and run tests allocated to them.*

•**Test Designer**: *A user can fully work (view and modify) with Test Specification and Requirements.*

•**Test Analyst (or senior tester)**: *A tester can view, create, edit, and delete test cases as well as execute them. Testers lack the permissions to manage test plans, manage Test projects, create milestones, or assign rights. (Initially Senior tester).*

•**Test Leader**: *A leader has all of the same permissions as a tester but also gains the ability to manage test plans, assign rights, create milestones, and manage keywords.*

•**Administrator**: *An administrator has all possible permissions (leader plus the ability to manage test projects and users)*

The roles above are resumed in the Figure 15

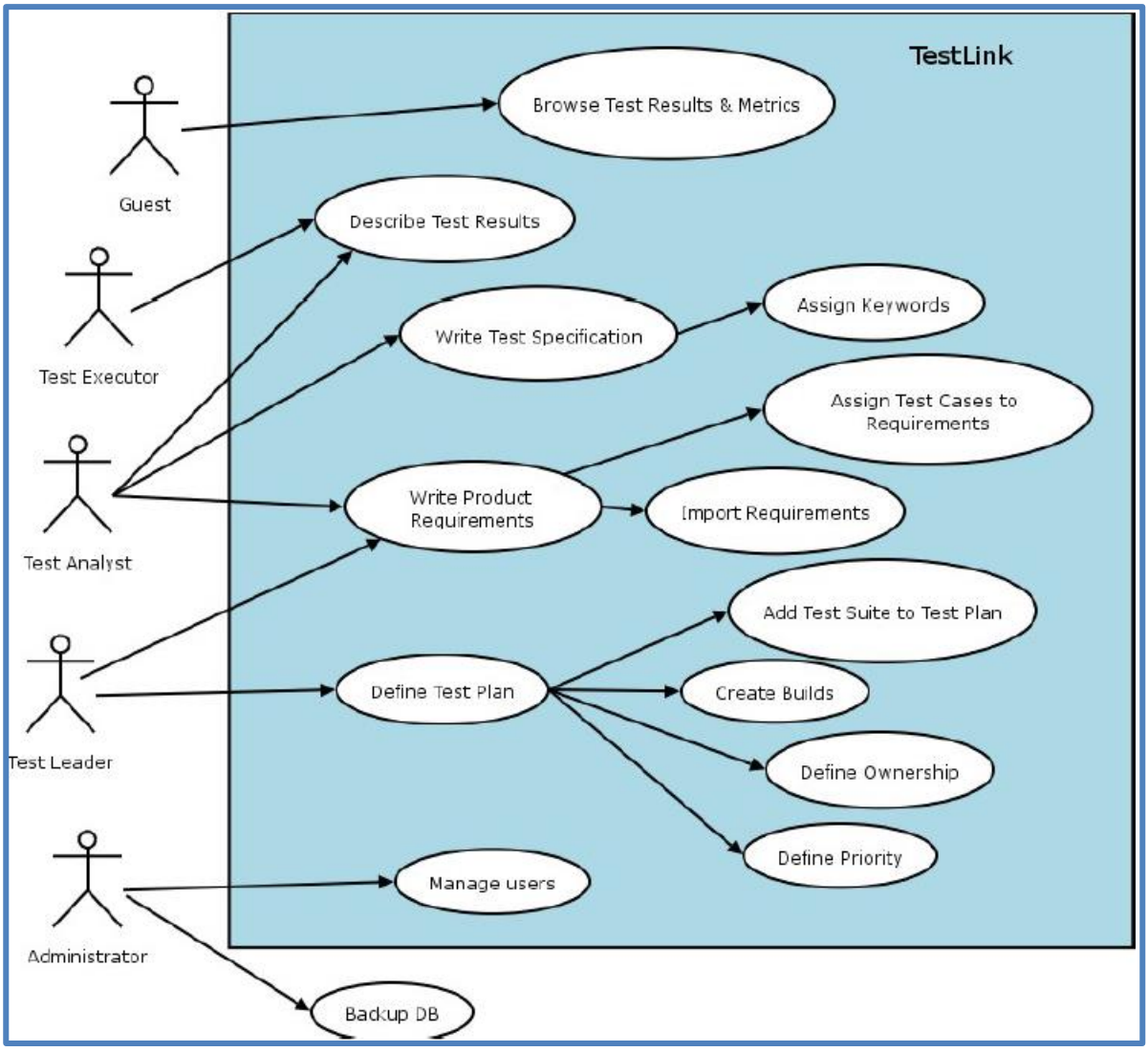

Figure 15: TestLink roles (source [10])

In order to preserve the coherence between the deliverables and for the sake of simplicity, the number of used roles will be preserved. Here is the proposed matching:

- **Testbed Test plan Editor** ➔ TestLink Test Analyst (Senior tester)
- **Testbed Test plan Executor** ➔ TestLink Test Executor

There is a third role, not directly related to the testing strategy, which is the administrator role. It will be played by the Testbed Operator as for any other service provided within the testbed.

5.1.2 Role endorsement

As described in the next section, the test plan will be divided in two threads: enabler feature sanity check and enabler security evaluation. Depending on the threat, endorsement will differ.

Enable feature sanity check

The main goal is to validate the integration of the feature in testbed.

- **Testbed Test plan Editor** ➔ Enabler Owner
- **Testbed Test plan Executor** ➔ Testbed Operator

The test are based on the unitary test cases defined on D3.4 [9]

Enabler security evaluation

- **Testbed Test plan Editor** ➔ Enabler Owner
- **Testbed Test plan Executor** ➔ Partners involved in 5G-ENSURE testbed test plan activities

In this case, the goal is that the enabler owner, in collaboration with WP2 members (see 4.4 Running an enabler security evaluation), establishes the test cases that would allow for evaluation of its enabler against the security threats covered by the enabler The testbed operator will afterwards check the feasibility of the test case within the testbed, and will support the enabler owner to describe them within the scope of the testbed.

5.2 Structure

This section will cover the test plan structure and its mapping against the test plan web tool. As described previously on the document, the goal of the test plan is to provide the means to evaluate the enabler's security claims against the identified security use cases, and their associated security threats. However, it is important also to check that the enablers have been properly integrated on the testbed, prior to start the security evaluation. All the project partners have agreed in structuring the test plan to cover both, the integration and the evaluation tests using TestLink [6].

In a first stage, there will be the unitary tests of D3.4 [9] driven as sanity checks. They will be run at the end of the testbed integration phase. Then, security evaluation related tests will take place during the enabler security evaluation.

In order to use a single tool to collect all test results, the unitary tests described in D3.4 [9] will be added to TestLink. This step will enhance their description in order to correspond with the deployment of the enabler within the testbed.

The structure used by TestLink is described in the detail in the user manual [10]. Here, the focus is on the most important concepts that have been applied to create the test plan. Figure 16 provides the relationship between the objects composing a test plan based on requirements specification as described in [10]. This

approach is particularly adapted to the 5G-ENSURE project, as there has been a considerable effort to describe the enablers feature requirements in the deliverables from WP3, and the UCs and the Threat requirements in the deliverables from WP2.

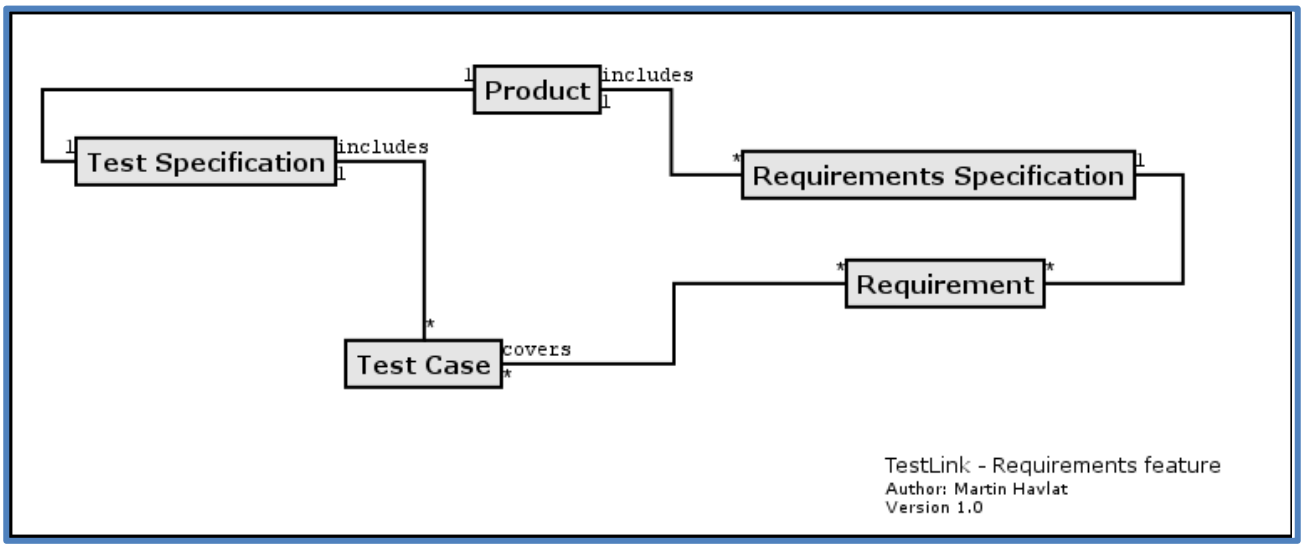

Figure 16: Requirement based test plan (source [10])

In Figure 16, the following elements are depicted:

- **Requirement**: It describes a requirement which can be related to a feature, a use case, a constraint, etc. In the current test plan, the enabler features will be described as feature requirements, and the Security UCs as use case requirements.
- **Requirement specification**: It is a group of related requirements. In the current test plan they are either related to an enabler or a use case cluster.
- **Test case**: It is the testing unit. For each test that needs to be executed on the testbed, there should be a test case providing scope, preconditions, steps to perform the test, and expected results.
- **Test specification (or test suite)**: It defines a group of related test cases. In the scope of the 5G-ENSURE testbook, it is either related to an enabler feature or to a use case.

