Grant Agreement No.: 952697
Call: H2020-SU-ICT-2018-2020
Topic: SU-ICT-02-2020
Type of action: RIA

ASSURED

D6.1 Evaluation Framework and Demonstrators Planning

Revision: v.1.00

Work package	WP 6
Task	Task 6.1
Due date	28/02/2022
Deliverable lead	SPH
Version	1.00
Authors	Nikolaos Drosos (SPH)
Reviewers	Stefanos Venios (S5), Thanassis Giannetsos (UBITECH)
Abstract	D6.1 defines the framework for the evaluation of the overall ASSURED framework in the context of the four envisaged reference scenarios. The focus is on the definition of the exact user stories to be considered per scenario including detailed sequence diagrams between the ASSURED components to be evaluated. The four use cases foresee the evaluation of specific functionalities of the ASSURED framework in different application domains, namely Smart Manufacturing, Smart Aerospace, Smart Cities and Smart Satellites. More specifically, the Smart Manufacturing reference scenario focuses on the operational assurance of the deployed edge devices and data aggregators in an ecosystem with strict time constraints, thus, the execution of attestation enablers should not impact the performance of other computational functionalities of the devices; the Smart Aerospace reference scenario focuses on the need for remote SW updates, of the devices comprising the aircraft, coming from secure and authenticate sources; the Smart Cities reference scenario focuses on the strict user privacy issues that need to be met and the requirement for role-based access control to groups of stakeholders requesting access to specific operational and attestation data recorded on the Blockchain; and, the Smart Satellite reference scenario focuses on monitoring and the establishment of trust between satellites and the Ground Station that are part of the same safety-critical mission
Keywords	Evaluation Framework, Demonstrators Setup, Scenarios, KPIs, Test Cases

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	15.12.2021	Table of Contents provided and Template files formulated for collecting the sequence diagrams, user confirmation stories, list of ASSURED components leveraged, and workflows, per use case	Nikos Drosos (SPH)
V0.2	07.01.2022	Definition of the evaluation methodology to be adopted in the context of ASSURED for testing the various functionalities in the context of the different use cases (Chapter 2)	Sotiris Kousouris, Stefanos Venios (S5) Ioannis Avramidis (INTRA) Ilias Aliferis (UNIS)
V0.25	21.01.2022	First iteration of the complete description of the user stories for the Smart Aerospace Use Case (Chapter 5)	Stelios Basagiannis, Riccardo Orizio (UTRCI) Sotiris Kousouris, Stefanos Venios (S5)
V0.3	28.01.2022	First iteration of the complete description of the user stories for the Smart Satellites Use Case (Chapter 6)	Nikos Drosos (SPH) Richard Mitev, Philip Rieger (TUDA)
V0.4	11.02.2022	First iteration of the complete description of the user stories for the Smart Cities Use Case (Chapter 4)	Dimitra Tsakanika, Ilia Christantoni (DAEM) Dimitris Papamartzivanos, Thanassis Giannetsos (UBITECH)
V0.5	18.02.2022	First iteration of the complete description of the user stories for the Smart Manufacturing Use Case (Chapter 3)	Sotiris Kousouris, Stefanos Venios (S5) Karthik Shenoy, Shantanoo Desai (BIBA) Nikos Drosos (SPH) Dimitra Tsakanika, Ilia Christantoni (DAEM) Stelios Basagiannis, Riccardo Orizio (UTRCI) Dimitris Papamartzivanos, Thanassis Giannetsos (UBITECH)
V0.6	11.03.2022	Definition of the generic test cases to be conducted in all use cases for testing the common ASSURED functionalities (Chapter 7)	Liqun Chen, Nada El Kassem (SURREY) Benjamin Larsen (DTU) Ioannis Avramidis (INTRA) Ilias Aliferis (UNIS) Dimitra Tsakanika, Ilia Christantoni (DAEM) Stelios Basagiannis, Riccardo Orizio (UTRCI) Dimitris Papamartzivanos, Thanassis Giannetsos (UBITECH)
V0.62	25.03.2022	Correction and update of the sequence diagrams to better depict some user stories based on the progress performed in the integration of the overall ASSURED framework (Chapters 3, 4, 5, and 6)	Karthik Shenoy, Shantanoo Desai (BIBA) Nikos Drosos (SPH) Dimitra Tsakanika, Ilia Christantoni (DAEM) Stelios Basagiannis, Riccardo Orizio (UTRCI) Dimitris Papamartzivanos, Thanassis Giannetsos (UBITECH)
V0.7	08.04.2022	Peer review by UBI	Stefanos Venios (S5), Thanassis Giannetsos (UBITECH)
V1.00	15.04.2022	Finalization of the document	Nikos Drosos (SPH), Dimitris Papamartzivanos (UBITECH)

DISCLAIMER

COPYRIGHT NOTICE

Project co-funded by the European Commission in the H2020 Programme		
Nature of the deliverable:	**R**	
Dissemination Level		
PU	Public, fully open, e.g. web	✔
CL	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to ASSURED project and Commission Services	

* R: Document, report (excluding the periodic and final reports)

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

OTHER: Software, technical diagram, etc.

EXECUTIVE SUMMARY

D6.1 defines the evaluation framework for the technical and business validation of the ASSURED platform along with the detailed scenarios to take place for the execution of each one of the four demonstrators that will be used – namely *Smart Manufacturing*, *Smart Cities*, *Smart Aerospace* and *Digital Security of Smart Satellite Communications*.

In this context, it provides a detailed view of ASSURED's evaluation framework which aims to verify that the platform implementation is aligned with the requirements defined and it provides the expected benefits to its stakeholders. The approach to be followed for both technical and business evaluation is presented, and specific metrics are detailed including in total **36 KPIs**.

In addition, the demonstrations to take place per use case are detailed in a dedicated chapter per use case. In those chapters, a short summary is provided, and the main challenges are enumerated per use case. The defined planning is presented, documenting the scenarios to be demonstrated per release (1st and 2nd release). A total of **23 User Stories** are detailed, to be demonstrated through the four use cases; for each one of them a set of scenario achievements is defined, and a detailed workflow diagram is presented depicting the interaction with various ASSURED components. Moreover, for each use case there is a demonstration setup description detailing the devices to be used for the demonstrator setup and the software running at each one of them. A set of KPIs (quantitative and qualitative) along with acceptance criteria is included in each one of the demonstrator specific chapters, defining specific metrics to be monitored in order to evaluate the operation of ASSURED.

In order to specify even more the execution of demonstrators, the unit tests to be applied in the ASSURED framework are described. More specifically description has been provided for a set of four horizontal test cases ensuring the correct operation of all involved ASSURED components. In addition, a set of **26 demonstrator specific test cases** in total has also been defined and described along with the components of the demonstrator involved in order to test the operation of ASSURED within the context of each one of the four use cases.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 SCOPE AND PURPOSE

The main goal of this deliverable is to put forth a **detailed testing and evaluation plan that will guide all of the functional and technical integration efforts of WP6 towards the setup of the overall ASSURED framework in the context of the envisioned use cases.** Focus is placed on the interfaces that need to be implemented/provided per component and per layer of the ASSURED architectural approach. Aspects related to the implementation, version control system, continuous integration, quality assurance, release planning and issue tracking will not be included in this deliverable since they have been already provided in Chapter 5 of Deliverable D1.2 [1].

1.2 RELATION TO OTHER WPS AND DELIVERABLES

As an evaluation framework and demonstrators planning deliverable, D6.1 relates with Deliverable D1.1 [2] as it further details the scenarios to be performed for evaluating the specific functionalities of the ASSURED framework. The deliverable is also related with D5.1 as it describes the setup of demonstrators and providing input about the demonstrator constraints, devices and related software to be used, which is necessary input for the integration activities planning and execution.

We have to highlight that during the compilation of this deliverable, the implementation and integration activities of the ASSURED framework had already commenced, thus, some of the scenarios were also updated to better reflect the various security processes and enablers to be evaluated. Considering the complete security pipeline of ASSURED, starting from the Risk Assessment Engine, to the Policy Recommendation Engine, to the deployment of the calculated attestation policies as smart contracts (through the Security Context Broker), to the download and execution of the attestation tasks, after the successful authentication of the devices (leveraging the designed TPM-based Wallet), to, finally, the verification and recording of the attestation results on the ledger.

Within WP6, D6.1 **sets the goals, the scenarios and the time plan for the demonstrator's execution**, and therefore will be the basis for the next deliverables about the Demonstrators Implementation (First D6.2 and Final D6.3), as well as for the performance evaluation and Adoption Guidelines (D6.4).

The current deliverable contributes to and concludes Milestone MS5, as it delivers the **demonstrators evaluation framework** concluding the work of T6.1. This deliverable will also be provided as an input to demonstrator's implementation tasks (Task 6.2 – 6.5) of the four different use cases, providing the scenarios to be executed, the ASSURED functionalities to be integrated to each one of them and specific test cases to be performed. **Finally, the metrics to be monitored including business and technical indicators to be used for ASSURED evaluation will contribute as a basis for ASSURED evaluation work at T6.6 and deliverable D6.4 (Performance Evaluation and Adoption Guidelines).**

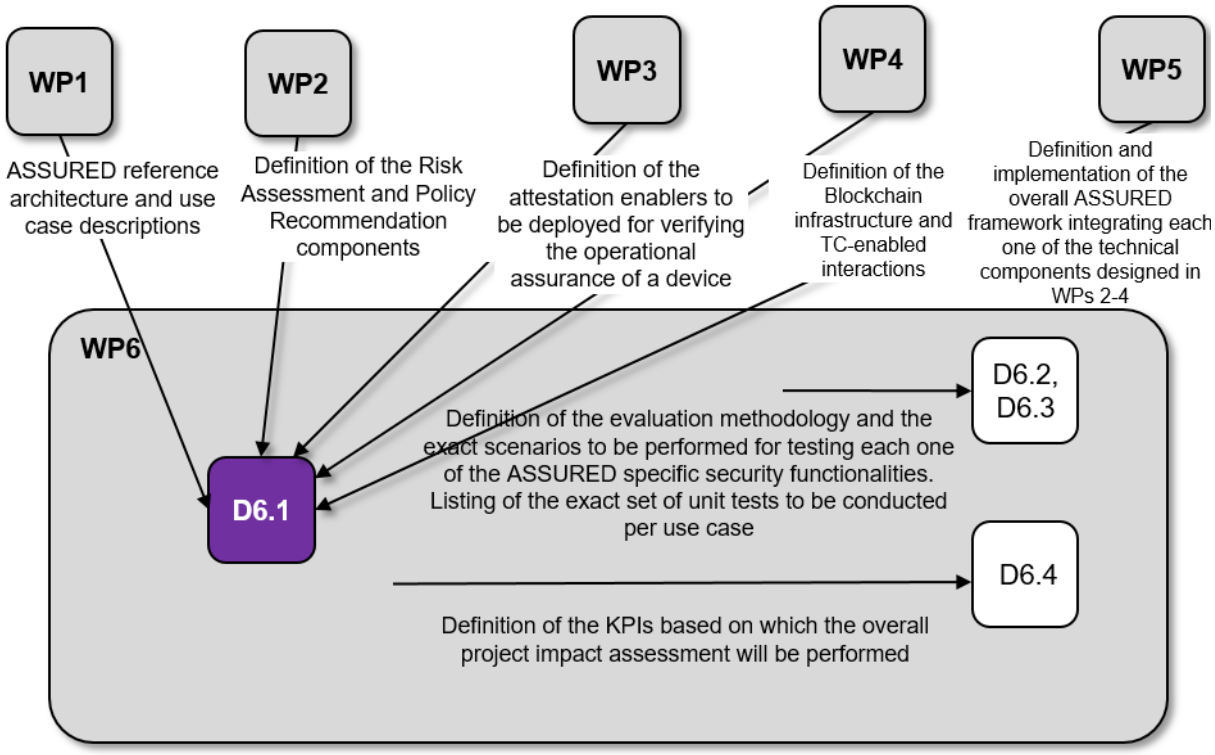

FIGURE: *RELATION OF D6.1 WITH THE OTHER ASSURED WPS & DELIVERABLES*

1.3 DELIVERABLE STRUCTURE

The deliverable is structured as follows: In **Chapter 2**, we describe the evaluation framework and its high-level goals. The approach and the metrics are described for the technical and business validation of the platform. Across the next four chapters, **Chapters 3, 4, 5** and **6**, a detailed analysis and planning follows for each one of the four use cases. Apart from a description and the challenges of each demonstrator, the scenarios to be used are outlined in detail and along with a workflow schema. At the same time, specific metrics (both quantitative and qualitative) have been defined and described setting the goals and acceptance criteria for the evaluations to take place to each one of the demonstrators. **Chapter 7** serves with providing the test cases to be used in order to test ASSURED functionalities as a whole (chapter 7.1) and also the demonstrator specific ones. Finally, **Chapter 8** concludes the deliverable.

2 EVALUATION FRAMEWORK AND PLANNING

The **evaluation framework of ASSURED** shall propose a series of coordinated evaluation actions which can be performed in an as much **unified manner as possible across all four demonstrators with a goal to demonstrate the value of ASSURED as a whole and contribute to shaping its Unique Selling Propositions.** The results of the evaluation actions shall be provided as feedback and support to the development teams, to ensure the growth, viability and sustainability of the ASSURED platform.

The high-level goals of the evaluation framework are:

(a) to **ensure that the ASSURED platform is built according to the requirements** set by and generates the expected benefits for the stakeholders of the platform as defined in Section 2.6 of Deliverable 1.1 [2] and the applications they build, and
(b) to **guide the continuous evaluation of the ASSURED platform throughout the whole implementation phase** of the project from M7 to M30. The evaluation activities across all demonstrators shall be monitored and aligned in order to provide structured and actionable feedback to the development teams.

The following subsections present the evaluation framework to be defined, executed and monitored in the context of WP6.

2.1 APPROACH TO EVALUATION

The consortium has opted to base the ASSURED evaluation framework on the basic principles of the **Validation and Verification (V&V) methodologies of software products**. V&V methodologies, following up on the V model approach [3] , cover the whole development cycle of a software product based on the active engagement of the demonstrators in multiple demonstration iterations, exposing them to incremental versions of the platform services and APIs and generating feedback loops, allowing the developers to improve their components and the platform as a whole.