5.2.1 Enabler's feature sanity checks

Figure 17 depicts the target structure for the testbook with regard the enabler features (sanity checks)

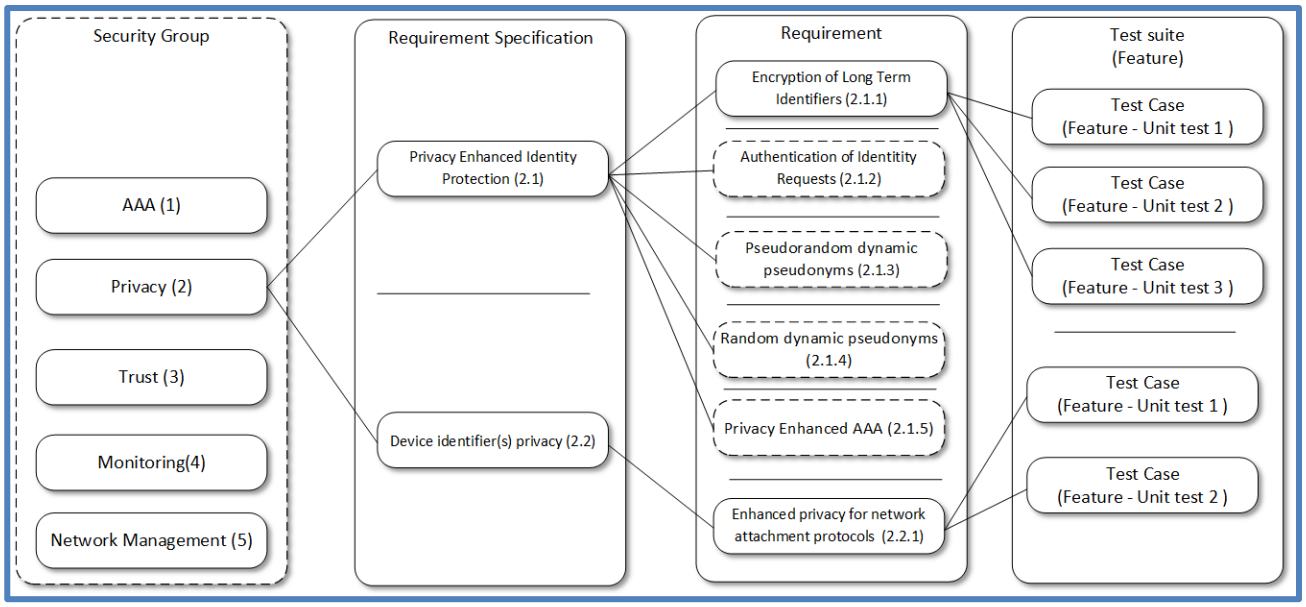

Figure 17: Testbook structure based on enabler features

The goal is to map the structure that has been defined by WP3 for the enablers and their features on the test plan structure. This structure should allow to have a product features based validation approach. This should be compliant with the feature sanity check to be run at the end of the enabler integration on the testbed.

In the rest of this section it is described the way to actually map the illustration of Figure 17 with the objects inside TestLink [6].

Requirement Specification

Figure 18 provides an example of requirement specification for the "*Privacy Enhanced Identity Protection enabler*"

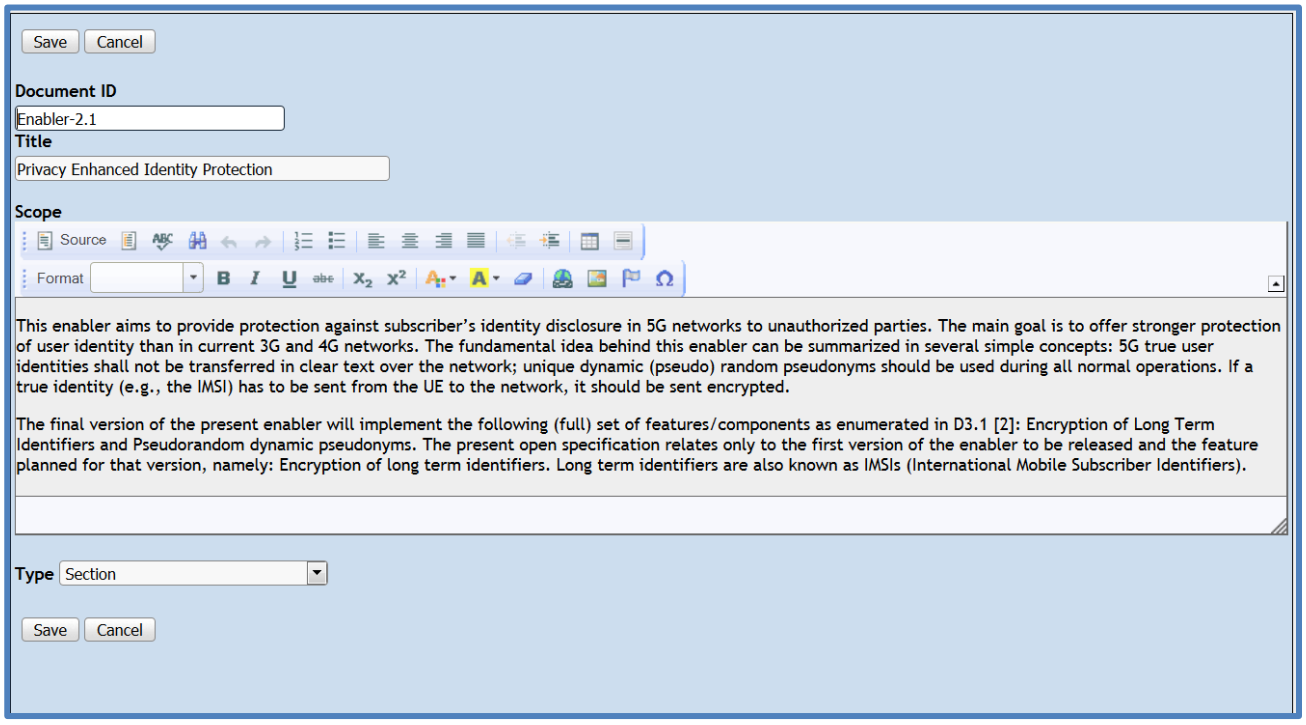

Figure 18: Requirement Specification for "Privacy Enhanced Identity Protection" enabler

The following fields are required to create a requirement specification object:

- **Document id**: Enabler-<enabler_id>.
  - The enabler id must correspond to the identifier assigned in Table 1
- **Title**: <Enabler name>.
  - As defined in D3.2 [2].
- **Scope**: A description of the enabler's scope. In the current example, the text has been extracted from the enabler's Preface section in D3.2 [2].
- **Type**: Section.
  - It is just used to group the features related to the same enabler.

Requirement

Once the Requirement Specification is defined, it is possible to add new requirements inside. Figure 19 provides an example of feature requirement definition based on "*Encryption of Long Term Identifiers*".

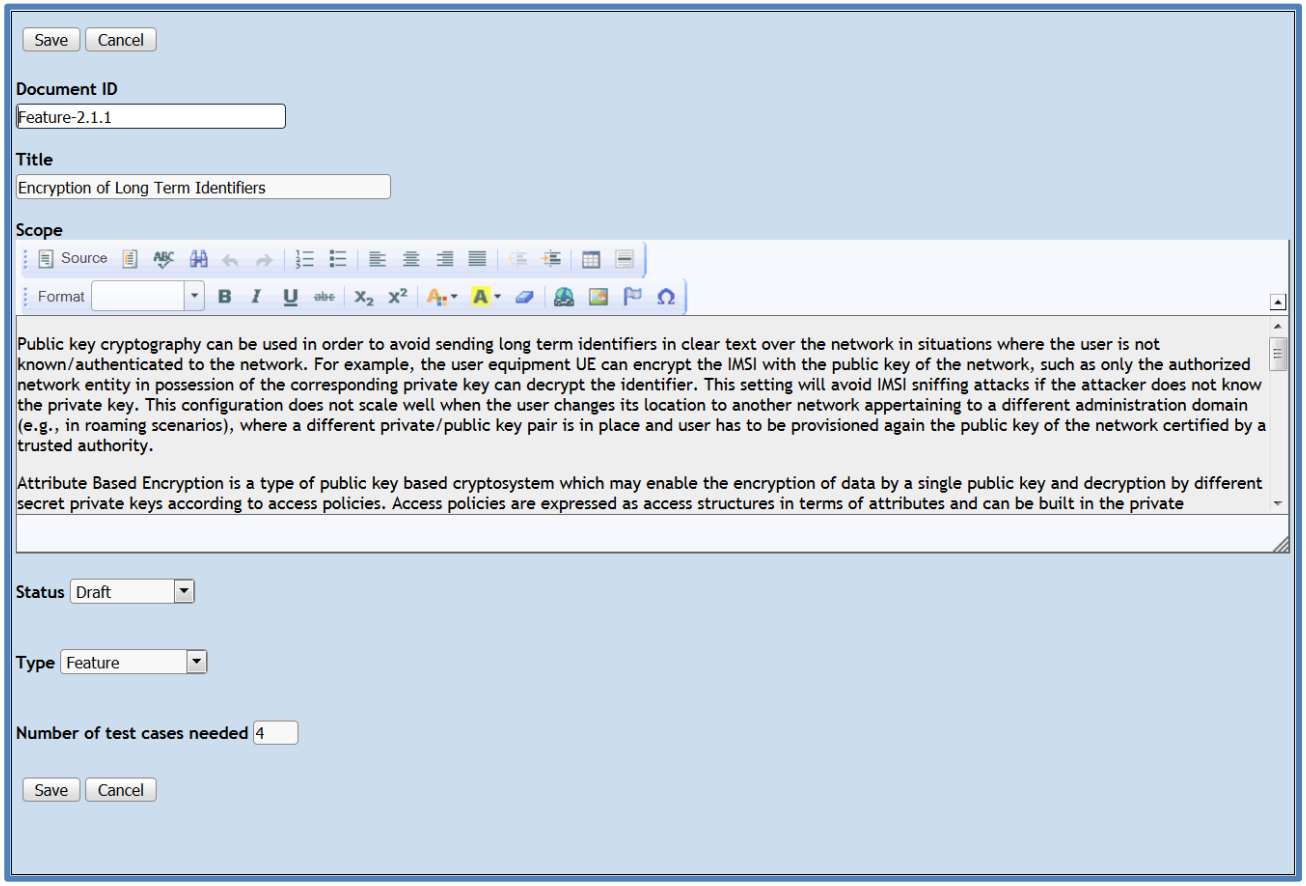

Figure 19: Feature "Encryption of Long Term identifiers" requirement

The following fields are required to create a requirement object:

- **Document id**: Feature-<feature_id>.
  - The feature id must correspond to the identifier assigned in Table 1
- **Title**: <Feature name>.
  - As defined in D3.2 [2].
- **Scope**: A description of the feature's scope. In the current example, the text has been extracted from the Feature basic concepts section in D3.2 [2].
- **Type**: Feature.

Test Suite

Figure 20 depicts an example of test suite definition for the tests related to "*Encryption of Long Term Identifiers*".

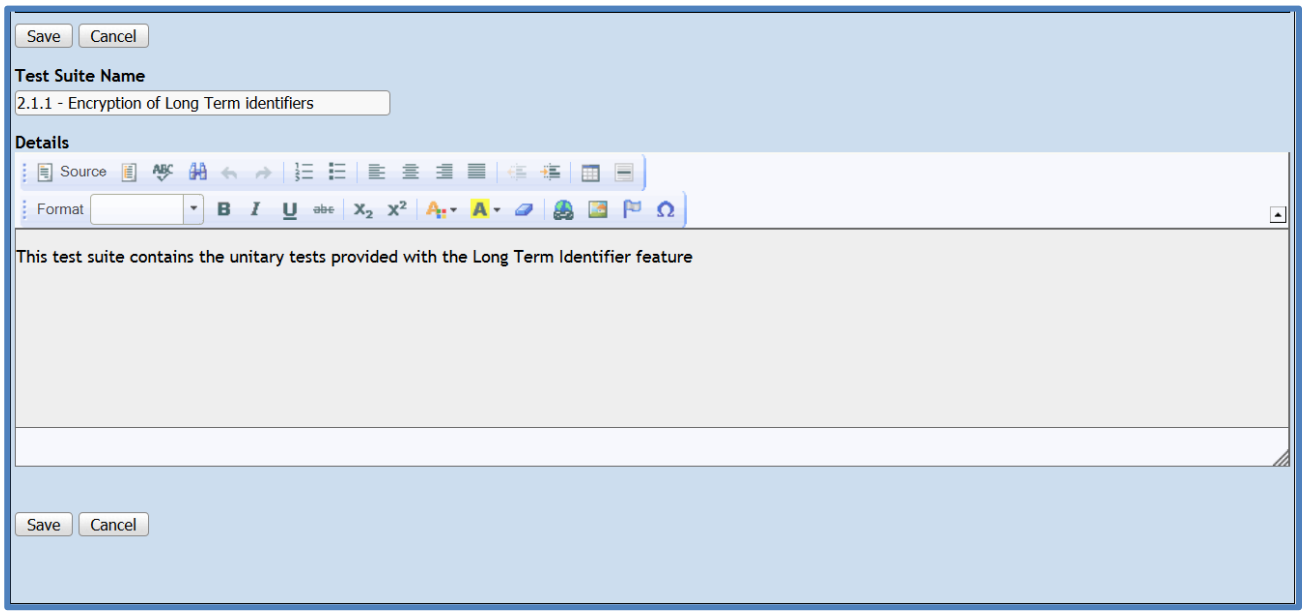

Figure 20: "Encryption of Long term identifiers" feature Test Suite

The following fields are required to create a Test Suite object:

- **Test Suite name**: <Feature_id>-<Feature name>.
- **Details**: A brief description of the scope of the tests cases that will group on the Test Suite.

Note that the Test Suites within TestLink are structured in a way to preserve the organisation of enablers / feature defined by the WP3. In this way some hierarchical sections have been added in order to preserve the classification established by the project. Figure 21 illustrates this organisation inside the tool.

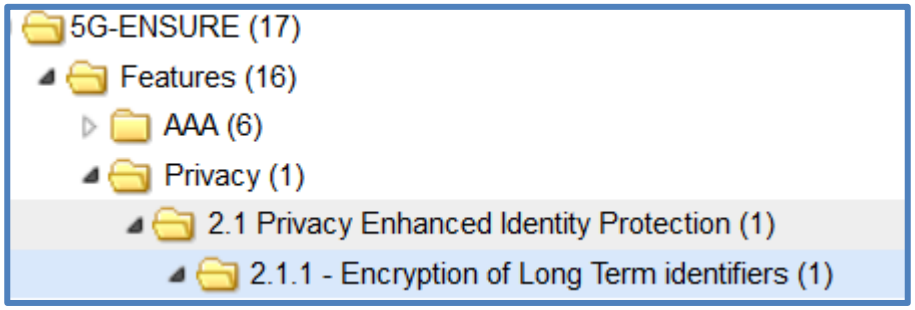

Figure 21: Test cases suites hierarchical organisation

Test case

Once the Test Suite has been defined, it is possible to start adding the test cases inside.

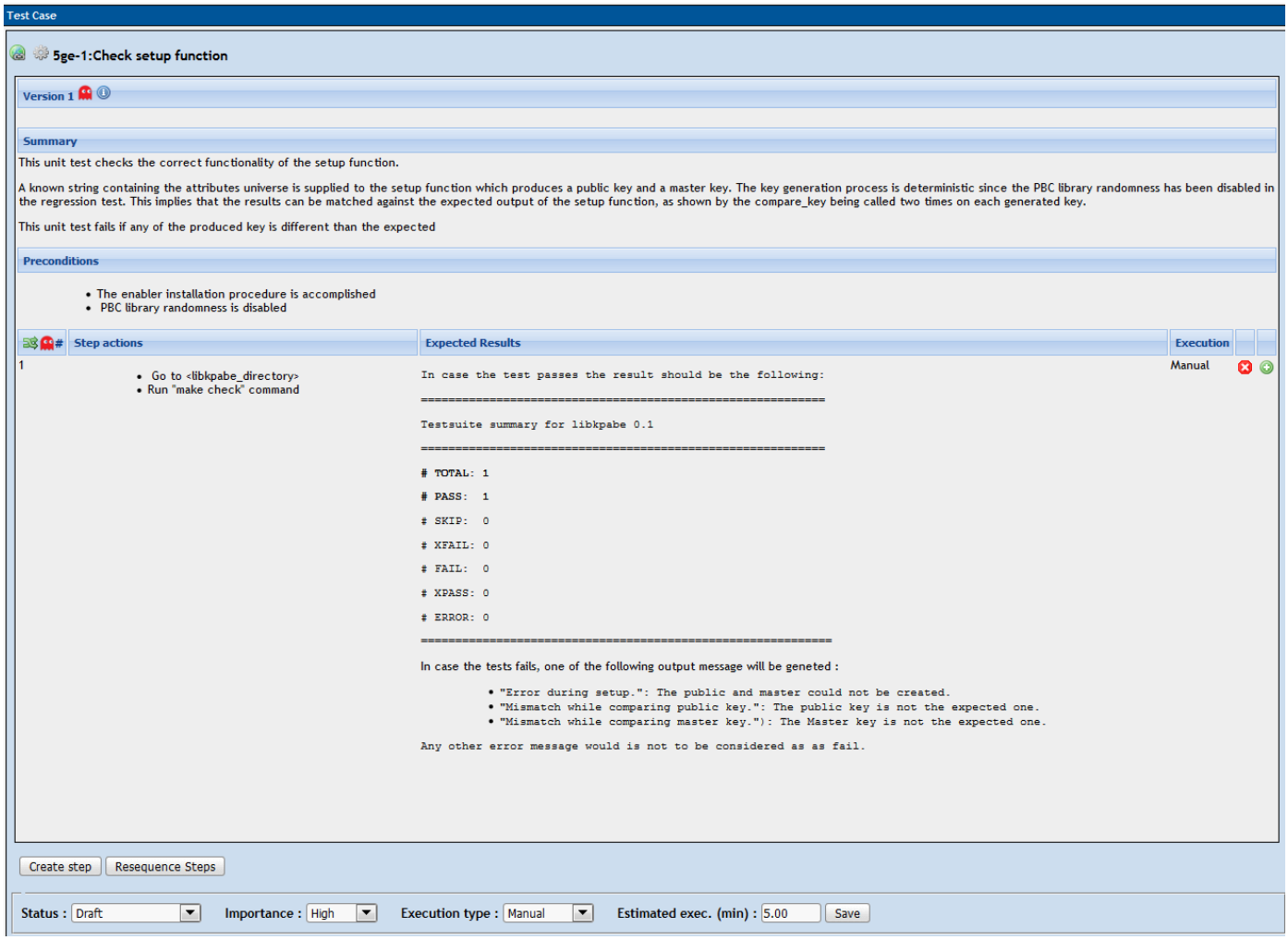

Figure 22 illustrates an example based on the "*Check Function Setup*" unitary test defined in D3.4 [9] for the feature "*Encryption of Long Term Identifiers*".