The application of V&V based methodologies addresses:

(a) verification, i.e., the **discovery and elimination of defects, gaps in development and possible security issues**, and
(b) validation, i.e., the **fulfilment of the stakeholders' needs and the generation of the expected benefits.**

The definition of the ASSURED evaluation framework should reply to the following questions:

Is the ASSURED platform operating according to its specifications? This question concerns the technical validation of the project and has to be answered by conducting a quantitative technical evaluation, e.g., testing technical parameters of system availability, functionality, security and performance. The baseline is the platform reference architecture as defined in Deliverable 1.2 [1] and the technical work performed in WP2-WP5.

Does ASSURED meet the defined objectives from the perspective of its users? This question is closely related to product validation and business validation; the demonstrator partners are directly involved in replying to it. During product validation, the focus is on platform usability, user acceptance, user satisfaction, etc. During business validation, the focus is on the contribution to different KPIs of business interest, from direct costs (and time therefore) to the strategic objectives of the call, to other aspects such as perceived Quality of Service, level of

trust, etc. The baseline is the use cases which have been defined in the project's DoA [4] and further elaborated in section 4 of deliverable 1.1, as well as the stakeholders identified in section 2.6 of deliverable 1.1 [2].

2.1.1 Technical Validation Approach

The ISO/IEC 25010:2011 "Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models" proposes a set of models that better address the evaluation of the software quality.

The product quality model is composed of eight characteristics (which are further subdivided into 31 sub-characteristics) that relate to static properties of software and dynamic properties of the computer system. The model is applicable to both computer systems and software products.

FIGURE 1: ISO/IEC 25010:2011 - PRODUCT QUALITY MODEL[5]

1. **Functional Suitability** - The degree to which the product provides functions that meet stated and implied needs when the product is used under specified conditions.
2. **Performance Efficiency** - The performance relative to the number of resources used under stated conditions.
3. **Compatibility** - The degree to which two or more systems or components can exchange information and/or perform their required functions while sharing the same hardware or software environment.
4. **Usability** - The degree to which the product has attributes that enable it to be understood, learned, used and attractive to the user, when used under specified conditions.
5. **Reliability** - The degree to which a system or component performs specified functions under specified conditions for a specified period.
6. **Security** - The degree of protection of information and data so that unauthorised persons or systems cannot read or modify them, and authorised persons or systems are not denied access to them.
7. **Maintainability** - The degree of effectiveness and efficiency with which the product can be modified.
8. **Portability** - The degree to which a system or component can be effectively and efficiently transferred from one hardware, software or other operational or usage environment to another.

The standard itself does not propose any metric, rather it is the adopter of the standard who should select which characteristics and sub-characteristics are applicable to the software under evaluation and create metrics for those. Table 1 shows in detail the sub-characteristics of each characteristic and indicates their suitability for ASSURED.

Sub-characteristics	Definition	Suitability for ASSURED
Functional Suitability		
Functional completeness	Degree to which the set of functions covers all the specified tasks and user objectives.	High
Functional correctness	Degree to which a product or system provides the correct results with the needed degree of precision.	High
Functional appropriateness	Degree to which the functions facilitate the accomplishment of specified tasks and objectives.	High
Performance Efficiency		
Time behaviour	Degree to which the response and processing times and throughput rates of a product or system, when performing its functions, meet requirements. This is especially important in the context of ASSURED since we are dealing with "Systems-of-Systems" providing safety-critical operations with strict time constraints (e.g., Smart Manufacturing – Chapter 3).	High
Resource utilisation	Degree to which the amounts and types of resources used by a product or system, when performing its functions, meet requirements.	Medium
Capacity	Degree to which the maximum limits of a product or system parameter meet requirements.	Low
Compatibility		
Co-existence	Degree to which a product can perform its required functions efficiently while sharing a common environment and resources with other products, without detrimental impact on any other product. This is rather important in the context of ASSURED where all of the security enablers are deployed at the edge devices for protecting the concurrent execution of the device computational tasks.	High
Interoperability	Degree to which two or more systems, products or components can exchange information and use the information that has been exchanged. In the context of ASSURED, this pertains to the data interoperability attributes for the attestation data recorded on the distributed ledger. It should be possible for an entity, with the appropriate privileges, to query/read from a ledger and then securely transfer this claim to another ledger for these registered devices to have access to.	Medium
Usability		
Appropriateness recognisability	Degree to which users can recognize whether a product or system is appropriate for their needs.	Low
Learnability	Degree to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use.	High
Operability	Degree to which a product or system has attributes that make it easy to operate and control.	High
User error protection	Degree to which a system protects users against making errors.	Low

User interface aesthetics	Degree to which a user interface enables pleasing and satisfying interaction for the user.	Low
Accessibility	Degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use.	Low
Reliability		
Maturity	Degree to which a system, product or component meets needs for reliability under normal operation.	High
Availability	Degree to which a system, product or component is operational and accessible when required for use.	High
Fault tolerance	Degree to which a system, product or component operates as intended despite the presence of hardware or software faults.	High
Recoverability	Degree to which, in the event of an interruption or a failure, a product or system can recover the data directly affected and re-establish the desired state of the system.	High
Security		
Confidentiality	Degree to which a product or system ensures that data are accessible only to those authorised to have access based on their ability to exhibit specific attributes and partial identifiers.	High
Integrity	Degree to which a system, product or component prevents unauthorised access to, or modification of, computer programs or data.	High
Non-repudiation	Degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later.	High
Accountability	Degree to which the actions of an entity can be traced uniquely to the entity.	High
Authenticity	Degree to which the identity of a subject or resource can be proved to be the one claimed.	High
Maintainability		
Modularity	Degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components.	High
Reusability	Degree to which an asset can be used in more than one system, or in building other assets.	Medium
Analysability	Degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified.	Low
Modifiability	Degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality.	Low
Testability	Degree of effectiveness and efficiency with which test criteria can be established for a system, product or component and tests can be performed to determine whether those criteria have been met.	Medium

Portability		
Adaptability	Degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments.	High
Installability	Degree of effectiveness and efficiency with which a product or system can be successfully installed and/or uninstalled in a specified environment.	Low
Replaceability	Degree to which a product can replace another specified software product for the same purpose in the same environment.	Low

TABLE 1: *PRODUCT QUALITY MODEL - TECHNICAL CHARACTERISTICS, SUB-CHARACTERISTICS AND RELEVANCE TO ASSURED*

2.1.2 Business Validation Approach

The ISO/IEC 25010:2011 Quality in Use model considers the user's point of view to measure the perception of the quality of the system. The different characteristics and sub-characteristics of this model are derived from testing or observing the results of real or simulated use of the system. The Quality in Use model is composed of five characteristics (some of which are further subdivided into sub-characteristics) that relate to the outcome of interaction when a product is used in a particular context of use. **This system model is applicable to the complete human-computer system, including both computer systems in use and software products in use.**

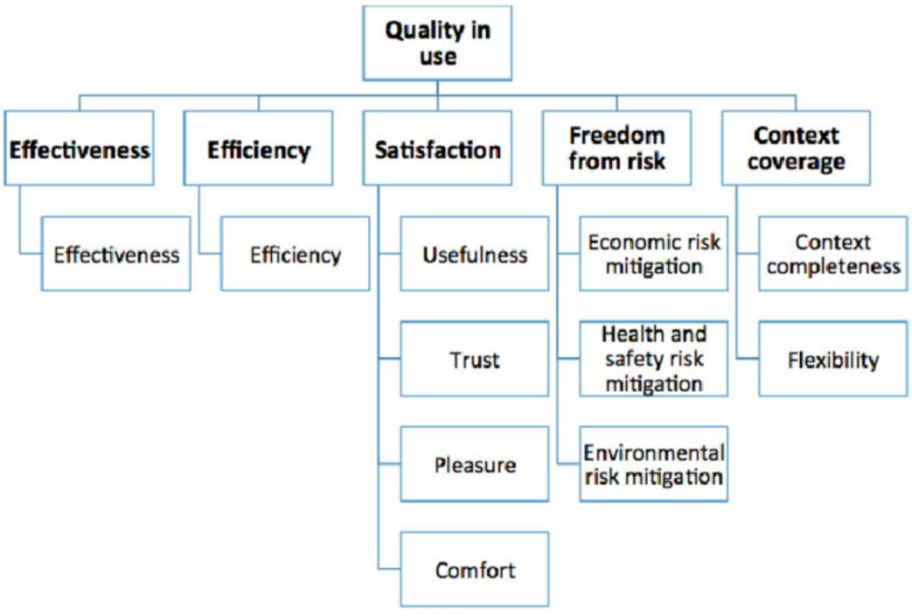

FIGURE 2: *ISO/IEC 25010:2011 - QUALITY IN USE MODEL*[6]

1. **Effectiveness** - The accuracy and completeness with which users achieve specified goals.
2. **Efficiency** - The resources expended in relation to the accuracy and completeness with which users achieve goals.
3. **Satisfaction** - The degree to which users are satisfied with the experience of using a product in a specified context of use.

4. **Freedom from risk** - The degree to which a product or system mitigates the potential risk to economic status, human life, health, or the environment.
5. **Context coverage** - The degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in both specified contexts of use and in contexts beyond those initially explicitly identified.

Table 2 shows in detail the sub-characteristics of each characteristic and indicates their suitability for ASSURED.

Sub-characteristics	Definition	Suitability for ASSURED
Effectiveness		
Effectiveness	Degree of accuracy and completeness with which users achieve specified security and operational assurance goals when using the system.	High
Efficiency		
Efficiency	Degree to which the users find that the software is efficiently covering its intended purpose. Particular focus of the Control-Flow Attestation enabler.	High
Satisfaction		
Usefulness	Degree to which a user is satisfied with their perceived achievement of pragmatic goals, including the results of use and the consequences of use.	High
Trust	Degree to which a user or other stakeholder feel that they can trust the system and have confidence that a product or system will behave as intended.	High
Pleasure	Degree to which a user finds the software's functions a pleasure to use (emotionally).	Low
Comfort	The degree to which users think that the system provides the comforts needed (physically)	Low
Freedom from risk		
Economic risk mitigation	Degree to which a product or system mitigates the potential risk to financial status, efficient operation, commercial property, reputation or other resources in the intended contexts of use.	High
Health and Safety risk mitigation	Degree to which a product or system mitigates the potential risk to people in the intended contexts of use.	Low
Environmental risk mitigation	Degree to which a product or system mitigates the potential risk to property or the environment in the intended contexts of use.	Low
Context coverage		
Context completeness	Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in all the specified contexts of use	High
Flexibility	Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in contexts beyond those initially specified in the requirements.	Low

TABLE 2: *QUALITY IN USE MODEL - CHARACTERISTICS, SUB-CHARACTERISTICS AND RELEVANCE TO ASSURED:*

2.2 THE ASSURED EVALUATION FRAMEWORK

The following ASSURED evaluation framework is suggested to enable the success of the platform and to learn as much as possible from the four demonstrators.

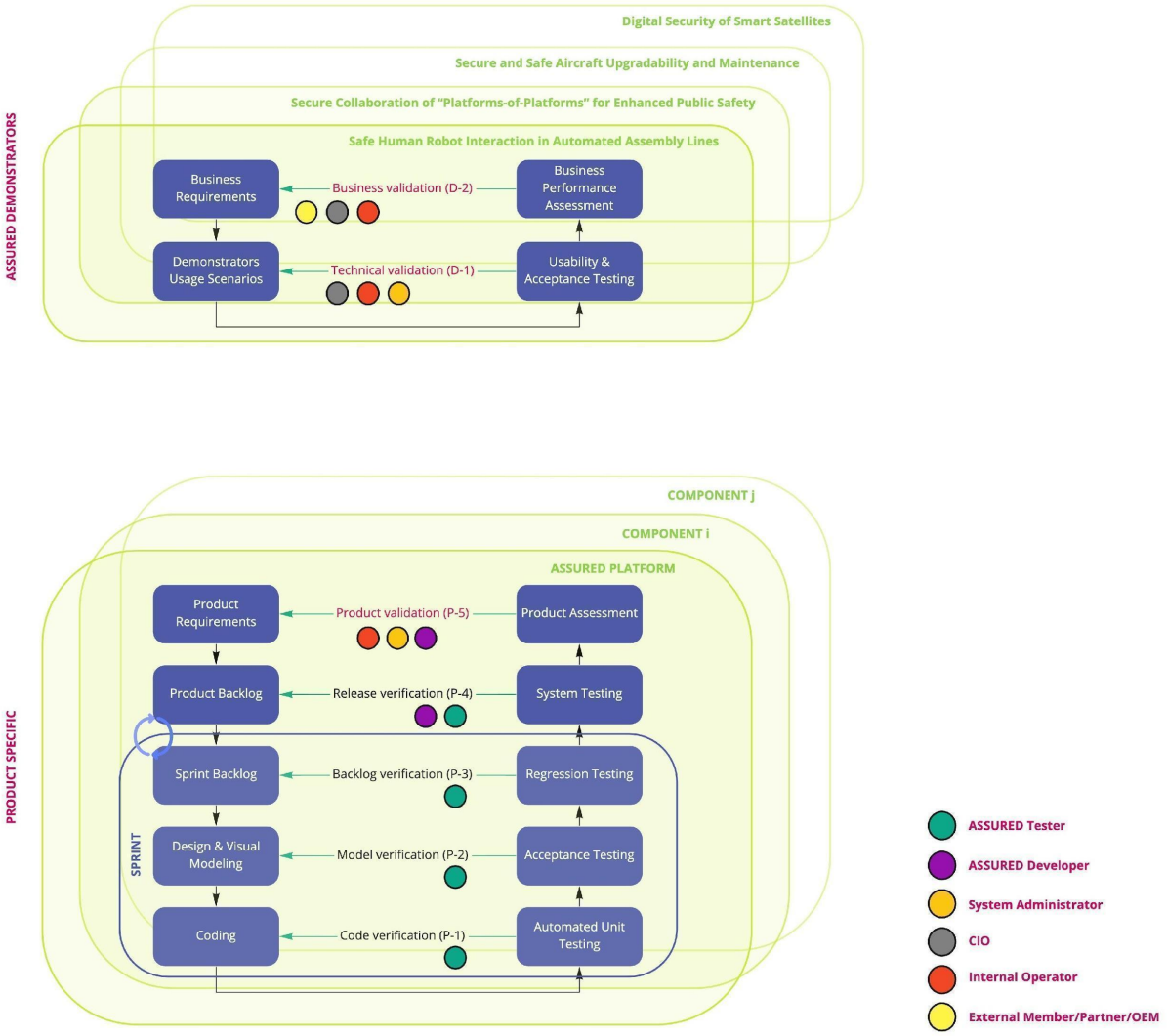

FIGURE 3: *ASSURED EVALUATION FRAMEWORK*

In Figure 3 above we see two main perspectives:

- **Product-specific perspective** which concerns the platform as a product and its individual components, including: Code Verification (P-1) to ensure functionality, correctness, reliability, and robustness of code; Model Verification (P-2) to align the design with collected requirements; Backlog Verification (P-3) to determine whether the requirements of the product after each sprint are met; Release Verification (P-4) to checks whether the requirements of each major release are met; and Product Validation (P-5) to investigate whether the platform as a whole satisfies intended use and user needs. P-1, P-2, P-3, and P-4 are not in the scope of WP6. All aspects of the V model are covered, though; the collection of requirements and use cases of demonstrators is covered in the context of WP, while WP5 covers the integration and

testing plan, as well as the code maintenance lifecycle and the continuous integration and delivery framework.