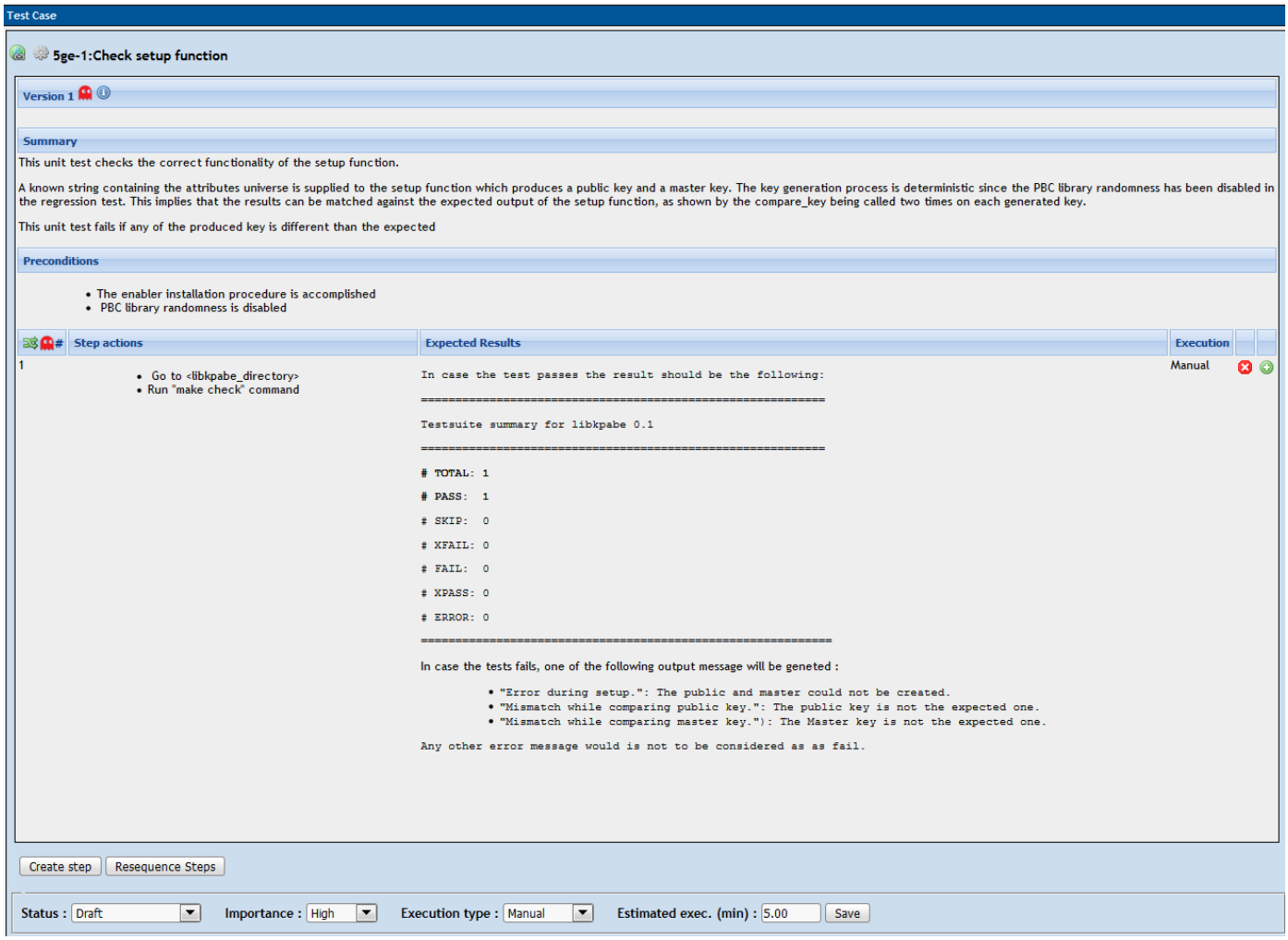

Figure 22: "Check Setup Function" test case

The following fields are required to create a test case object:

- **Test case name**: <Test name>.
- **Summary**: A description of the test scope and the expected results.
- **Preconditions**: This section should address all the requirements needed in order to grant the correct execution of the test case.
- **Steps**: A step by step sequence providing the actions to perform and the expected results for each action.
- **Requirements**: **Link to the feature** for which the test case has been defined.
- **Relations**: It is possible to in relation several test cases if needed.

5.2.2 Enabler security evaluation tests

Figure 23 shows the test plan structure based on the security use cases and their associate threats.

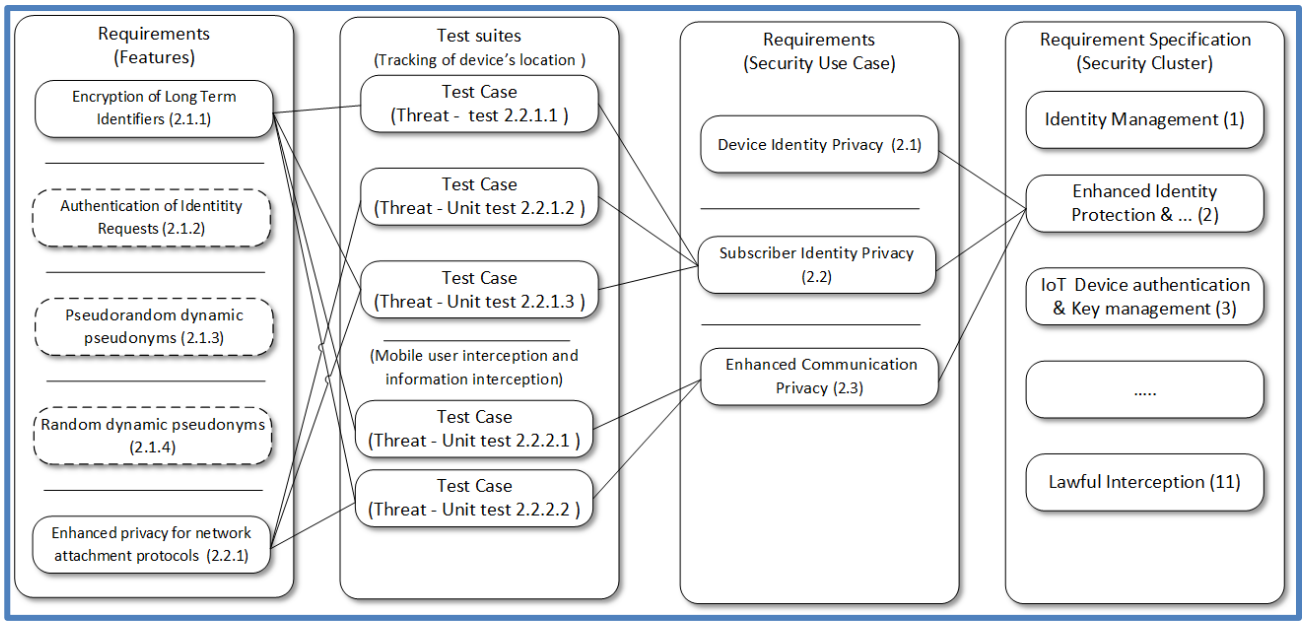

Figure 23: Testbook structure based on security use cases

Figure 23 proposes the structure for security threats "*Tracking of device's location*" and "*Mobile user interception and information interception*" from the requirements of the use case 2.2 "*Subscriber Identity Privacy*" as example. Note that the test cases will also be in relation with the enabler(s) feature(s) requirements that are supposed to tackle the security threat. By doing so, there will be an established relationship between the security use case and the enabler feature, by means of the threat test case.

The rest of the section illustrates how the proposed structure is mapped against TestLink.

Requirement Specification

Figure 24 provides an example of requirement specification for the "*Privacy Enhanced Identity Protection enabler*".

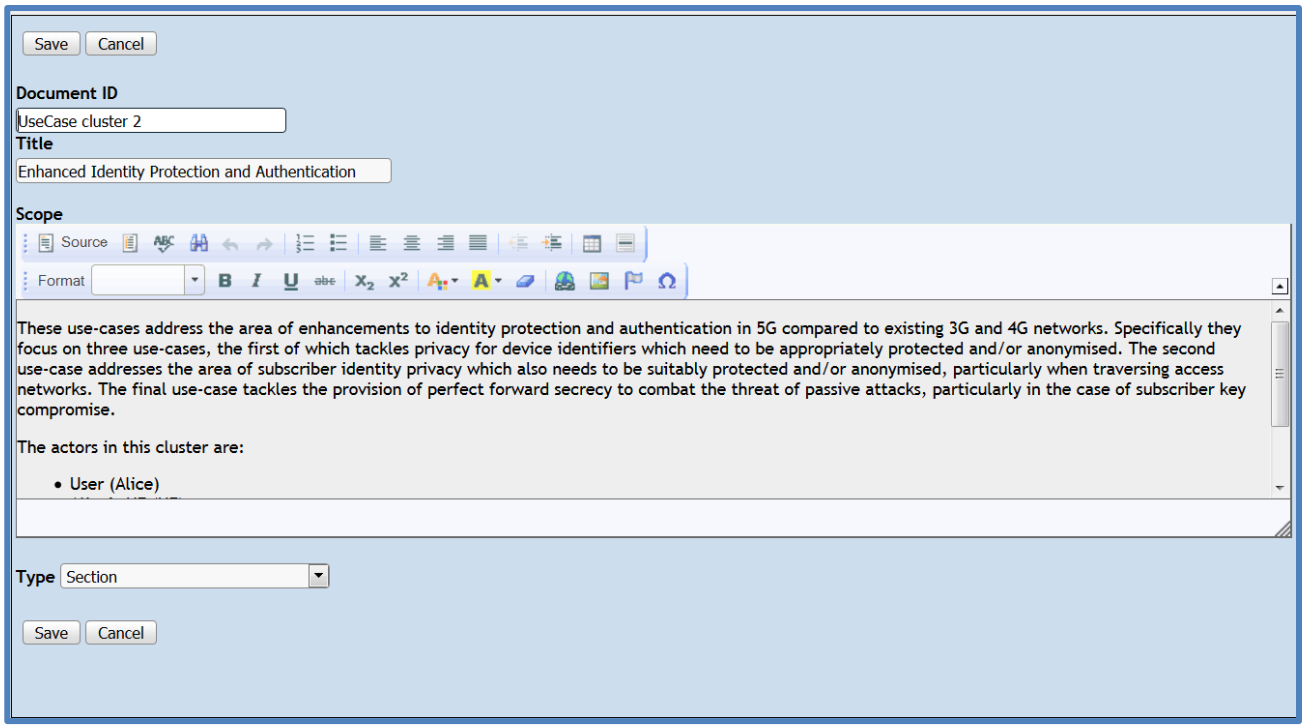

Figure 24: Requirement Specification for "Enhanced Identity Protection and Authentication" security cluster

The following fields are required to create a security cluster object:

- **Document id**: use case cluster<cluster_id>.
  - Where cluster id corresponds to the cluster defined in D2.1 [1]
- **Title:** <use case cluster name>
  - As defined in D2.1 [1]
- **Scope:** A description of the security cluster scope. In the current example, the text has been extracted from the Introduction section of the Security Cluster in D2.1 [1].
- **Type:** Section.
  - Used only to group UCs from the same cluster

Requirement

Once the requirement specification is defined, it is possible to add new requirements inside. Figure 25 provides an example of the use case definition.

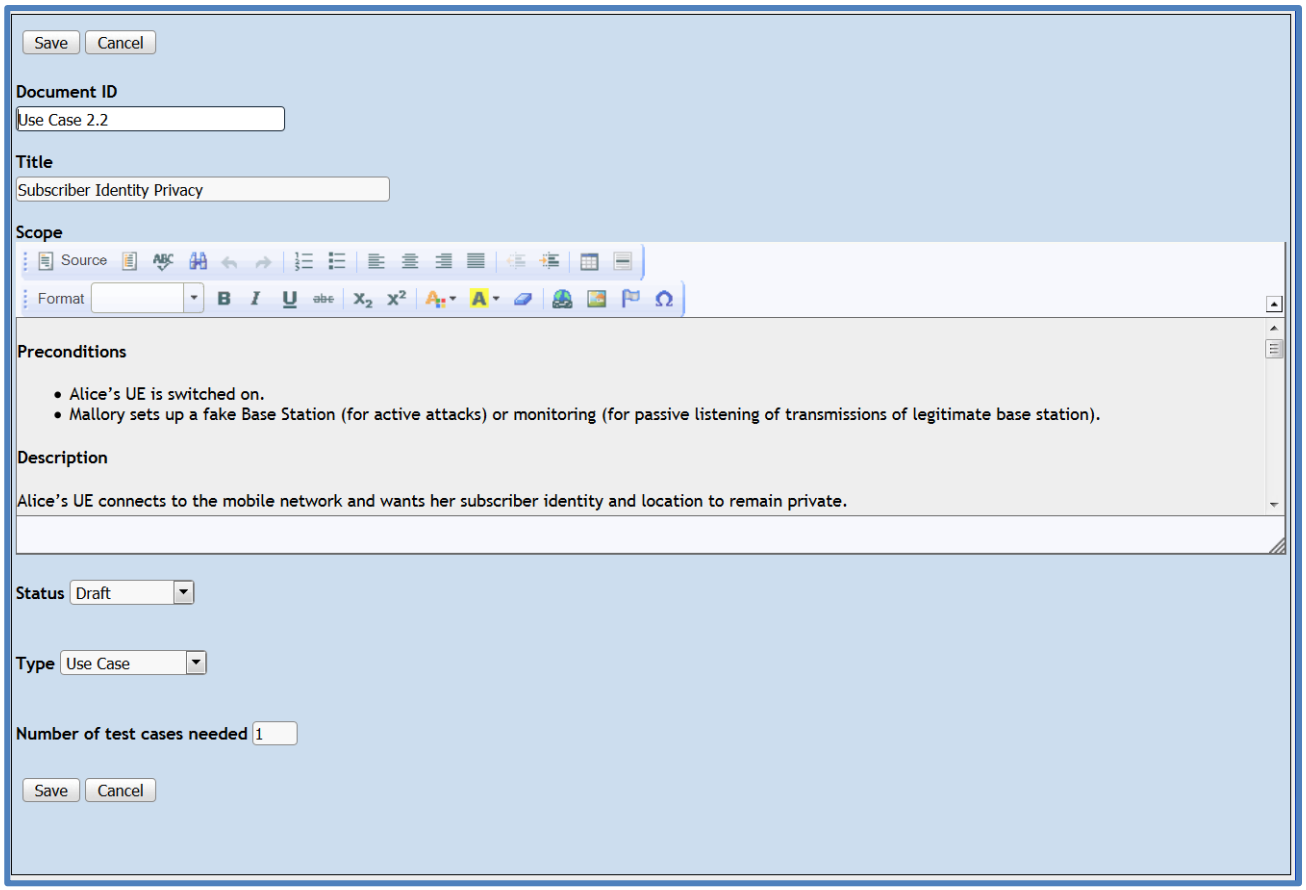

Figure 25: Use case "Subscriber Identity Privacy" requirement

The following fields are required to create a requirement object:

- **Document id**: use case <use case id>.
  - As defined in D2.1 [1].
- **Title**: <use case name>.
  - As defined in D2.1 [1].
- **Scope**: A description of the use case scope. In the current example, the text has been extracted from the use case definition present in D2.1 [1].
- **Type**: use case.

Test Suite

The test suites refers to the threats identified in D2.3 [3]. Figure 26 provides an example of the "*Mobile user interception and information interception*" threat.

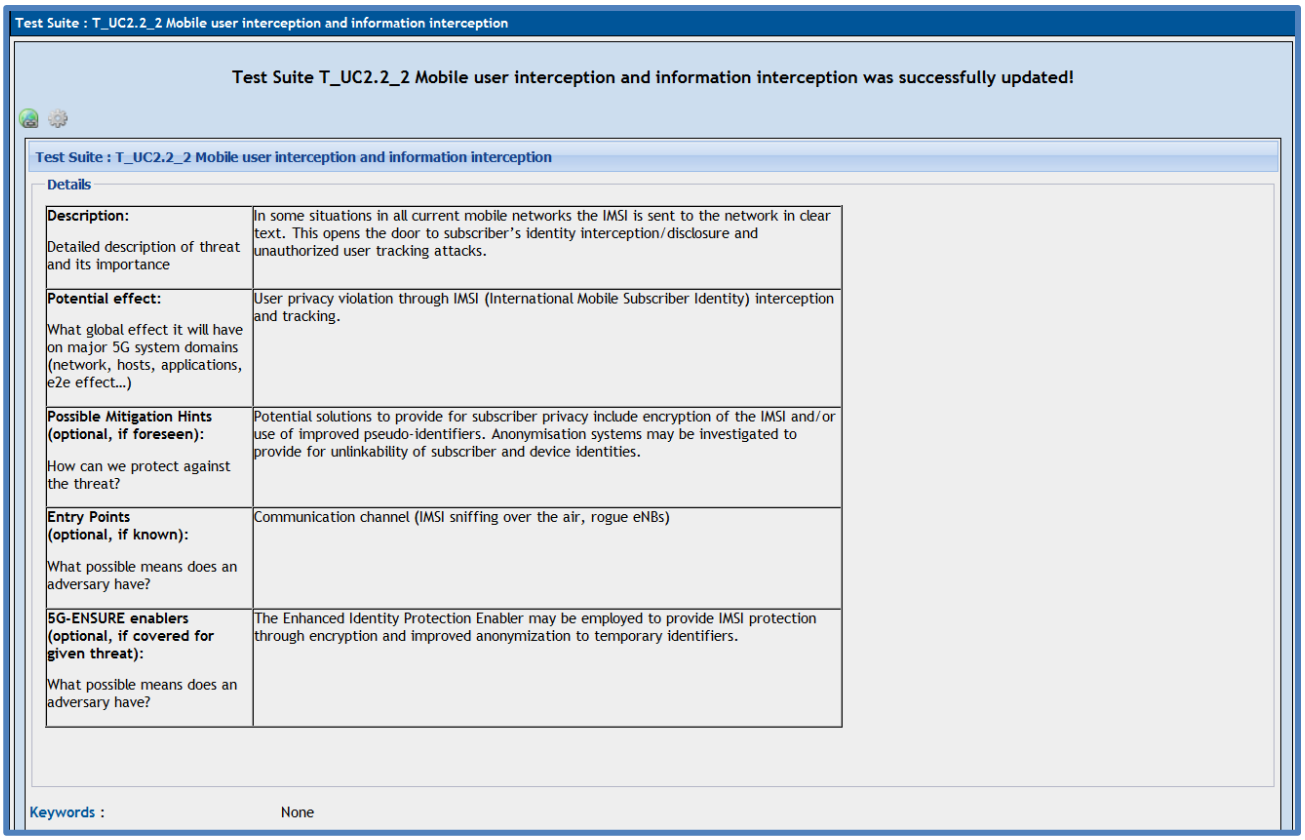

Figure 26: "T_UC2.2_2 Mobile user interception and information interception" threat test suite

The following fields are required to create a test suite object:

- **Test Suite name**: <Threat ID> <Threat name> as specified in D2.3 [3].
- **Details**: The threat description as identified in D2.3 [3]. Only the relevant fields providing a scope for the test cases definition are included.
    - **Description**
    - **Potential effect**
    - **Possible mitigation**
    - **Entry points**
    - **5G-Ensure enablers**

Test case

Once the test suite has been defined, it is possible to start adding the test cases. Test cases, as previously explained, will refer to the threat identified in the test suite. The cases should illustrate the way the enablers developed within 5G-ENSURE will cover the given threat. Figure 27 illustrates an example of what could be a test case under the "*Mobile user interception and information interception" threat,* which is called "*IMSI protection on air interface*". This test case should be tackled by the feature "*Encryption of Long Term Identifiers*"