- **Demonstrator perspective** which involves the four demonstrators evaluating the platform and the applications that are created using the platform depending on their use case, in the following steps: Technical Validation (D-1) to guarantee that the overall platform satisfies intended use and user needs from a technical and functional point of view only; Business Validation (D-2) to assess whether the overall platform eventually offers sufficient added value and has clear business benefits to the demonstrator, allowing it to operate more efficiently.

The following sub-sections present the **quantitative and qualitative metrics which will be used by the consortium to evaluate the performance of ASSURED**. Following the main directions of the chosen standard and the KPIs used in Section 2 of the Description of Action (DoA), Part B [4] to measure the impact of the project, different indicators been adapted appropriately to the scope and nature of the project in order to produce an evaluation framework that can be utilised for evaluating each one of project's assets.

2.2.1 Technical Validation

Based on the product quality model characteristics of high importance according to Table 1, the following **list of metrics has been devised in order to allow the technical assessment of the ASSURED solution.** It needs to be noted that due to the nature of the project and based on the operation conditions of the pilots, measuring some of the below-mentioned indicators might not be possible or alternatively not producing meaningful results.

Based on the following Table 3, the specific technical KPIs will then be put forth for each one of the envisaged reference scenarios in the following Chapters. The goal here is to give an overview of the spectrum of values that is considered as an acceptable behaviour for the ASSURED framework prior to extracting the exact needs tailored to each use case requirements.

	Sub-characteristics	Metrics	Calculation Type	Recommended Limit	Mandatory/ Optional
	Functional Suitability				
	Functional completeness	Number of devices that can achieve the required level of security and safety as has been defined by the Systems Administrator	Number of threats and vulnerabilities that can be protected against Number of re-usable attestation methods	> 70% (at least all devices should be able to detect any attacks that affect their control-flow or configuration) > 4 (Control-Flow Attestation, Configuration Integrity Verification, Swarm Attestation, Zero Knowledge Attestation)	Mandatory
	Functional correctness	Number of devices, in an infrastructure, whose configuration and execution correctness can be monitored in real-time	Number of devices for which no real attestation data can be monitored either due to compromise or ASSURED communication malfunctioning	100%	Mandatory

	Functional appropriateness	Correctness of the monitored system traces and the extracted attestation results depicting the operational assurance of a device	Integrity of logs against alteration attacks	100%	Mandatory
Performance Efficiency					
	Time behaviour	Time that an attestation process takes to execute in a device broken down to all internal operations: Trace Extraction + Encrypt.Sign + Verification	Completed attestation processes/Execution time available for security operations	No computational task should be affected by the execution of a security enabler (100%)	Mandatory
	Resource utilisation	CPU cycles required for supporting the execution of all ASSURED security enablers and crypto operations offered by the TPM-based Wallet	CPU Cycles required per security operation and interaction with the underlying TPM	ALL ASSURED security operation should need < 50% of an available device resources	Desirable
	Capacity	Percentage of attacks and threats that can be identified by the ASSURED security enablers	Number of host-based and network-based of attacks identified per attestation scheme	> 80% of sw-based attacks; > 40% data oriented attacks > 75% network-based attacks	Mandatory
Compatibility					
Usability					
	Operability	Automation, Run-time, User Notification in case of security alerts, Easy to run and update	Number of alerts produced for different types of attacks to be understandable by the end users Number of re-suable lightweight crypto operations	> 1 property definition language for expressing the mitigation logic of specific threats > 3 (ABE, ABAC, SE)	Desirable
	User error protection	Automation of the deployment process of all ASSURED components and artefacts to minimize user burden	Number of scripts to be provided for instantiating the ASSURED framework	< 10 scripts	Desirable
Reliability					
	Maturity	All ASSURED components should be able to withstand attacks against themselves	Number of attacks mitigated trying to affect the operation of the attestation enablers, TPM	> 70% of attacks targeting the ASSURED software stack (besides the	Mandatory

			Wallet and the on-chain interactions	networking interface cards)	
	Availability	ALL devices should be able to be attested and verified against their operational assurance at any time during their lifecycle. Thus, ASSURED attestation enablers should be continuously operating based on the latest optimal set of security policies	Number of queries supported for checking the operational assurance of a device – and for different resources to be attested	100%	Mandatory
	Fault tolerance	ALL devices should be able to quickly detect and react to any indicators of compromise as detected by the ASSURED attestation enablers and subsequently by the Attack Validation component	Number of ASSURED attestation enablers affected by a software bug; Number of software bugs not identified by the attestation enablers and the fuzzing mechanisms	< 90% (all attacks that affect the control-flow or configuration of a device should be detectable. *Currently excluding data oriented attacks to be checked in the second release*)	Mandatory
Security					
	Confidentiality	Access to both operational and attestation data, recorded on the Blockchain, should be granted to only those users that can depict the appropriate attributes in a verifiable manner (Verifiable Credentials and Verifiable Proofs)	Number of attributes and data models to be supported for depicting verifiable proofs; Number of different sets of stakeholders to be supported in the access control scheme	> 2 (data models from the W3C [8]) > 8	Mandatory
	Integrity	Unauthorized or Compromised processes should not have access to sensitive material within a device; In the same context, unauthorized users should not have access to recorded data. *In both cases, the provenance of data should be verifiable*	Number of unauthorized process and/or users been able to extract or leak sensitive information	0% (data and digital integrity should be met for all data transactions taking place within the ASSURED ecosystem)	Mandatory

	Non-repudiation	Provenance of all actions should be kept in an auditable manner. Either been operational actions or attestation-related actions (i.e., Verifier device checking the traces of the Prover and recording the result on the ledger)	Number of unauthorized and non-signed actions been allowed in the ASSURED ecosystem; Number of (Verifier) devices that deviate from the attestation protocol and record falsified results	0% 0% (all such devices should be correctly captured by the Jury-based Attestation scheme once such a deviation is detected)	Mandatory
	Accountability	ALL actions as it pertains to the attestation of the operational assurance of a device should be linkable back to the Verifier	Time needed for signing the attestation result based on the Attestation Key usage policies defined in the TPM-based Wallet	< 500 ms	Mandatory
	Authenticity	Number of Verifiable Credentials and Verifiable Proofs that can be managed by the TPM-based Wallet for making sure that an entity is the one that it claims to be	Number of devices and users producing wrong Verifiable Credentials or impersonating another user by using his/her credential	0% (ASSURED Wallet should be binded to the Holder device or user)	Mandatory
Maintainability					
	Modularity	Change in the ASSURED software stack, running at the edge, should not have any impact to the backend ASSURED insfrastructure (Risk Assessment, Policy Recommendation, Blockchain)	Number of updates in the ASSURED attestation enablers and/or the Tracer that might affect the rest of the ASSURED operational chain	< 10% (Only changes to the type of system traces to be monitored might require update of the smart contracts created for holding them on the ledger)	Desirable
Portability					
	Adaptability	ALL ASSURED components should be able to run in heterogeneous types of systems deployed in the edge	Number of devices been able to instantiate and execute all ASSURED security enablers Number of reusable SDKs per component (for supporting different OSes)	> 6 (considering all of the different devices to be leveraged in the context of the four envisaged reference scenarios) > 2	Mandatory

TABLE 3: *PRODUCT QUALITY EVALUATION – QUANTITATIVE METRICS SELECTED FOR ASSURED*

2.2.2 Business Validation

The success of the ASSURED solution and the project as a whole is closely linked to the successful implementation and execution of the four project's demonstrations, which are expected to play the role of success stories for the project. To include the demonstrators' perspective in the evaluation and address the expectations and requirements of the stakeholders, each demonstrator shall define their KPIs and formulate their test scenarios and test cases (in Chapters 3 to 6 below).

Together with the demonstrator specific KPIs, the **following KPIs are proposed as a means to measure the project's impact on various application domains.** *It needs to be mentioned, that the KPIs presented in Table 4 might not be measurable in every demonstrator.* They should be considered as baseline KPIs to demonstrate the impacts of the ASSURED solution as a whole.

ID	Business Metric	Units	Description
GENERIC BUSINESS KPIs (COMMON TO ALL USE CASES)			
ASRD-KPI-01	Revenues from selling security solutions based on ASSURED	€ / time	The amount of revenue in € in a given period of time from selling solutions/services based on ASSURED.
ASRD-KPI-02	Deployment time for cybersecurity services through ASSURED SDKs	minutes	The time it takes to deploy a new instance of the cybersecurity service.
ASRD-KPI-03	Delivery cycle for cybersecurity services through ASSURED SDKs	minutes	How long it takes to deliver a change in the service into production.
ASRD-KPI-04	Cloud Infrastructure Costs (OPEX)	€ / time	Total Infrastructure Cost (i.e., for supporting Blockchain infrastructure management, remote sw- and firmware update, etc.) for running the service per unit of time.
ASRD-KPI-05	Successful attempts at breaching privacy to personal, societal and industrial data	No. of Security Incidents / time	Number of security incidents recorded per unit of time.
ASRD-KPI-06	Identification, reporting and decrease of cyber-threats per organizational entity	No. of Cyber-threats / time	Number of cyber-threats detected per unit of time.
BUSINESS KPIS FOR "PUBLIC SAFETY" USE CASE			
ASRD-KPI-07	Scalability and applicability to other smart cities	No. authorization accounts/devices	Number of city actors and external parties that can connect to a smart city Number of edge devices dynamically attested and added in a system (swarm)

ASRD-KPI-08	**Extensibility** to other technologies and city services; **Interoperability** with other Service Providers (including device vendors), thus, having mixed-ownership requirements	No. verified services	Number of services that can be integrated in the framework, potentially from external providers
ASRD-KPI-09	Policy making for cities sustainability	No. countermeasures and decisions on attacks	Number of measures/actions taken from cities in case of attacks that can result to the launch of a city policy/decision
ASRD-KPI-10	Decrease security incidents	% of security incidents decrease	Decrease in the number of security incidents.
BUSINESS KPIs FOR HRI "HUMAN ROBOT INTERACTION" USE CASE			
ASRD-KPI-11	More secure and efficient software updates distribution.	No. of protection mechanisms integrated.	Support the secure and efficient distribution of software updates based on different modes of operation: (i) Same SW Update to all devices through the Blockchain, and (ii) Direct remote update of a deployed robot
ASRD-KPI-12	Deployment and Dynamic, Efficient delivery of newly created policies.	Time	Time to deploy a newly created policy.
ASRD-KPI-13	Remote update procedure improved.	Costs	Costs improvements on the current procedure, including engineers' expertise, travel costs, and airplane forced on the ground.
ASRD-KPI-14	Trusted traceability of all the operations made on any device.	Time	Approval time of any procedure in the whole supply-chain, i.e., remote update and maintenance, considerably reduced.
BUSINESS KPIs FOR "SECURE AEROSPACE" USE CASE			
ASRD-KPI-15	Remote maintenance lifecycle improved.	Costs	Maintenance costs benefit from secure remote maintainability.
ARD-KPI-16	Reduction of secure data acquisition time	Time vs. Resources used for the setup of secure & authentic communication channels	Time needed for securely extracting information/data from the onboard devices in an aircraft. This can either be operational data (used for maintenance) or threat intelligence data
BUSINESS KPIs FOR "SMART SATELLITES" USE CASE			

ASRD-KPI-17	Ease of Integration of Components on Trusted Device	Minutes/hours compared to days, Reduced installation costs and on-site adaptation changes for engineers	Time taken to enrol and register new and similar devices securely
ASRD-KPI-18	Ease of Dynamic Policy Manipulation / Update	Minutes compared to hours. Reduced software debugging costs for potential policy update and changes	Interaction with system, Intuition development in terms of software from UI/UX perspective
ASRD-KPI-19	Reduction of cost for security enablers deployment	Time (minutes)	Time needed for deploying the appropriate security (attestation) enablers and activating them in the deployed satellites

TABLE 4: *BUSINESS KPIS*

3 SAFE HUMAN ROBOT INTERACTION IN AUTOMATED ASSEMBLY LINES DEMONSTRATOR

3.1 SAFE HUMAN ROBOT INTERACTION IN AUTOMATED ASSEMBLY LINES

In this chapter, we revise the user stories provided in D1.1 [2] and re-evaluate both the quantitative and qualitative Key Performance Indicators (KPIs) per reference scenario that will be extensively tested in the context of the "***Safe Human Robot Collaboration***" (HRC) use case. As was already introduced, the focus in this use case is the **convergence of the security, trustworthiness and safety requirements so as to be able to assess the trust level of each deployed device prior to making a safety-critical decision while considering the Zero Trust principle.** More specifically, this use case focuses on allowing **real-time interaction between humans and cyber-physical systems in a safe and reliable manner.**

At a high-level, an HRC system is a Collision Prediction and Avoidance system aimed at reducing risk of accidents involving Personnel and Robots in an indoor environment as shown in Figure 4. The following information is required at periodic interval: **Personnel's current 3D Coordinates and motion dynamics**, and **Robot's current 3D Coordinates and motion dynamics**.

Using the above information, predictions on collision are made a-priori. Based on the probability of collision, the collision prediction and avoidance system sends control messages to slow down or stop the Robot, thus avoiding the collision between Personnel and Robot and hence avoiding workplace mishaps.