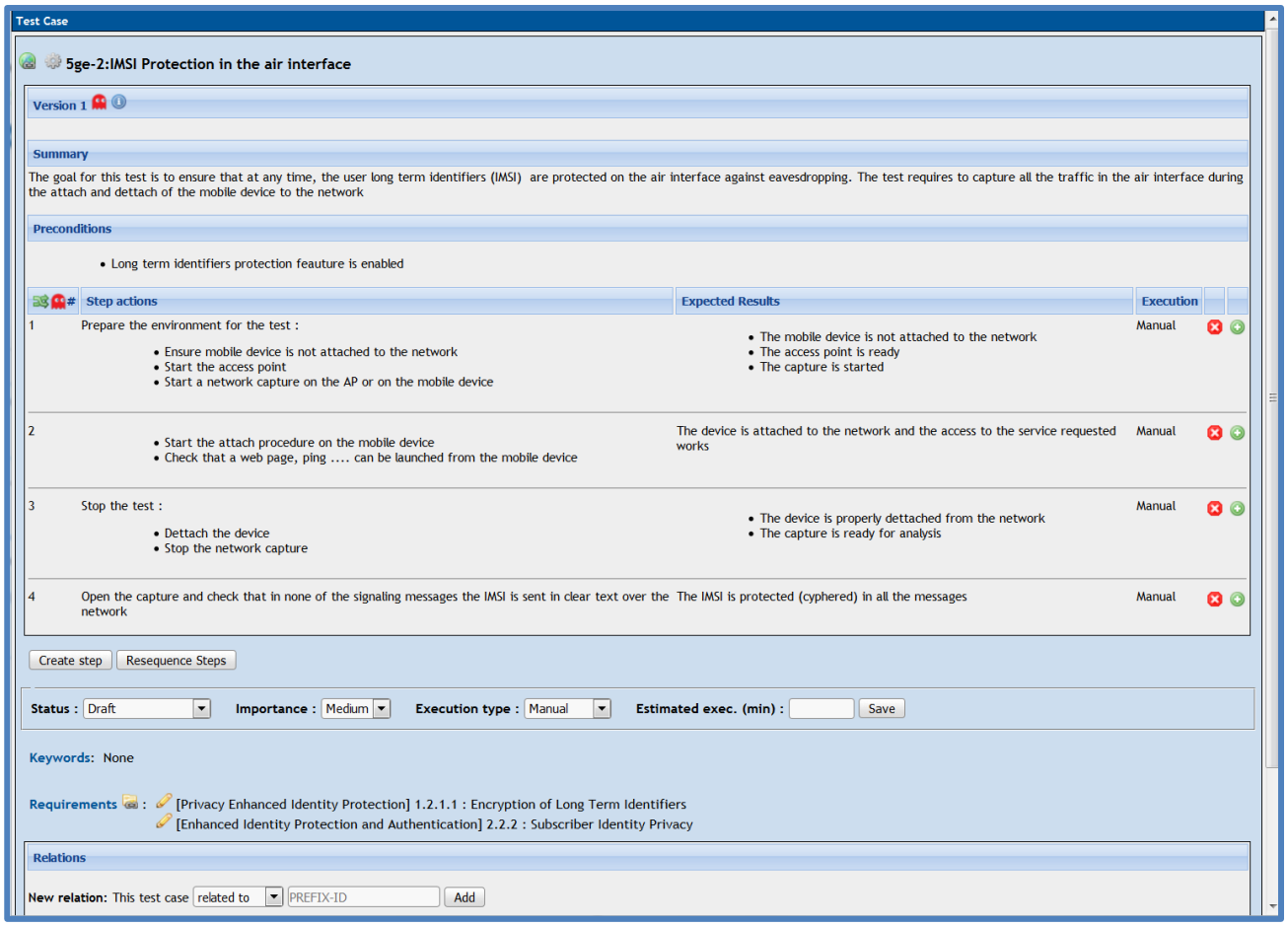

Figure 27: "IMSI protection in the air interface" test case

The following fields are required to create a test case object:

- **Test case name**: <Test name>.
- **Summary**: A description of test scope and the expected results.
- **Preconditions**: This section should address all the requirements needed in order to grant the correct execution of the test case.
- **Steps**: A step by step sequence providing the actions to perform and the expected results for each action.
- **Requirements**: **Link to feature(s) and threat** requirements for which the test case has been defined.
- **Relations**: It is possible to in relation several test cases if needed.

5.3 Test Plan execution planning

TestLink allows to define the way the defined test cases will be executed, namely which test case to execute, which test case version, who executes the test, and the order in which tests are executed. On the other hand, it also makes possible to collect the results from the test case execution and to generate various rapports.

At the time of the writing, it is still premature to identify some of the above mentioned items, because the test cases are not yet ready. Thus this topic will be covered by D4.3 "*Test plan (final): Final description of how to evaluate the selected security enablers*".

5.4 Threat coverage test cases

Test cases will be grouped per security threat, based on enabler's technical tests description with an objective of completeness between UCs and enablers real security coverage. The goal is to describe the list of tests and the expected results.

The Test cases will be written in TestLink. Upon completion, it will possible to extract a report containing the Test cases definition and also the execution results.

The main information a Test case should contain is:

- **Name**: The test case name (as meaningful as possible)
- **Summary**: Describe the test goals (security requirements and UC addressed) and the expected result
- **Pre-conditions**: Provide the requirements that will allow to run the test. One important pre-condition is: The enabler should have successfully passed the unitary tests.
- **Steps**: Describe the test step by step, and provide the expected result for each of the steps
- **Requirements**: It allows one to link the test case with the enabler's features and the use cases. The link between enabler's features and the threats is established in section

The test cases definition will be included on the final version of this document D4.3 "*Test plan (final): Final description of how to evaluate the selected security enablers*") that will be delivered by M18 (April 2017), and the tests execution result will be part of the D4.4 "*Evaluation of the security enablers: Results and analysis of the Testbed runs*" due at M24 (October 2017). Within the scope of the actual version of the document, a reference example of Test case is provided in the previous section.

The following sub-sections provide, based on the test cases described in previous section, an example of the test plan target format. This is the result of the export of test plan from TestLink web tool using its report export tool

5.4.1 Test Suite: Encryption of Long Term identifiers

This test suite contains the unitary tests provided with the Long Term Identifier feature. This is only provided as illustration example (it is not a binding description)

Table 4: Test case – Check setup function

Test Case 5ge-1: Check setup function	
Author:	smorant
Summary: This unit test checks the correct functionality of the setup function. A known string containing the attributes universe is supplied to the setup function which produces a public key and a master key. The key generation process is deterministic since the PBC library randomness has been disabled in the regression test. This implies that the results can be matched against the expected output of the setup function, as shown by the compare_key being called two times on each generated key. This unit test fails if any of the produced key is different than the expected	
Preconditions: • The enabler installation procedure is accomplished	

<table>
<tr><td colspan="3">

- PBC library randomness is disabled

</td></tr>
<tr><td>#:</td><td>Step actions:</td><td>Expected Results:</td></tr>
<tr><td>1</td><td>

- Go to <libkpabe_directory>
- Run "make check" command

</td><td>

In case the test passes the result should be the following:

==========
==========

Testsuite summary for libkpabe 0.1

==========
==========

TOTAL: 1

PASS: 1

SKIP: 0

XFAIL: 0

FAIL: 0

XPASS: 0

ERROR: 0

==========
==========

In case the tests fails, one of the following output message will be generated :

- "Error during setup.": The public and master could not be created.
- "Mismatch while comparing public key.": The public key is not the expected one.
- "Mismatch while comparing master key."): The Master key is not the expected one.

Any other error message would not to be considered as fail.

</td></tr>
<tr><td>Execution type:</td><td colspan="2">Manual</td></tr>
<tr><td>Estimated exec. duration (min):</td><td colspan="2">5.00</td></tr>
</table>

Priority:	High
Requirements	1.2.1.1: Encryption of Long Term Identifiers

5.4.2 Test Suite: T_UC2.2_2 Mobile user interception and information interception

Table 5: Test Suite - T_UC2.2_2 Mobile user interception and information interception

Description: Detailed description of threat and its importance	In some situations in all current mobile networks the IMSI is sent to the network in clear text. This opens the door to subscriber's identity interception/disclosure and unauthorized user tracking attacks.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	User privacy violation through IMSI (International Mobile Subscriber Identity) interception and tracking.
Possible Mitigation Hints (optional, if foreseen): How can we protect against the threat?	Potential solutions to provide for subscriber privacy include encryption of the IMSI and/or use of improved pseudo-identifiers. Anonymisation systems may be investigated to provide for unlinkability of subscriber and device identities.
Entry Points (optional, if known): What possible means does an adversary have?	Communication channel (IMSI sniffing over the air, rogue eNBs)
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	The Enhanced Identity Protection Enabler may be employed to provide IMSI protection through encryption and improved anonymization to temporary identifiers.

Table 6: Test Case – IMSI Protection in the air interface

Test Case 5ge-2: IMSI Protection in the air interface	
Author:	TIIT
Summary: The goal for this test is to ensure that subscriber long term identifiers (IMSI) are encrypted and different values are sent any time they are transmitted over the air interface during the attach procedure to avoid traceability of user locations. The test requires to capture all the traffic in the air interface during the attach and detach of the device to the network.	
Preconditions:	

- Long term identifiers protection feauture is enabled

#:	Step actions:	Expected Results:
1	Prepare the environment for the test : • Install TIIT enabler libkpabe (according to the manual D3.4) on 2 Linux systems (client system and authentication server system) • Run the setup function on the Universe in order to obtain the public key of the crypto system and then run the key generation function for the configured attribute (e.g., SSID1) in order to obtain a private key for the attribute/AP. • Configure the attribute on the client (wpa_supplicant) • Configure the private key and attribute on the server (hostapd). • Get USIMs with known secrets (Ki) or a programmable USIM and an USB card reader. The IMSI, Ki and OPc values shall be known otherwise tool such as pySim (http://cgit.osmocom.org/pysim/) can be used to program the USIM card. • Edit the hostapd runtime configuration file on the server. For each allowed user add a line in the EAP-AKA milenage file (used by the hlr_auc_gw component) matching the following format: <IMSI> <Ki> <OPc> <AMF> <SQN>, with AMF and SQN parameters set respectively to 0000 and 000000000000) • Start a network capture on the AP or on the mobile device	The mobile device is not attached to the network The access point is ready The capture is started
2	• Insert the SIM card in the smart card reader • Start the attach procedure connect the device to the SSID1 WiFi network with EAP-AKA full authentication method.	The device is attached to the network and the access to the service requested works. Verify from the pcap trace that the IMSI value in Identity Response messages are encrypted
3	• Disconnect the device	The device is properly disconnected from the network.
4	• Connect again the device to the SSID1 WiFi network with EAP-AKA full authentication.	Verify from the pcap trace that the IMSI value in Identity Response messages are encrypted and different from the value observed in step 2.
5	• Repeat step 3 and 4 a desired number of times	Verify from the pcap trace that the IMSI value in Identity Response messages are encrypted and different each time.
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	

Requirements	1.2.1.1: Encryption of Long Term Identifiers 2.2.2: Subscriber Identity Privacy

6 Conclusions

This document provides the required procedures to evaluate the enablers' features in the testbed. It provides the test plan structures and some test case examples. This deliverable will be completed and will take its final form in D4.3 "Test plan (final): Final description of how to evaluate the selected security enablers" (M18). The evaluation results from the test plan execution and the result analysis will be provided at the end of the project (M24) on the D4.4 "Evaluation of the security enablers: Results and analysis of the Testbed runs".

The 5G-ENSURE enablers R1 are now under integration phase in the Testbed. The enabler's unitary tests are under adaptation for execution on the testbed environment using TestLink, and the evaluation tests for enablers will have to be defined and jointly validated as Scenarios by both WP2 and WP4. The list of validated evaluation Scenarios for each of the enabler will be delivered in an external document.

The work reported in this document was performed in close technical collaborations with WP2 and WP3, and based on all other technical deliverables already produced by the project. In particular with an analysis of Enabler's security claims described in D3.2 [2] of each enabler's feature, but also against the different use cases defined in D2.1 [1], and their associated security threats identified in D2.3 [3]). Finally, this document delivers the consolidated list of threats coverage by the R1 Enabler features.

References

[1] 5G-ENSURE, "5G-ENSURE D2.1 Uses Cases," [Online]. Available: http://5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.1-UseCases.pdf.

[2] 5G-ENSURE, "5G-ENSURE D3.2 5G-PPP security enablers open specifications (v1.0)," [Online]. Available: http://5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D3.2-5G-PPPSecurityEnablersOpenSpecifications_v1.0.pdf.

[3] 5G-ENSURE, "5G-ENSURE D2.3 Risk assessment, mitigation and requirements (draft)," [Online]. Available: http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.3-RiskAssessmentMitigationRequirements.pdf.

[4] 5G-ENSURE, "5G-ENSURE D4.1 5G Security testbed architecture," [Online]. Available: http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D4.1-5G_Security_testbed_architecture_v1.0.pdf.

[5] 5G-ENSURE, "5G-ENSURE D3.1 5G-PPP Security Enablers Technical Roadmap early vision," [Online]. Available: http://5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D3.1-5G-PPPSecurityEnablersTechnicalRoadmap_early_vision.pdf.

[6] "TestLink home page," [Online]. Available: http://testlink.org/.

[7] Ansible, "Ansible home page," [Online]. Available: https://www.ansible.com/.

[8] "Artifactory home page," [Online]. Available: https://www.jfrog.com/confluence/display/RTF/Welcome+to+Artifactory.

[9] 5G-ENSURE, "5G-ENSURE D3.4 5G-PPP_Security_Enablers_Documentation," [Online]. Available: http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D3.4_5G-PPP_Security_Enablers_Documentation.pdf.

[10] "Testlink user manual," [Online]. Available: https://wiki.openoffice.org/w/images/1/1b/Testlink_user_manual.pdf.

[11] "TestLink Screencast," [Online]. Available: https://www.youtube.com/watch?v=6s48WGuX2WE.

[12] W. Rudin, Functional Analysis, McGraw-Hill, 1973.

[13] 5G-ENSURE, "5G-ENSURE D2.2 Trust Model (draft)," [Online]. Available: http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.2-TrustModel.pdf.

[14] "Artifactory as a Debian repository," [Online]. Available: https://www.jfrog.com/video/setting-up-artifactory-4-as-a-debian-repository-in-minutes/.

[15] “Artifactory as a YUM repository,” [Online]. Available: https://www.jfrog.com/video/artifactory-yum-repository/.

[16] “Artifactory as a Docker registry,” [Online]. Available: https://www.jfrog.com/video/install-artifactory-docker-registry-one-minute-less/.

[17] “Artifactury user manual,” [Online]. Available: https://www.jfrog.com/confluence/display/RTF/Welcome+to+Artifactory.

[18] 5G-ENSURE, “5G-ENSURE D3.1 5G-PPP Security Enablers Technical Roadmap early vision,” [Online]. Available: http://5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D3.1-5G-PPPSecurityEnablersTechnicalRoadmap_early_vision.pdf.

[19] 5G-ENSURE, “5G-ENSURE D2.1 Uses Cases,” [Online]. Available: http://5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.1-UseCases.pdf.

[20] 5G-ENSURE, “5G-ENSURE D3.2 5G-PPP Security Enablers Open Specifications,” [Online]. Available: http://5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D3.2-5G-PPPSecurityEnablersOpenSpecifications_v1.0.pdf.

[21] “KVM4FV,” [Online]. Available: http://artifacts.opnfv.org/kvmfornfv/docs/all/all.pdf.

[22] “IPSecS2S vpn template,” [Online]. Available: https://workspace.vtt.fi/sites/5g-ensure/Shared%20Documents/Workpackages/WP4/T4.1/Testbed/Nodes_interconnection/IPsecS2S-vpn-template.docx.

[23] “Open Air Interface,” [Online]. Available: http://www.openairinterface.org/.

[24] A. Diez, “Understanding NFV Management and Orchestration,” 2015.

[25] “ETSI NFV home page,” [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/nfv.

[26] “RHEL Virtualisation KVM timing management,” [Online]. Available: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Virtualization_Deployment_and_Administration_Guide/chap-KVM_guest_timing_management.html.

[27] “Cisco Anyconnect,” [Online]. Available: http://www.cisco.com/c/en/us/support/security/anyconnect-secure-mobility-client/tsd-products-support-series-home.html.

[28] “IPerf home page,” [Online]. Available: https://iperf.fr/.

[29] “OpenEPC Home Page,” [Online]. Available: http://www.openepc.com.

[30] VTT, “QoSMet home page,” [Online]. Available: http://www.vttresearch.com/qosmet.

[31] Wikipedia, “Wikipedia - Orchestration,” [Online]. Available: https://en.wikipedia.org/wiki/Orchestration_(computing).

[32] Wikipedia, “Wikipedia - Blackbox definition,” [Online]. Available: https://en.wikipedia.org/wiki/Black_box.

[33] B. Dictionary, “Businees Dictionary - White box,” [Online]. Available: http://www.businessdictionary.com/definition/white-box.html.

[34] 5GNorma, “5G Norma D3.1 Functional network architecture and security requirements,” [Online]. Available: https://5gnorma.5g-ppp.eu/wp-content/uploads/2016/01/5G_NORMA_D3.1.pdf.

[35] 5GNorma, “5G Norma D2.1 Use cases, scenarios and requirements,” [Online]. Available: https://5gnorma.5g-ppp.eu/wp-content/uploads/2015/11/5G-NORMA_D2.1.pdf.

[36] 5G-PPP, “5G-PPP 5G Architecture White Paper,” [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-5G-Architecture-WP-For-public-consultation.pdf.

A Testbed Terms of Use (Provisional)

Foreword

The present document (hereinafter referred to as "**Terms of Use**") describes the terms of use and participation to the Testbed provided by Testbed Owners to the other participants in the 5G-ENSURE Project, as per the work packages 3 and 4 of Annex 1 of the Grant Agreement n°671562. The Project participants that have signed the present Terms of Use will hereinafter, jointly or individually, be referred to as "**Parties**" or "**Party**".

The mission of the 5G-ENSURE Testbed is to develop and test a set of useful and usable security enablers for 5G for the implementation of the Project (hereinafter referred to as "**Purpose**"). Participation to the 5G-ENSURE Testbed is subject to the rules of the Grant Agreement n°671562 and the Consortium Agreement, completed by the present Terms of Use's rules.

Two copies of the signed Terms of Use have to be submitted to both Project Manager and Technical Manager by any participant who wishes to be involved in Testbed activities.

Disclaimer

The EC flag in this deliverable is owned by the European Commission and the 5G PPP logo is owned by the 5G PPP initiative. The use of the flag and the 5G PPP logo reflects that 5G-ENSURE receives funding from the European Commission, integrated in its 5G PPP initiative. Apart from this, the European Commission or the 5G PPP initiative have no responsibility for the content.

Copyright notice

Definitions

As used herein and throughout these Terms of Use, the following capitalized terms, in singular or plural, shall have the meanings respectively ascribed to them below:

1. **Consortium Agreement**: the 5G-ENSURE Consortium Agreement executed on the 1st November 2015 by the Parties and the other participants of the Project as part of the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n°671562.
2. **Input**: all data sets, security enablers, tools & methodologies or any other test materials input in the Testbed by the Parties.
3. **Output**: all results of the Parties' use and tests conducted on the Testbed.
4. **Project**: the research project "5G Enablers for Network and System Security and Resilience", which is part of the European Union's Horizon 2020 research and innovation programme and is governed by the Grant Agreement n°671562 executed on July 28 2015, and by the Consortium Agreement.
5. **Testbed**: the 5G-ENSURE security Testbed object of these Terms of Use, implemented by the Parties as in scope of the Work Package WP4 of Annex 1 of the Grant Agreement n°671562.
6. **(Testbed) Node**: set of hardware and software resources provided and operated by some Parties for the other Parties for the implementation of the Project. For the sake of clarity, a Testbed Owner's corporate infrastructure which is interconnected with the Testbed remains outside of the scope of these Terms of Use, and no Party except that Testbed Owner is authorized to access it.
7. **Testbed User**: a Party requiring to run activities on or have access to the 5G-ENSURE Testbed.
8. **Testbed Owner**: a Party providing one of the Nodes hosting the 5G-ENSURE security Testbed.
9. **Testbed Operator**: a Party managing enablers' deployment within Testbed and the operational status of the infrastructure. It is composed of Testbed Owner technical operators.
10. **Enabler Owner**: a Party owning one or more of the 5G security enablers produced in the course of the Project.