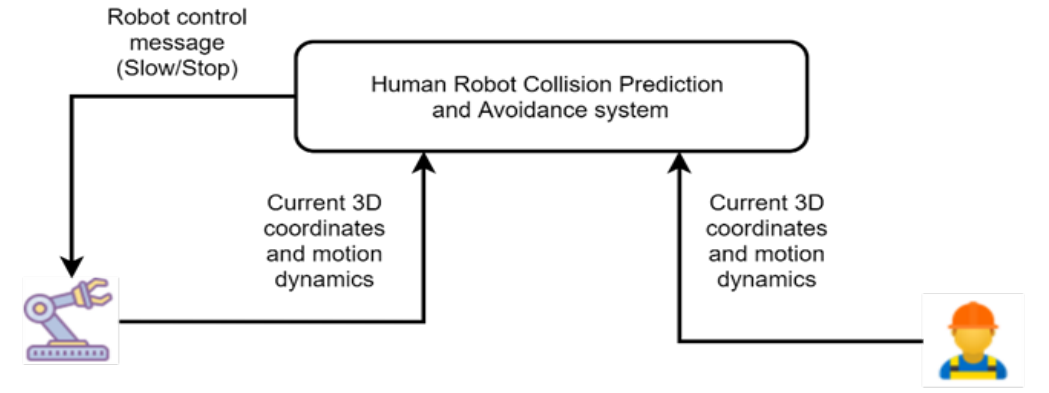

FIGURE 4: *SIMPLIFIED DIAGRAM OF HUMAN ROBOT COLLABORATION USE CASE*

Table 5 provides a summary of the user stories which were defined in D1.1. The goal here is to summarise the scenarios and to pinpoint the specific functionalities to be demonstrated for each story. The aim is to capture the entirety of ASSURED functionalities and evaluate them all, under the defined use case scenarios in order to provide a complete evaluation testbed.

In the context of this demonstrator, we have defined a set of use case scenarios to demonstrate the operation of the designed ASSURED artifacts. Among the numerous scenarios there is an overlapping among the technologies used, but there is always a distinct core functionality (or a unique combination of them) being evaluated each time. Thus, we document in Table 5

below the ASSURED exploitable artifacts that take part in the demonstrator, and we highlight, at the "Functionalities" column, the particular artifacts being applied and evaluated.

Among the functionalities of ASSURED, some act as prerequisites in order to offer a complete flow to the demonstration scenarios. Thus, for this demonstrator the following components are present to all the user stories: Risk Assessment, Policy Recommendation Engine, and Blockchain services.

User Story	Security Property	Functionalities
BIBA.US.1	Securely run use case application such as RMT, PLMC, CPA services on IoT Gateway. ➢ Operational assurance ➢ Data integrity	✓ Risk Assessment ✓ Policy Recom. Engine ✓ Attack Validation Comp. ✓ Control-Flow Attestation ✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet
BIBA.US.2	Secure retrieval of logs / data of RMT, PLMC, CPA services running on IoT Gateway. ➢ Data Integrity ➢ Confidentiality of net. Com. ➢ Continuous authentic. & Authorisation	✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet
BIBA.US.3	Attest and monitor correct execution behaviour of data filtering application, which processes the incoming data before forwarding to RMT, PLMC and CPA modules, running on IoT Gateway (Raspberry Pi) or multiple IoT Gateway (Raspberry Pi cluster in same network). ➢ Operational Assurance ➢ Data Integrity ➢ Confidentiality of net. Com.	✓ Control-Flow Attestation ✓ Config. Integrity Verif. ✓ Swarm Attestation ✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet
BIBA.US.4	All connected trusted devices like IoT Gateway, Data aggregator in the infrastructure must register and establish a secure communication channel. ➢ Data Integrity ➢ Confidentiality of net. Com. ➢ Continuous authentic. & Authorisation	✓ Blockchain services ✓ TPM-based Wallet
BIBA.US.5	Secured communication channel between trusted devices (such as data aggregators) and IoT Gateway (Raspberry Pi) running RMT, PLMC, CPA services. ➢ Operational Assurance	✓ Attack Validation Comp. ✓ Control-Flow Attestation ✓ Config. Integrity Verif. ✓ Runtime Tracing ✓ Blockchain services

	➢ Confidentiality of net. Com. ➢ Continuous authentic. & Authorisation	✓ TPM-based Wallet
BIBA.US.6	Enrol a new trusted device such as IoT Gateway, Data aggregator into the smart manufacturing infrastructure without any manual provisioning. ➢ Operational Assurance ➢ Data Integrity ➢ Confidentiality of net. Com. ➢ Continuous authentic. & Authorisation	✓ Attack Validation Comp. ✓ Config. Integrity Verif. ✓ Swarm Attestation ✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet
BIBA.US.7	Query execution details at different level of the services running on IoT Gateway. ➢ Data Integrity ➢ Confidentiality of net. Com.	✓ Attack Validation Comp. ✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet

TABLE 5: SMART MANUFACTURING REFERENCE SCENARIO OVERVIEW

3.1.1 Planning

For this demonstrator, five user stories have been identified and a planning for the first release has been completed considering the readiness of the pilot site and the technical advancements of ASSURED. Thus, **User stories BIBA.US.1, BIBA.US.4, BIBA.US.7 will be tested in the 2nd release, while user stories BIBA.US.2, BIBA.US.3, BIBA.US.4 (partially), BIBA.US.5 and BIBA.US.6 will be tested in the first release**.

In the following section, the focal point is the user stories of the first release where sequence diagrams, workflows and detailed descriptions are included. The rest are mentioned in summary.

For the first release following, the following user stories will be validated:

ID	User story	Validations
BIBA.US.2	As a System Administrator, I want to **ensure that all the devices' (configuration and execution) log traces are stored securely**, in order for the IoT Gateway to be able to verify the device integrity.	Configuration files and log traces of RMT, PLMC, CPA services are stored securely by using the ABE scheme of ASSURED to encrypt and store them to the data storage engine. By using smart-contracts (for trusted access control) and the policy-compliant Blockchain technology from the ASSURED framework, these files can be secured, and access authentication and authorisation can be provided.
BIBA.US.3	As a System Administrator, I want **to continuously monitor run-time execution of the software components on a single device** or a network of	The execution of the data filtering application, which processes incoming data before forwarding to RMT, CPA and PLMC modules, is continuously

	interconnected devices, in order to attest their correct behaviour when calculating the worker positioning data.	monitored. By continuously monitoring the application, using ASSURED runtime tracing and CFA, undefined behaviours of the software can be halted before a failure occurs.
BIBA.US.4	As a System Administrator, I want **to verify in real-time all interconnected devices against baseline security authentication requirements**, in order to monitor the overall trust state of the manufacturing infrastructure.	Interconnected systems, such as Data aggregator, must be verified in real-time against baseline security authentication. By utilising ASSURED frameworks, continuous authentication and authorisation of devices will be achieve using the BC-wallet and the BC services, and attacks that involve the use of unidentified devices can be mitigated.
BIBA.US.5	As a System Administrator, I want **to secure the wired or wireless communication between devices** belonging to the same manufacturing environment, in order to establish a trusted management channel with the IoT Gateway.	Systems such as Data aggregator must establish a trust management channel with the IoT Gateway. Specified attestation policies will be deployed to regulate the trusted management of the devices. The use of the TPM-based wallet and Smart contracts will be key enablers to meet this requirement.
BIBA.US.6	As a System Administrator I want **to securely enrol new devices in my existing manufacturing environment**, without the need of any manual intervention, in order to enhance the accuracy of my CAM component and service.	IoT Gateway must securely enrol new systems such as Data aggregator in the manufacturing environment. The use of the TPM-based wallet, the Smart contracts that describe the necessary device on-boarding policies, and the ASSURED Blockchain will be key enablers to meet this requirement.

TABLE 6: *REFERENCE SCENARIO 1 FIRST RELEASE DEMONSTRATOR SUMMARY*

3.1.2 Description and User Stories

The user stories presented in the Safe Human Robot Interaction in Automated Assembly Lines use case are focused towards reducing susceptibility within the industrial environment where compromised software may lead to work mishaps between industrial-grade robotic arms and workers within the same physical environment. The main component of the use case is the IoT Gateway, where necessary software runs to acquire information from the indoor-localization system and the robotic motion information is collected and necessary processing on such data streams in order to predict collisions and avoid them by sending necessary control signals to the robotic arms. The IoT Gateway serves as the brain and the entry-point for the CP SoS in the use case and provides necessary information of the current functioning software to the System Administrator to ascertain that no security breaches or manipulation of the software has occurred through the utilisation of components from the ASSURED framework, such as the usage of secure enrolment schemes through Private Certificate Authority, providing traces of the running software through the ASSURED Tracer, as well as security policy management for updated system. A tighter integration with such components will help the user stories provide a concrete base of use secured solutions in industrial spaces for critical solutions in terms of collaborative workspaces where humans are involved with heavy machinery. More detailed information on what sort of threats such an industrial workspace can face is documented in detail within D1.3.

3.2 DETAILED SCENARIOS

3.2.1 BIBA.US.1

As a System Administrator, I want to know the security assurance provided by a software component and assess the risks related to it before being deployed, in order to avoid attacks that can cause system failure or compromise personnel's safety.

User Story Confirmations:

- ✓ *Risk assessment is performed on critical software components (RMT, PLMC, CPA services) so that to identify the risks which will be avoided using the Policy Recommendation. The latter is used to extract the attestation policies which will be instruct the run-time attestation mechanisms to verify the integrity of specified software components.*

ASSURED Functionalities:

- ✓ Risk Assessment, Policy Recommendation Engine, Attack Validation Component, Control-Flow Attestation, Runtime Tracing, Blockchain services, TPM-based Wallet

User Story Implementation:

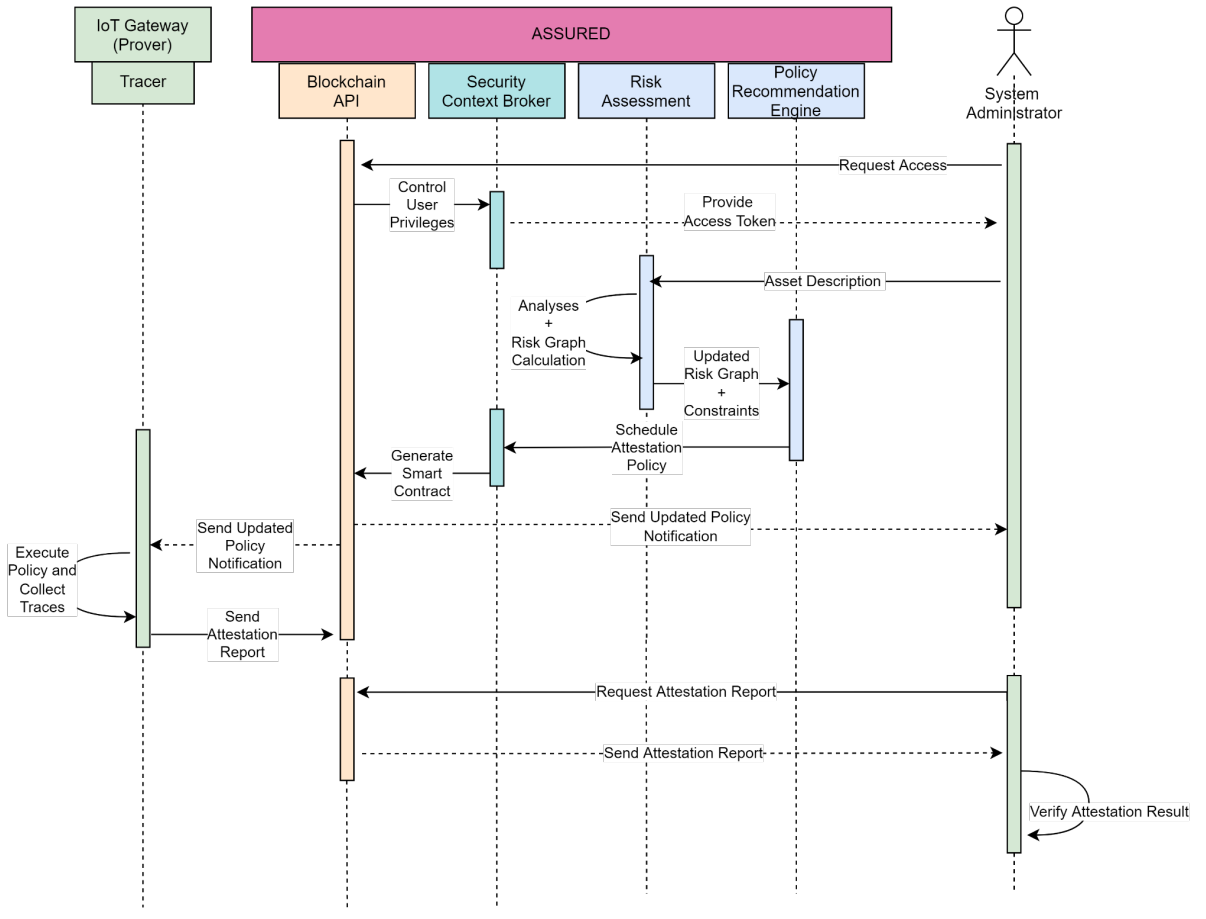

FIGURE 5: *BIBA.US.1 SEQUENCE DIAGRAM*

Workflow:

1 The System Administrator uses the Risk Assessment Dashboard to update / include a software component such as Robot Motion Tracking (RMT), Personnel Localization and Motion Capturing (PLMC), Collision Prediction and Avoidance (CPA) services to be deployed on the trusted device (IoT Gateway).
2 If the System Administrator needs to deploy a monitoring software for the services running on the IoT Gateway, these services (RMT, PLMC, CPA) are presented to the Risk Assessment Dashboard.
3 The Risk Assessment engine generates a Risk-Graph to be evaluated.
4 Post-Assessment an updated Risk-Graph and Constraints are provided to the Policy Recommendation Engine for an actual Attestation Policy which is sent to Security Context Broker.
5 The Broker communicates with the Blockchain infrastructure to generate a new smart contract for the updated attestation policy.
6 This generated new policy is notified to the trust devices IoT Gateway and the device at the System Administrator's end.
7 The Prover then executes the new policy and collects necessary traces and sends an updated report to the Blockchain infrastructure.
8 The System Administrator requests the results of the updated policy.
9 The attestation policies must be updated dynamically for the deployment of RMT, PLMC, CPA deployment.

3.2.2 BIBA.US.2

As a System Administrator, I want to ensure that all the devices' (configuration and execution) log traces are stored and communicated securely, in order for the IoT Gateway to be able to verify the device integrity.