Other capitalized words of the Terms of Use that are not defined in the present Terms of Use shall have the meaning attributed to them in the Consortium Agreement or, if not defined in the Consortium Agreement, in the Grant Agreement n°671562.

Technical undertakings

Use

Any involvement in the Testbed entails the following commitments:

- Limited use: The Parties may only use the Testbed in accordance with the Purpose. The Testbed Operators have the possibility to restrict the access of a Party to resources if the Testbed is misused by that Party.

- Property rights: A Party shall not incorporate in the Testbed or use any information or intellectual property rights that are owned by a third party, unless that Party has first secured a right to do so.

- Input inventory: Any Party that considers providing Input to the Testbed has to send the Parties a written comprehensive description of the Input considered, of its characteristics and of how to use it (i.e. for enablers this includes the software release and the documentation including Installation and Administration Guides; User and Programmers Guides; Unit Testing Plan as well the Unit Tests Report to certify that the enabler successfully passed the sanity checks). Once an Input has been validated by Technical Project Manager and the Work Package 3 and 4 Leaders and assigned to the Testbed, this Input cannot be removed without the prior approval of Technical Project Manager. Input shall not contain any personal data.

- Configuration: The configuration of the Testbed is implemented by the Testbed Operators in accordance with the agreed test plan between Enabler Owners and Testbed Operators. All Testbed Users shall be given by the Testbed Operators the possibility to check that the required configuration is effective (right to access and read the configuration).

- Additional documentation: Documentation (such as user manuals or detailed interface descriptions) about the Testbed, restricted to essential information required to prepare and execute the tests, can be provided by Testbed Owners to Testbed Users which ask for it during the setup phase.

- Login credentials: A Party's login credentials for connection to the Testbed are provided by Testbed Operators on a confidential basis and may not be disclosed to anyone. The login credentials are valid for the project duration.

- Planning: Any tests and experiments must be planned in advance during time periods agreed by the involved Testbed Operator, in order for the latter to book the resources required by the test session.

Nodes interconnection

Testbed Owners may interconnect their own remote Testbed Nodes to the 5G-ENSURE Testbed subject to the following rules:

- Nodes inventory: Any Testbed Owner that considers providing Nodes to the Testbed has to send the Parties a written comprehensive description of the Node, the manpower that it will commit to operating the Node, and the elements considered, of their characteristics and of how to use them. It is the responsibility of the 5G-ENSURE Steering Committee to validate and manage any change in the Testbed Nodes allocation. Once assigned to the Testbed, Nodes cannot be removed without the prior approval of Technical Project Manager.

- Accurate development / availability of resources: The Testbed Owners must do their best efforts to ensure that their Nodes operate properly that they are reliable and secure, and that they are available in the timetable that they provided to the Parties.
- Data update cycle: The Testbed Owners must do their best efforts to provide Nodes that are coherent, completely operative and regularly updated.
- Audit procedure: Technical Project Manager may choose to conduct an audit procedure to observe and inspect the controls, compliance, performance, etc. of a Testbed Owner's Node and Input. All Testbed Owners are required to comply with any audit procedure and any recommended corrective actions that Technical Project Manager may come to suggest.

Access and support

- Access: The Testbed Operators undertake to do their best efforts to provide a reliable remote access to the Testbed from a Party's remote facility, and to provide inter-connectivity with Testbed Owners' Nodes, as well as comply with all their obligations derived from the Grant Agreement n°671562 and the Consortium Agreement. Testbed Operators cannot guarantee however a 24/7 availability of the Testbed, especially as the operation of all the Testbed's Nodes relies on several different entities at the same time.
- Support: The Testbed Owners and the Testbed Operators undertake to do their best efforts to provide the other Parties with assistance in case of technical problems in relation to the Testbed.
    - Problem severity is classified in three levels to be used for notification of Testbed Owners and Operators in Problem Report (PR):
        - Critical: A default is critical when it leads to the inoperability of the service and no fallback or workaround solution is available.
        - Major: A default is major when it leads to a limitation of the functionalities or the performances of the service, or to the necessity to use fallbacks mechanisms or workarounds.
        - Minor: A default is minor when it has no operational impact but leads to difficulties to operate the service.
    - In order for Parties to access the helpdesk and submit Problem Reports, b<>com provides an online web portal accessible at: https://helpdesk.b-secure.irt-b-com.org.
    - Testbed Owners' and Operators' operation staff is on duty 5 days / week (week-ends, bank holidays and days-off not included), at 9:30-12:00 – 14:00-17:30 CET/CEST Time
    - If a Testbed Owner or a Testbed Operator is not able to fix a notified issue related to one of the Nodes it provides or operate in ten (10) calendar days, it has to warn Technical Project Manager and the other Parties on a timely basis about the impact of the issue on the

Testbed. After this period and if the issue is critical, the Parties can decide to refer the issue to Technical Project Manager.

- Generally, unless Open Source is used, the source code of Testbed software is not accessible to the user. An option is to provide accessibility of the user to this source code if it is required for execution of a testing project.
- As an option, direct configuration by the enabler owner may be authorized by Testbed owner for specific purpose (access rights to write the configuration).

Confidentiality

Confidential information exchanged in the context of the present Terms of Use remains subject to the confidentiality obligations of the 5G-ENSURE Consortium Agreement. Consequently, all information in whatever form or mode of communication, which is disclosed by a Party or another participant in the 5G-ENSURE Project (the "**Disclosing Party**") to any other Party or participant in the 5G-ENSURE Project (the "**Recipient**") in connection with the Project or with the Testbed, and which is marked or otherwise identified as confidential at or prior to the time of disclosure, or which nature would give a reasonable person to understand it to be confidential at the time of disclosure, is "**Confidential Information**".

For the sake of clarity, Confidential Information shall be deemed to include all Input and all Output of the Testbed, as well as all information provided by Testbed Owners and Operators in order to enable the Parties to operate and use the Testbed.

As stipulated in the Consortium Agreement, the confidentiality obligations undertaken by the Parties for the Project shall remain in force during the Project and for a period of four (4) years after the end of the Project.

Intellectual property & Access Rights

The Parties' intellectual property and Confidential Information (be it Background, Input, Output, Results, etc.) that is used in the context of the present Terms of Use remain subject to the relevant obligations of the 5G-ENSURE Consortium Agreement.

Consequently, Output shall be owned by the Party who generated it.

For the sake of clarity, when a Testbed User carries out a test by itself, with its own Input, and if the Testbed is simply put at its disposal by the Testbed Owners without further operation, the Testbed User shall remain the owner of the Output that it generates.

Two or more Parties shall own Output jointly if:

- They have jointly generated the Output in question; and
- It is not possible to:

- establish the respective contribution of each Party; or
- separate each Party's part of the Output for the purpose of applying for, obtaining or maintaining protection.

Each joint owner shall have an equal, undivided interest in and to a joint Output as well as in and to resulting Intellectual Property Rights in all countries. Each of the joint owners and their Affiliated Entities shall be entitled to exploit the jointly owned Output as they see fit, and shall be entitled to grant non-exclusive licenses, without obtaining any consent from, paying compensation to, or otherwise accounting to any other joint owner(s).

For the sake of clarity, Access Rights applicable to Input and Output in accordance with the 5G-ENSURE Consortium Agreement are the following:

- Access Rights to Input and Output of the Testbed needed for the implementation of the Project are hereby requested, and shall be deemed granted, as of the Effective Date, on a royalty-free basis to and by all Parties, and shall either terminate upon completion of the Project or upon termination of a Party's participation to the Testbed.
- Access Rights to Output of the Testbed needed for internal research, development and teaching are hereby requested and shall be deemed granted as of the date of the Output arising, on a royalty-free basis to and by all Parties.
- Access Rights to Output of the Testbed needed for any other Exploitation (including as needed for Use of a Party's own Results) shall be granted on Fair and Reasonable Conditions subject to the conditions of the Consortium Agreement.

Duration

The present Terms of Use shall have effect from 1st of August 2016 ("**Effective Date**"). The Terms of Use's obligations will remain in force for the duration of the Project or, where relevant, the duration assigned to them by the Grant Agreement n°671562 and then the Consortium Agreement.

****

The undersigned hereby acknowledges that it has read and understood the present Terms of Use and agrees to be bound by all their rules as well as by the other rules applicable to the 5G-ENSURE Project (i.e. Grant Agreement n°671562 and Consortium Agreement).

Party: ……..

_____

By: ……..

Title: ……..

Date: ……..

B Proposed structure for evaluation Scenario description

This content is under investigation within the work packages WP2, WP3 and WP4. It is delivered as a draft, and it will be finalized in a specific document by end of 2016.

For the pair (enabler feature, threat), the Enabler Owner (E.O.) delivers:

- The relations and differences between use cases, attacks, threats and the proposed evaluation Scenario.
- Clear explanation on how the E.O. interprets the threat(s) to be covered by the enabler feature (threat objectives interpretation)
- Motivation of the chosen techniques, algorithms, heuristics to cover the threat. There is the need of adding reference of publication supporting evidence
- TestLink based description of its Scenario, which:
    - Relates the steps of the Scenario to the threat interpretation.
    - Explains the choice (and motivation) of attacks (real or simulated) and the strategy of evidence for the coverage
    - Describes the strategy in a step by step manner
    - Identifies the required preconditions of each step
    - Describes the attack scenario proposed in a step by step manner
    - Describes what are the evidences of threat coverage (related to threat interpretation) in a step by step manner
    - Describes the enabler feature's property used and why it is an evidence of threat coverage (and potential links with other threats)

WP2 will review and validate the Scenario proposed based on TestLink input.

Based on this WP2 validation and the scope of the test Scenario defined in Testlink, WP4 proceed to the following steps:

- Establish if the proposed preconditions, step and attacks are feasible and reproducible on the testbed.
- Identify the building blocks required to describe / operate the test
- Validate the technical strategy to be implemented over the testbed.

After WP4 validation, it is then possible to proceed to an Evaluation of the pair (enabler feature, threat).

A specific workflow to monitor this procedure will be required. It is proposed to use the HelpDesk facilities for this monitoring).