User Story Confirmations:

- ✓ *Successful retrieval of logs / data of RMT, PLMC, CPA services is achieved through the ASSURED Data Storage Engine. The acquisition of the logs engages the use of the ASSURED Tracer and the attestation mechanisms (CFA, CIV), and through the TPM-based wallet the logs are stored and referenced on the Blockchain infrastructure. The system administrator is able to access these logs on the data storage engine, based on proper authorisation (based on ABE) and Searchable Encryption mechanisms. Thus, refusal of access to data / logs for unauthorised users is achieved.*

ASSURED Functionalities:

- ✓ Risk Assessment, Policy Recommendation Engine, Control-Flow Attestation, Configuration Integrity Verification, Runtime Tracing, Blockchain services, TPM-based Wallet

User Story Implementation:

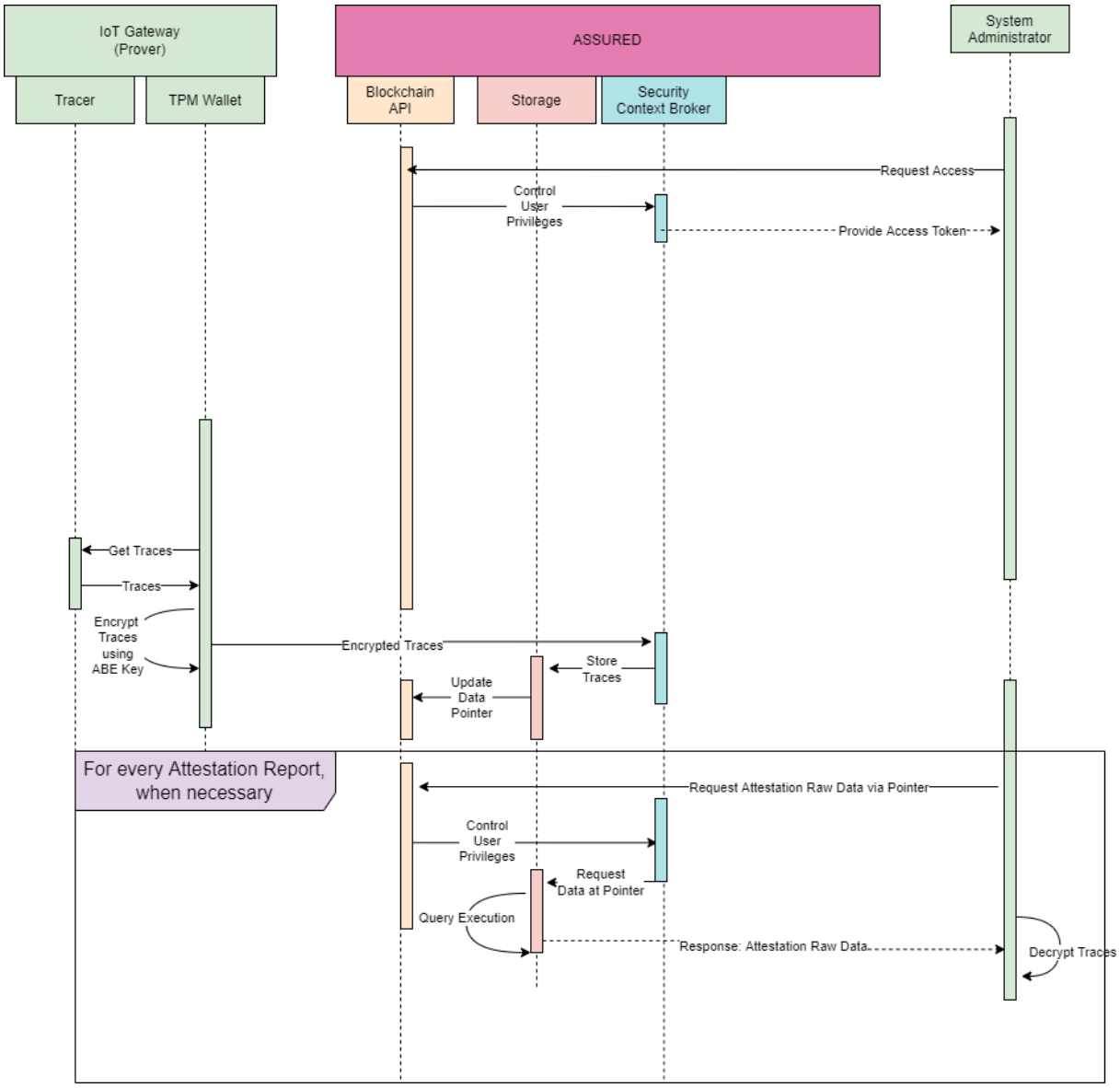

FIGURE 4: *BIBA.US.2 SEQUENCE DIAGRAM*

Workflow:

1 As a way to understand how the system is behaving, the necessary logs and traces from the IoT Gateway in case of accident are generally required to be stored in a persistent manner. These logs and traces help to understand the behaviour of the services such as RMT, PLMC, CPA in case of an attack/accident.

2 The encrypted information is stored on the Off-Chain storage engine via the security context broker.

3 The Storage Engine updates its data pointers.

4 The System Administrator requests an Attestation Report for the updated policy which is presented from the prover (IoT Gateway) along with the data pointers to the updated traces in the Storage Engine.

5 For a user to access these logs, they should be authorised via the Certification Authority from the platform. Via necessary Access Tokens, the user can query the storage engine for actual data and decrypt the information for further usage.

3.2.3 BIBA.US.3

As a System Administrator, I want to continuously monitor run-time execution of the software components on a single device or a network of interconnected devices, in order to attest their correct execution behaviour.

User Story Confirmations:

- ✓ *Successfully Attest and monitor correct execution behaviour of RMT, PLMC, CPA running on IoT Gateway (Raspberry Pi) or multiple IoT Gateway (Raspberry Pi cluster in same network), using CFA, CIV and, when applicable, Swarm attestation.*
- ✓ *If the execution behaviour of the above-mentioned services running in IoT Gateway is modified, then the attestation must be able to report it and take appropriate actions.*

ASSURED Functionalities:

- ✓ Risk Assessment, Policy Recommendation Engine, Attack Validation Component, Control-Flow Attestation, Configuration Integrity Verification, Swarm Attestation, Runtime Tracing, Blockchain services, TPM-based Wallet

User Story Implementation:

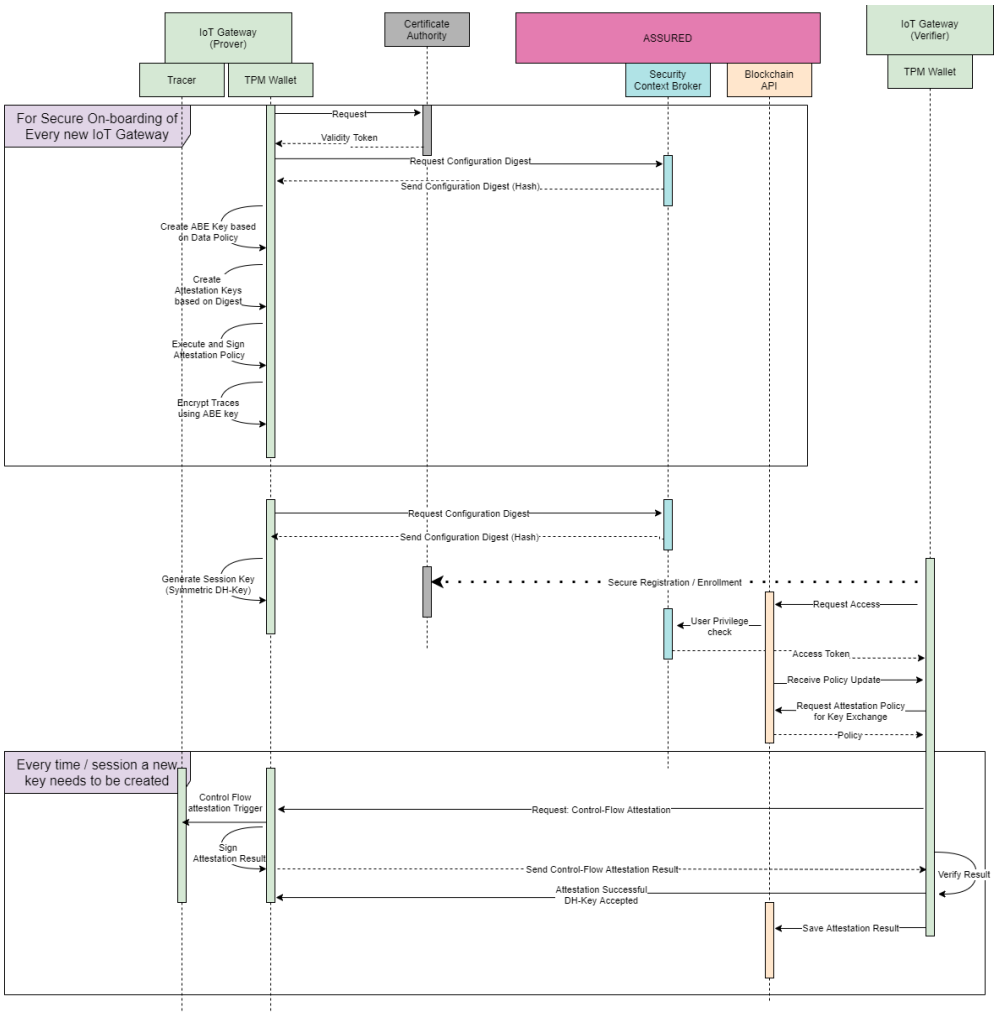

FIGURE 6: *BIBA.US.3 SEQUENCE DIAGRAM*

Workflow:

1 The System Administrator introduces the IoT Gateways into the application area, where secure on-boarding through registration and enrolment request by the IoT Gateways to the Private Certificate Authority
2 The Private Certificate Authority generates a token upon the request. This generated token is used for interacting with the Blockchain CA for issuing access credentials. Then these issued credentials are stored securely in the TPM wallet.
3 Depending on the Verifier (either the System Administrator / IoT Gateway) a request is made to the prover Gateway to trigger a control-flow attestation to the software code keeping track of the data being consumed from the Indoor Localization System
4 The TPM wallet triggers the tracer to begin with the control-flow attestation and the resultant traces are encrypted by the Gateway through the generated ABE key
5 The encrypted traces are sent back either to the System Administrator where the information is decrypted
6 If the request is to be sent to another IoT Gateway seeking Control-Flow Attestation proof for data filter application, metric collection containerised application, then a Symmetric DH-key session key is generated on the prover. Based on the success of the Control-Flow Attestation result, the verifier sends a challenge with an accepted DH-Key and can be used for further communication
7 The verifier sends the attestation result finally to the Blockchain API for storage of traces

3.2.4 BIBA.US.4

As a System Administrator, I want to verify in real-time all interconnected devices against baseline security authentication requirements, in order to monitor the overall trust state of the manufacturing infrastructure.

User Story Confirmations:

- ✓ *All connected trusted devices like IoT Gateway, Data aggregator in the infrastructure must adhere to security and authentication requirements set by the Assured framework by successfully registering and establishing a secure communication channel. This will be achieved using the TPM-based wallet which operates in synergy with the Blockchain infrastructure.*

ASSURED Functionalities:

- ✓ Blockchain services, TPM-based Wallet

User Story Implementation:

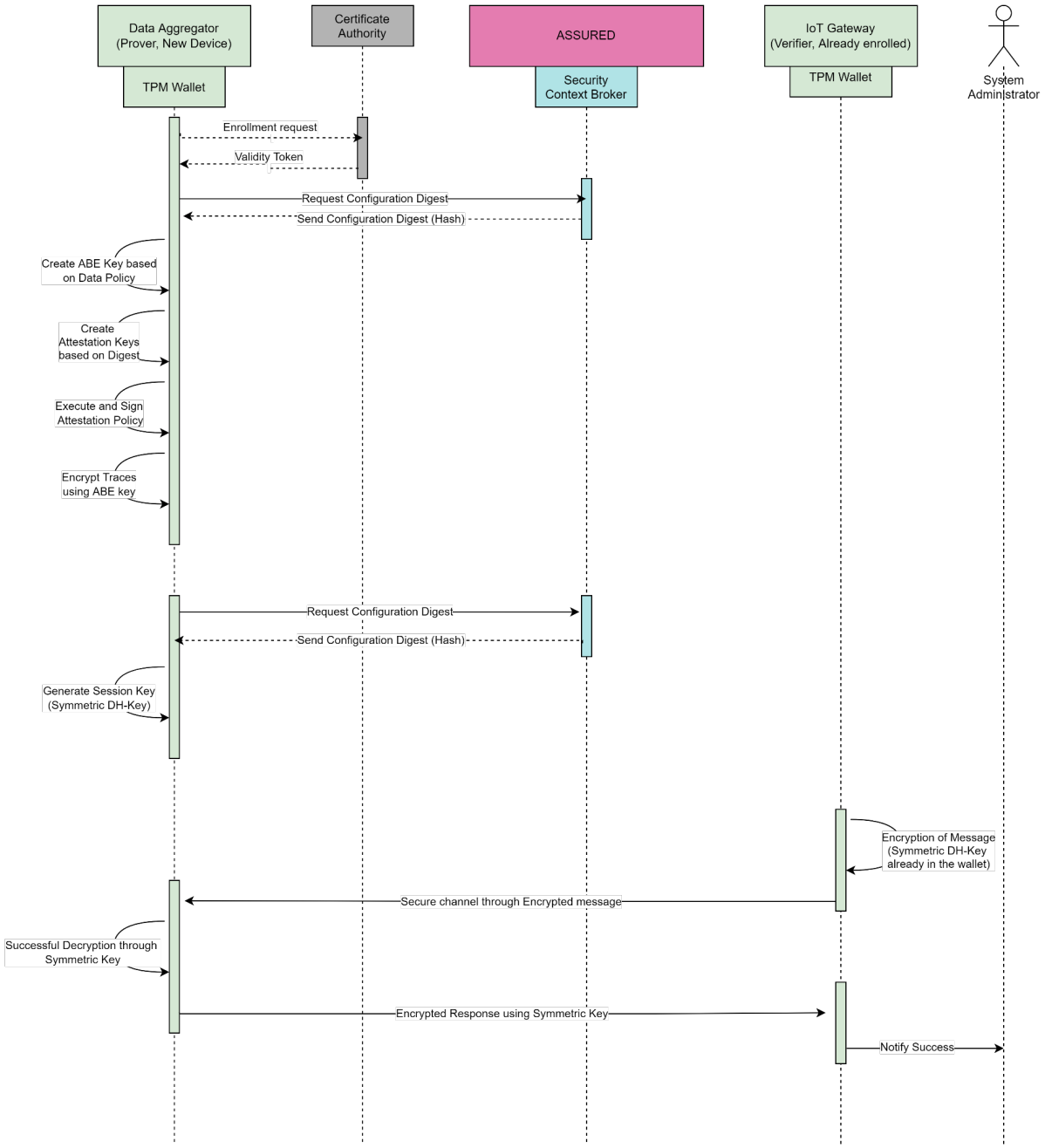

FIGURE 7: BIBA.US.4 SEQUENCE DIAGRAM

Workflow:

1 Newly introduced IoT Gateway with dedicated TPM Wallet is introduced into the smart manufacturing environment with necessary software running on it.

2 IoT Gateway begins with a secure enrolment process by communicating with the ASSURED private certification authority.

3 During the enrolment the necessary configuration digests are obtained and used by the TPM Wallet to generate certificates on IoT Gateway like ABE keys and symmetric DH Keys for session management.
4 In scenarios where another trusted IoT Gateway might require communicating with the prover (Newly enrolled IoT Gateway) the session management keys generated via TPM Wallet and the configuration digest provide necessary encryption/decryption of the data exchange through usage of symmetric DH keys

5 Attestation of successful communication is sent to the system administrator.

3.2.5 BIBA.US.5

As a System Administrator, I want to secure the wired or wireless communication between devices belonging to the same manufacturing environment, in order to establish a trusted management channel with the IoT Gateway.

User Story Confirmations:

- ✓ *Successfully establish a secured communication channel between trusted devices (such as data aggregators) and IoT Gateway (Raspberry Pi) running RMT, PLMC, CPA services. TPM-based wallet will be used to establish the establish encrypted communication channels among the devices.*

ASSURED Functionalities:

- ✓ Control-Flow Attestation, Configuration Integrity Verification, Runtime Tracing, Blockchain services, TPM-based Wallet

Workflow:

1 Secure transmission of data from trusted devices requires Secure Keys within the IoT Gateway.
2 The IoT Gateway requires an ABE Key and Attestation Keys that are used for encrypting traces from the tracer and sending the requested attestation result.
3 A secure initial registration with the Blockchain Private Certification Authority occurs which provides necessary verification tokens for the IoT Gateway that are used to certify the TPM Wallet on the Gateway, once registration occurs the device is enrolled to the Certificate Authority making it a trusted on-boarded device.
4 Based on the enrolment, the Gateway can now generate the required ABE and Attestation Keys for encryption of traces.
5 The keys and the encryption provide a secure method to transmit data over wired/wireless media.

User Story Implementation:

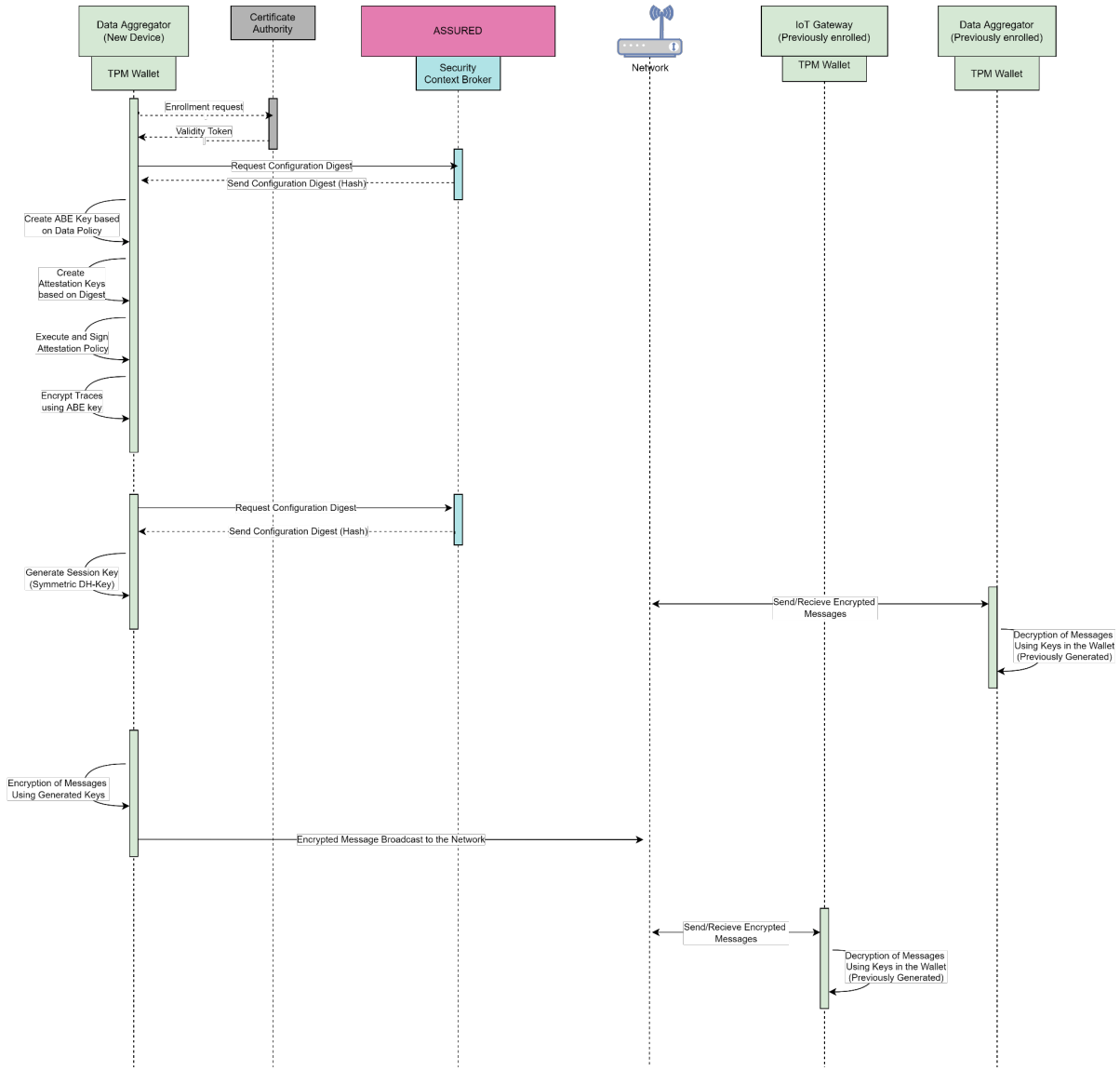

FIGURE 8: *BIBA.US.5 SEQUENCE DIAGRAM*

3.2.6 BIBA.US.6

As a System Administrator I want to securely enrol new devices in my existing manufacturing environment, without the need of any manual intervention, in order to enhance the productivity by scaling services.

User Story Confirmations:

- ✓ *Successfully enrol a new trusted device into the smart manufacturing infrastructure without any manual provisioning. This operation will be confirmed though the use of the TPM-based wallet, Swarm attestation and the qualities of the Blockchain infrastructure.*
- ✓ *If an untrusted device is introduced into the smart manufacturing infrastructure, provisioning must fail, and the device must not be added.*

ASSURED Functionalities:

- ✓ *Swarm Attestation, Runtime Tracing, Blockchain services,TPM-based Wallet*

User Story Implementation:

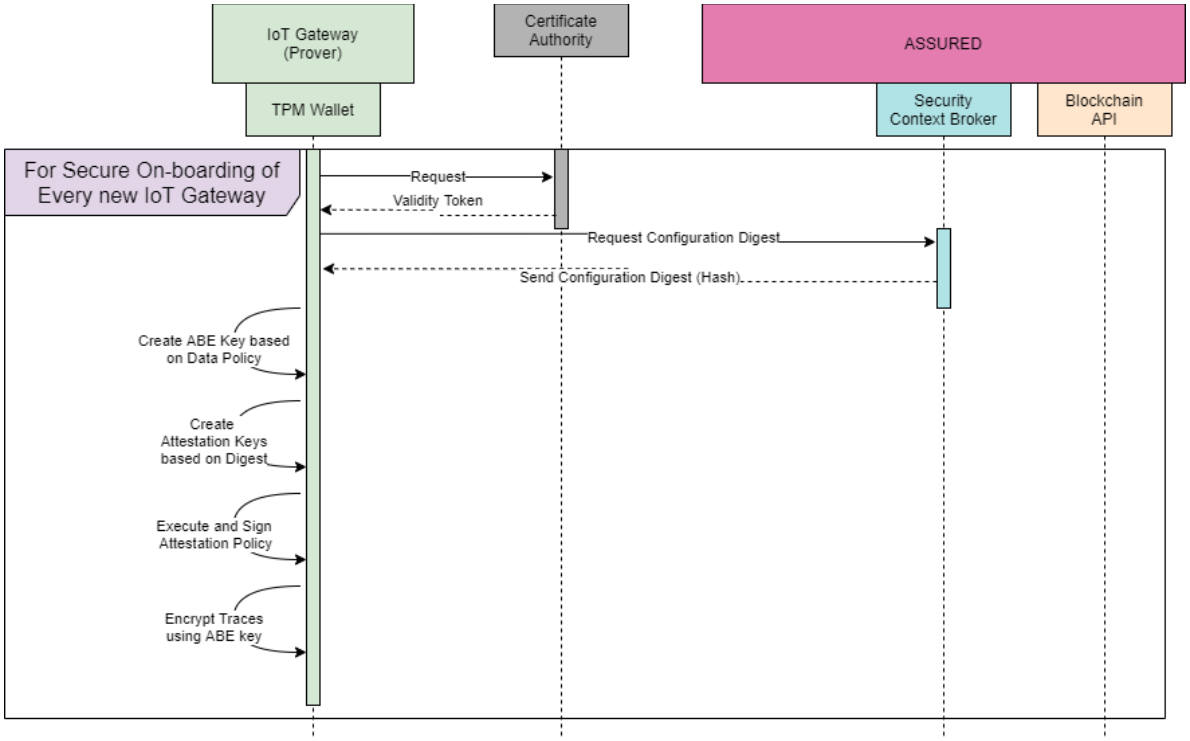

FIGURE 9: *BIBA.US.6 SEQUENCE DIAGRAM*

Workflow:

1 Systems may require introduction of new devices with new services that consume the data generated from the CP SoS within the smart manufacturing case. Instead of executing lots of software on a single hardware instance of IoT Gateway. The admin may need to run Robot Motion Tracking, Personnel Localization Motion Capturing, Collision Prediction Avoidance services on different devices. Thus, the introduction to similar IoT Gateways securely is a primary requirement

2 The System Administrator potentially introduces a new IoT Gateway which requests registration and enrolment to the Blockchain Private Certificate Authority

3 The CA returns a valid token which serves as a secure entity to obtain necessary digests for further configurations

4 The new IoT Gateway requests the Security Context Broker for a configuration digest through the use of the obtained token from the CA

5 The returned Configuration Digest is used by the TPM wallet to generate the ABE key, Attestation Key and furthermore session management keys (Symmetric Diffie-Hellman Keys for inter-gateway secure communication)

6 The System Administrator can request for an Attestation Report from the newly on-boarded IoT Gateway and decrypt the results securely since the on-boarding process was executed in a secure manner using the ASSURED components (CA, SCB)

3.2.7 BIBA.US.7

As an OEM, I want to receive behavioural information about my components deployed in the work area, in order to allow for better understanding and modelling for future enhancements – e.g., optimise mixed-criticality CPS service execution and attestation schedule so as to improve health state information of the entire manufacturing environment and the CAP process chain.

User Story Confirmations:

- ✓ *Successfully query different levels of details in the execution of a device through the Assured framework so the services deployed on IoT Gateway can be enhanced. This is achieved through the use of the Blockchain for the acquisition of data that have been traced from the components, using the runtime tracer.*

ASSURED Functionalities:

- ✓ Risk Assessment, Policy Recommendation Engine, Attack Validation Component, Runtime Tracing, Blockchain services, TPM-based Wallet

Workflow:

1 OEM provides ASSURED Attack Validation Engine with System Description, which has system states, such as motor run variable for a manufacturing robot.

2 Attack Validation Engine applies mutation fuzzing using the provided system description and creates a result of potential vulnerabilities such as unintended activation of the motor of the robot, by manipulating motor run variable.

3 These potential vulnerabilities are forwarded to Policy Recommendation Engine to generate and schedule according to policies, which is then forwarded to Security Context Broker.

4 Security Context Broker updates the Blockchain API in the form of Smart Contract.

5 Blockchain API notifies the OEM IoT Gateways TPM Wallet with the updated policy.

6 As the notification is received, TPM Wallet triggers the Tracer to apply new policies for attestation.

7 For every failed attestation, Tracer sends the traces to the Attack Validation Engine for evaluation.

8 Attack Validation Engine applies mutation fuzzing to the traces to identify new vulnerabilities.

9 Newly found vulnerabilities are forwarded to Risk Assessment Engine to be quantified. This quantified risk is sent to Policy Recommendation Engine to create new policies, which then triggers the whole process again for attestation according to new policies.

10 By utilizing the attestation reports, OEM can be informed about the critical variables of the system such as unwanted motor activation, and enhance the system accordingly, i.e., software updates to secure vulnerable variables.

User Story Implementation:

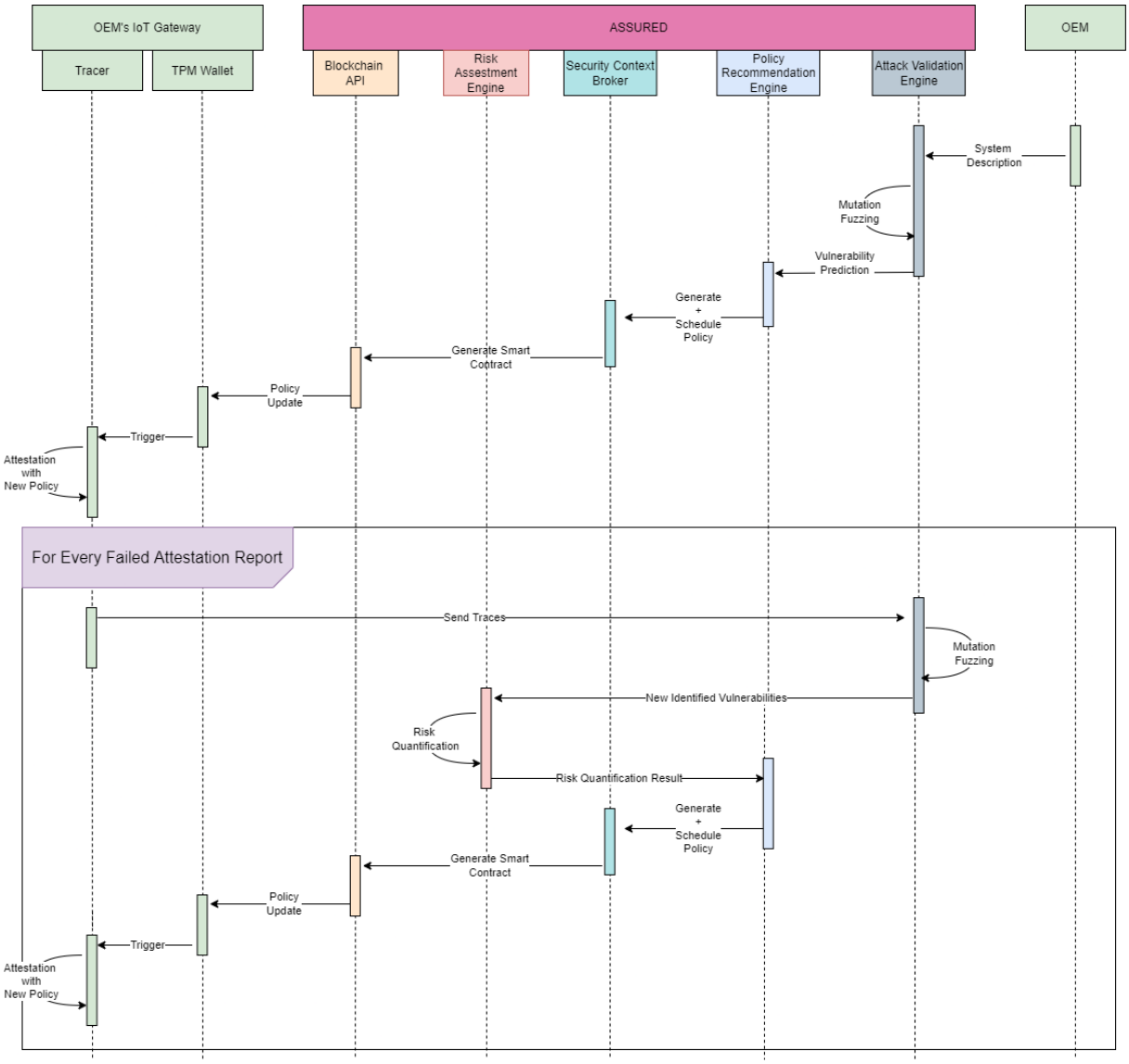

FIGURE 10: BIBA.US.7 SEQUENCE DIAGRAM

3.3 CONDITIONS

For the Smart Manufacturing scenario, BIBA currently provides a virtual demonstrator. The virtual demonstrator consists of the following components:

3.3.1 Data Generator

Data generators are simulation agents for personnel walking and robot performing tasks on a factory floor. Data generators are run on separate PCs and data is exchanged via RabbitMQ broker.

Personnel Walk Data Generator: Simulates positioning tags mounted on personnel using Inertial Motion Generator (IMG) and Sigmoid Walk Angle Generator (SWAG) to produce Noisy 3D Acceleration and Noisy 3D Positional coordinates which are published periodically to Message Broker using Message Client.

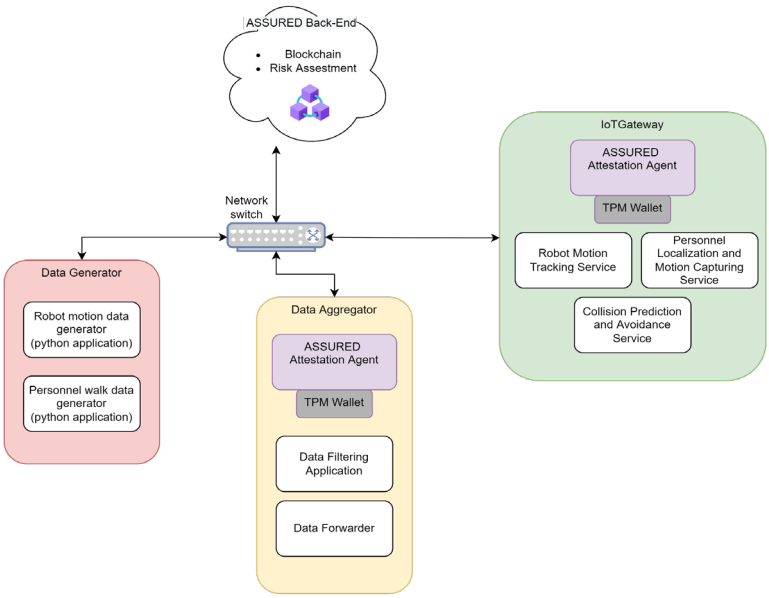

FIGURE 11: *SMART MANUFACTURING ENVISIONED DEMONSTRATOR SETUP*

Robot Motion Data Generator: This simulates working of a physical robotic arm in workspace by generating a motion pattern for Robots both operational and non-operational. For Robots the Motion patterns for end effectors are stored in Lookup. This information is used to obtain joint angles of the robot using the Robot model and applying an inverse kinematic approach. As the motion pattern is fixed and stored up in lookup, thus it is possible to obtain next Joint Angles ahead of time.

3.3.2 Data Aggregator

Here the generated data from the application, coming from the data generator in the case of the virtual demonstrator, is accumulated, filtered and converted to the correct message format. After the conversion to a format IoT Gateway expects, the data is forwarded to the gateway for further processing. An attack on the aggregator, i.e. changing the data that is collected by attacking the filter, can cause faulty processing on the IoT Gateway, which can result in harm to collaborating humans. With integration of the ASSURED framework, the integrity of the collected data is protected. This protection is provided by the components of the framework such as secure enrolment, runtime risk assessment, control-flow attestation, attack validation and enabling secure communication channels through TPM wallet.

3.3.3 IoT Gateway

Robot Motion Tracking (RMT), Personnel Localization Motion Capturing (PLMC) and Collision Prediction and Avoidance (CPA) services run on the IoT Gateway. These services track personnel and robots in a factory floor and with this data performs prediction to avoid any accident. These services are critical in terms of safety and must be protected from the attacker.

Robot Motion Tracking: This service tracks robot arm movement and also provides a future motion path. One instance of this service is assigned to exactly one Robot in the work-place area. The service provides the following: Instantaneous 3D Coordinate of the robot joints, and Future motion path of robot joints ahead of time.

Personnel Localization Motion Capturing: This service provides an optimal estimate of personnel's 3D coordinates and predicts their future motion trajectory time ahead with certain

confidence level in different regions. This service provides following information: Optimal estimate of personnel's instantaneous position, Estimate of personnel's future position/region of presence, small time ahead, and Monitor Node device QoS (Quality of Service) Parameters

Collision Prediction and Avoidance (CPA): CPA subscribes for the Current and Next end effector coordinates of Robots in the workspace from corresponding instances of RMT service. Also, CPA subscribes for the Personnel Coordinate estimates and Predicted Occupancy Coordinates of Personnel in the workspace from corresponding instances of PLMC service. Then CPA combines this information and uses a probabilistic algorithm to predict the probability of collision between a given personnel and robot in a work-place area ahead of time. If the possibility of collision is detected, based on likelihood, safety distance and velocities of approaching Personnel and Robot, CPA service either slows down the robot or stops the Robot by sending appropriate control signals to PLC via IPC (Inter Process Communication).

3.4 KPIS AND ACCEPTANCE CRITERIA

3.4.1 Quantitative Metrics

ID	Metric	Target Value	Acceptance Criteria	(M)andatory / (G)ood to Have / (O)ptional	1st Release / 2nd Release
BIBA-QUAN-01	Devices in the infrastructure whose configuration and execution integrity is monitored by the IoT Gateway	100%	100%	M	1st and 2nd Release
BIBA-QUAN-02	Devices in the infrastructure whose integrity status are hidden.	0%	0%	M	1st and 2nd Release
BIBA-QUAN-03	Attestation Process Time (single device) for increasing code complexity	CIV < 800 ms CFA < 2 sec CFA (with ML) < 1 sec	3-5s	G	1st and 2nd Release
BIBA-QUAN-04	Attestation process time for swarm of devices	< 2 sec	< 5sec (based on the number of devices in the swarm)	G	2nd Release
BIBA-QUAN-05	Zero-touch Device secure registration and on-boarding	< 500 ms	<850 ms	M	1st Release
BIBA-QUAN-06	Blockchain API Response Time (single-device)	<1s	1-2s	G	2nd Release

BIBA-QUAN-07	Delay between Attestation Request and Response (multiple-devices)	<2s	4-5s	G	1st Release
BIBA-QUAN-08	Resource Consumption for Encryption / Decryption of traces (CPU)	<30%	<40%	M	2nd Release
BIBA-QUAN-09	Deterministic latency for secure data transfer from devices to IoT Gateway	< 1 sec	< 1sec	M	1st and 2nd Release
BIBA-QUAN-10	Number of motion tracking events securely managed	Such messages are sent in a frequency between 1 and 10 Hz (need to be supported by ASSURED ABE Encryption)	Support the default rate of messages to be exchanged	O	2nd Release

TABLE 7: *SMART MANUFACTURING REFERENCE SCENARIO – QUANTITATIVE METRICS OF SUCCESS*

3.4.2 Qualitative Metrics

ID	Metric	Target Value	(M)andatory / (G)ood to Have / (O)ptional	1st Release / 2nd Release
BIBA-QUAL-01	Trusted / Attested Device	Success/ Fail	M	1st and 2nd Release
BIBA-QUAL-02	Basic Trace Logs From Trusted Device	Visible Encrypted Logs	G	1st and 2nd Release
BIBA-QUAL-03	Easy-to-Use Blockchain API for Interaction	Supported	M	2nd Release
BIBA-QUAL-04	Ease of deployment of a trusted device.	Supported	M	1st and 2nd Release
BIBA-QUAL-05	Secure data sharing between registered, authenticated users with the appropriate access control attributes	Membership Service for hierarchical role-based access control (O(sec))	M	2nd Release

TABLE 8: *SMART MANUFACTURING REFERENCE SCENARIO – QUALITATIVE METRICS OF SUCCESS*

4 SECURE COLLABORATION OF "PLATFORMS-OF-PLATFORMS" FOR ENHANCED PUBLIC SAFETY DEMONSTRATOR

4.1 SECURE COLLABORATION OF "PLATFORMS-OF-PLATFORMS" FOR ENHANCED PUBLIC SAFETY

As summarised in the table below, the public safety testbed is focusing on several aspects referring to city systems that aim to monitor potential events against public safety and also to secure the city systems and actors/beneficiaries against cyber-attacks. The use cases are formulated in order to demonstrate crucial aspects of the above procedures starting from the authentication of devices or the access rights of a user, moving to the monitoring of the data flows and health state assets and finally, ensuring the notification of city officers and execution of mitigation actions. The challenges to be tackled in this specific demonstration referring mainly to the time-wise response and notification as well as to the foremost issue of citizens' safety and citizens' data security that are managed within city-systems. The above has resulted in 7 use cases as presented in D1.1 and more analysed in the sections below.

User Story	Security Properties	Functionalities
DAEM.US.1	Secure access to public data. Operational assurance of systems. Validation of data flows. Identification of public safety events ensuring user privacy. ➢ Operational Assurance ➢ Privacy ➢ Data Integrity ➢ Confidentiality of net. Com. ➢ Continuous authentic. & Authorisation	✓ Risk Assessment ✓ Policy Recom. Engine ✓ Control-Flow Attestation ✓ Config. Integrity Verif. ✓ Direct Anon. Attestation ✓ Blockchain services ✓ TPM-based Wallet
DAEM.US.2	Hub for public safety and security to support the decision making of Operators. The system requires accounts management, user authentication and access control. ➢ Operational Assurance ➢ Privacy ➢ Data Integrity ➢ Confidentiality of net. Com. ➢ Continuous authentic. & Authorisation	✓ Config. Integrity Verif. ✓ Direct Anon. Attestation ✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet
DAEM.US.3	Alert of potential attacks using risk assessment feature. Ensure the operational assurance of critical components and receive annotated alerts. ➢ Operational Assurance ➢ Data Integrity	✓ Risk Assessment ✓ Policy Recom. Engine ✓ Control-Flow Attestation ✓ Config. Integrity Verif. ✓ Direct Anon. Attestation ✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet

DAEM.US.4	Mitigation mechanisms at a device level for incidents' counteraction, though the use of attestation. ➢ Operational Assurance ➢ Data Integrity ➢ Confidentiality of net. Com.	✓ Risk Assessment ✓ Policy Recom. Engine ✓ Control-Flow Attestation ✓ Config. Integrity Verif. ✓ Swarm Attestation ✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet
DAEM.US.5	Ensure data trustworthiness, secure data flows and access control for first responders. ➢ Privacy ➢ Data Integrity ➢ Confidentiality of net. Com. ➢ Continuous authentic. & Authorisation	✓ Direct Anon. Attestation ✓ Blockchain services ✓ TPM-based Wallet
DAEM.US.6	Policy enforcement and compliance for city operators through a dashboard to monitor the health state of municipal assets. Reports on cyber-security attacks and set of potential countermeasures. ➢ Operational Assurance ➢ Data Integrity ➢ Confidentiality of net. Com.	✓ Risk Assessment ✓ Policy Recom. Engine ✓ Attack Validation Comp. ✓ Control-Flow Attestation ✓ Config. Integrity Verif. ✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet
DAEM.US.7	Real-time secure notifications on security incidents. Support of real-time mitigation decision. Responsiveness of city systems in case of a cyber-attack incident. ➢ Operational Assurance ➢ Data Integrity ➢ Confidentiality of net. Com.	✓ Risk Assessment ✓ Policy Recom. Engine ✓ Control-Flow Attestation ✓ Config. Integrity Verif. ✓ Swarm Attestation ✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet

TABLE 9: REFERENCE SCENARIO 2 USER STORIES SUMMARY

4.1.1 Planning

The public safety use case testbed has identified seven user stories and an elicitation for the first release has been completed according to the feasibility of the first version on the pilot site and the technical advancements of ASSURED. **User stories #1, 4, 5 and 7 will be tested in the second release while user stories #2, 3 and 6 in the first release**.

For the first release, the following user stories will be validated:

ID	User story	Validations
DAEM.US.2	As an Internal Operator (city official), I want **to have a complete, usable toolkit in order to create a hub of public safety and security for my city** with specified accounts and access control roles assigned to relevant stakeholders.	Each new user or device is securely authorised before granted access to the system (using the cryptographic offering of the ASSURE

		Blockchain and the TPM-based wallet), all assets under the city system are included in the toolkit and monitored, through security assessments.
DAEM.US.3	As a city System Administrator, I want **to have an alert of potential attacks** through the risk assessment feature, in order to perform corrective actions at real-time.	Generation of notifications in case of an attack or violation of systems' integrity considering the defined KPIs.
DAEM.US.6	As an Internal Operator (security policy officer) I want **to have a policy enforcement and policy compliance monitoring tool** in order to review and confirm the health state of municipal assets (e.g., Devices) while proposing a set of potential countermeasures against baseline cyber-security attacks.	An instantiation of the environmental assets in the risk assessment engine is available, annotated according to their risk level. The policy recommendation engine is used to define attestation policies to be placed on systems as mitigation actions.

TABLE 10: *SMART CITIES REFERENCE SCENARIO FIRST RELEASE OVERVIEW*

In the following section focus is given to the user stories of the first release where sequence diagrams, workflows and detailed descriptions are included. The rest are mentioned in summary.

4.1.2 Description and User Stories

The public safety testing for city services in Athens, focuses on scenarios that aim to simulate real-time attacks on city systems. The demonstrator will test the attestation on devices and user roles in existing systems, indicatively the security state of components like sensors, cameras etc. Data sharing and verification of users of the system is tested through Blockchain especially for external stakeholders such as LEAs, so as to ensure privacy and security of data flows. Attestation mechanisms will be challenged with specific security attacks when security policies will also be re-configured following the risk assessment. The ASSURED framework in the testbed will be evaluated in terms of its components, while also the imposed prevention against attacks as described in D1.3 such as unauthorised access, control flow, code injection etc.

4.2 DETAILED SCENARIOS

4.2.1 DAEM.US.1

As an Internal Operator (city official), I want to ensure trustworthiness of shared information flows among diverse stakeholders while protecting the privacy of citizens' data, in order for municipal entities to be able to correctly identify any hazardous events (for public safety) without impeding user privacy.

User Story Confirmations:

- ✓ *Data generation and manipulation in the context of smart city is spread across multiple locations accessed by multiple stakeholders. It is of paramount importance to*

secure access to public data that may include sensitive citizens' information while ensuring the overall secure functionality of the system and its constituent components. The trusted sharing of data will be achieved through the use of Blockchain, supported by cryptographic tools such as ABE and Searchable encryption.

ASSURED Functionalities:

- ✓ *Risk Assessment, Policy Recommendation Engine, Control-Flow Attestation, Configuration Integrity Verification, Direct Anonymous Attestation, Blockchain services, TPM-based Wallet*

User Story Implementation:

- ASSURED technology will provide the option of authorization of diverse user roles and input of data flows. The verification engine will play a vital role in monitoring the trustworthiness of each data source deriving from different stakeholders and other bodies collaborating with the city.
- All shared information will be protected, ensuring data privacy of citizens using ASSURED's lightweight cryptography at a device level. ASSURED Trust Enhancing Blockchain technology and smart contracts for Secure Data Sharing and Policy enforcement.

As this user story (DAEM.US.1) is to be demonstrated at the 2nd iteration of demonstrations, the detailed implementation will be analysed at a later stage.

4.2.2 DAEM.US.2

As an Internal Operator (city official), I want to have a complete, usable toolkit in order to create a hub of public safety and security for my city with specified accounts and access control roles assigned to relevant stakeholders.

User Story Confirmations:

A city official/administrator will be able to monitor the state of the environment through the dashboard of the risk assessment engine. The management of user roles and access control will be achieved through the use of the Blockchain and the enforcement of access control policies based on smart contracts.

- ✓ *The complete ASSURED framework will enable pilot benchmarks executing security experiments in the format of a usable toolkit at a laboratory setting.*
- ✓ *ASSURED authentication mechanisms will ensure the secure management of different users and assess access control rights for each individual who is granted access to smart city data.*
- ✓ *Authentication through Blockchain access control, searchable encryption, credentials handling, certificates for accounts/roles*
- ✓ *TPM-based wallet (2nd release)*

ASSURED Functionalities:

- ✓ Config. Integrity Verification, Direct Anonymous Attestation, Runtime Tracing, Blockchain services, TPM-based Wallet

User Story Implementation:

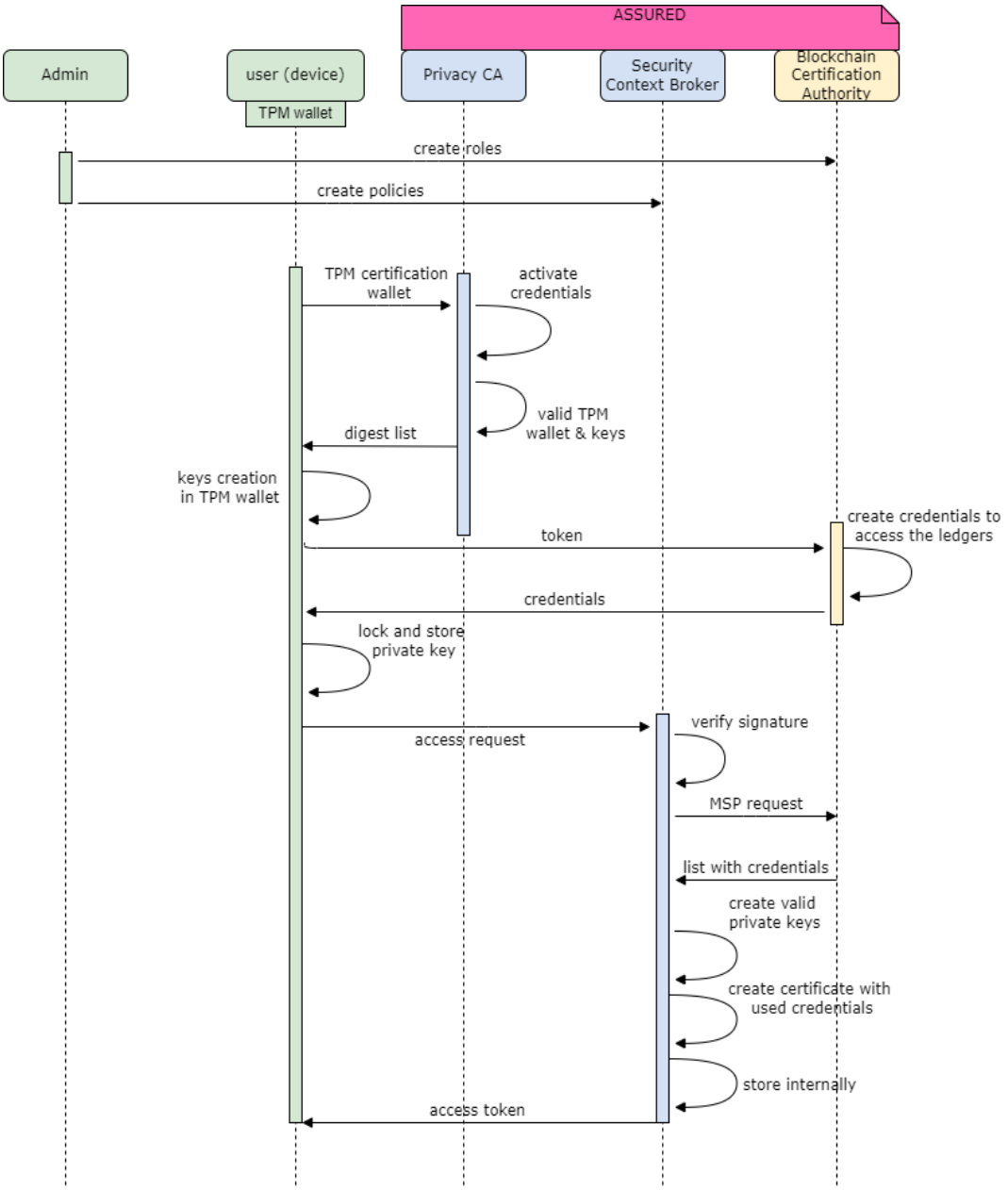

FIGURE 12: *DAEM.US.2 SEQUENCE DIAGRAM*

Workflow:

As depicted in the above diagram this user story initiates with a user or a device e.g., a smoke detector requesting access to the system e.g., new personnel in the internal operators' team. In the workflow diagram the sequence of actions for a user is presented, since we resume that the system devices are already verified.
Prior to the above an internal operator or administrator has already created the access control policies at the Security Context Broker and the appropriate Roles at the Blockchain. Then the following steps take place:

1 A user sends a TPM certification request to the privacy CA that is responsible for the secure enrolment of users in order to certify the TPM wallet.

2 The Privacy CA certifies the TPM wallet of the user, after it has received the digest list, and as the keys are valid and then it activates the user's credentials. The keys are based on the access control policies.
3 The CA sends back to the user the policies (digest list).
4 The user creates the keys in its TPM wallet according to the digest list. The User receives the token and then sends it to the "Blockchain CA"
5 The Blockchain creates credentials to access the Ledgers.
6 The created credentials, these are sent back to the User where the private key is loaded and stored in the TPM Wallet.
7 The User tries to access either the public or private ledger (depending on the query he/she wants to make) by making an Access Request to the SCB.
8 The SCB then checks the access privileges (through ABAC) and verifies the token's signature and forwards a request to the Blockchain certification authority through MSP in order to receive back the list of attributes.
9 The Blockchain certification authority creates the attributed-based credentials and sends back the list credentials to the SCB.
10 The latter creates valid private keys and a certificate including the user's attributes and stores them internally.
11 The SCB replies back to the User with an access token. The keys are sent to the user and stored to the user TPM wallet.

Finally, there is a differentiation in case a user aims to access a private or public ledger e.g a city system admin or a first responder. For the first case the user directly goes to the Blockchain API and makes an attestation report query. While for a public ledger, the user performs a query to the SCB that must include searchable encryption. In the diagram depicting the workflow the process included is the access request to the SCB. Both scenarios are finalised with the same step where the access token is received by the user.

4.2.3 DAEM.US.3

As a city System Administrator, I want to have an alert of potential attacks through the risk assessment feature, in order to perform corrective actions in real-time. Also, I want to have a dashboard where I can overview the annotated components of the system that are classified according to their risk potentiality.

User Story Confirmations:

- ✓ *The risk assessment dashboard will address the need of the administrator. A hierarchical approach is followed on the annotation (high or low risk), in order to help prioritise corrective actions and countermeasures. Corrective actions will be instructed as attestation policies in order to be enforced by the policy recommendation and the Security context broker. The risk assessment dashboard will be used of the expression of attestation tasks.*

ASSURED Functionalities:

- *Risk Assessment, Policy Recommendation Engine, Control-Flow Attestation, Configuration Integrity Verification, Direct Anonymous Attestation, Runtime Tracing, Blockchain services, TPM-based Wallet*

User Story Implementation:

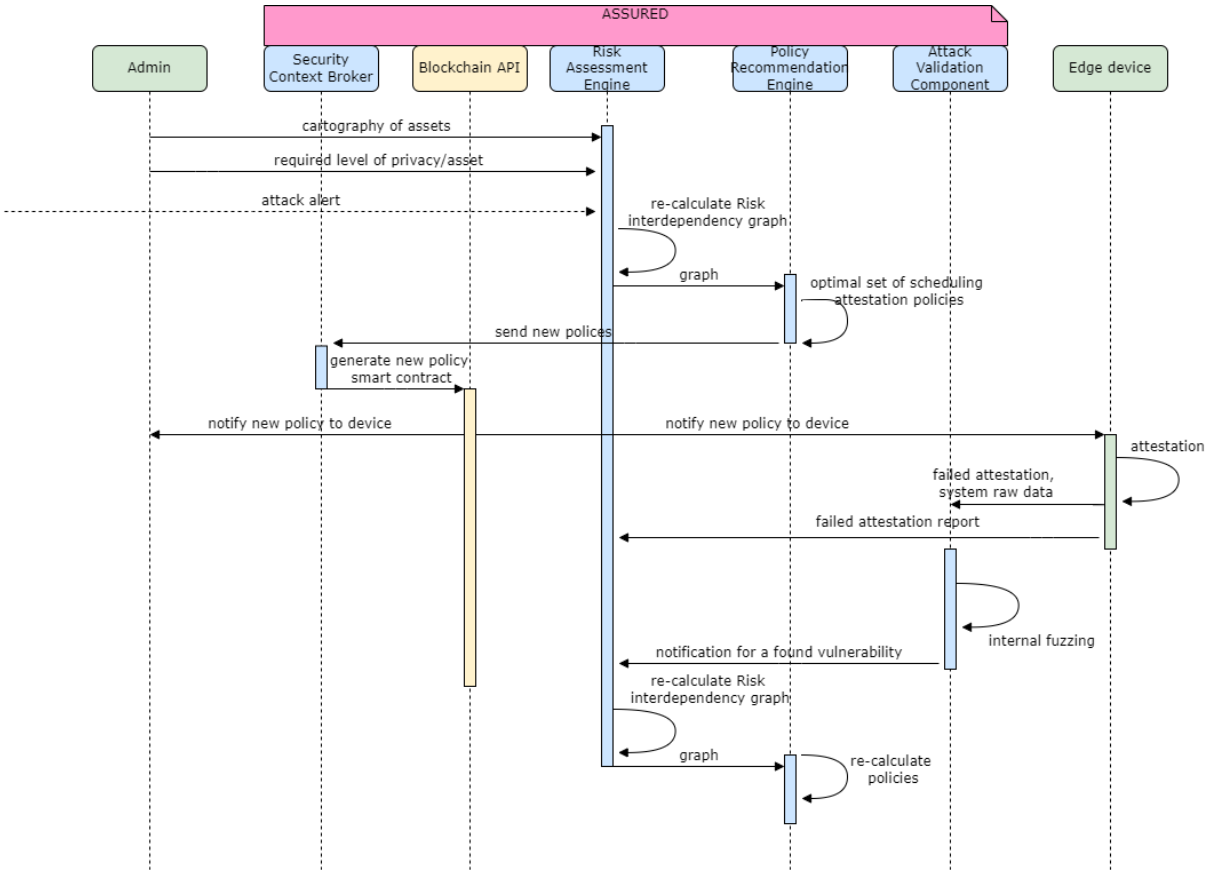

FIGURE 13: *DAEM.US.3 SEQUENCE DIAGRAM*

Workflow:

The attack validation component will not be deployed for the 1st release. The workflow above refers to the alert for attack while the risk assessment process is considered to be already executed. In this user story initially the system administrator provides to the Risk Assessment Engine, the information of the asset cartography and provides the required safety level per asset to the Risk Assessment Engine. The workflow of actions is launched with a risk alert is received for a potential attack to the city systems:

1 The Risk Assessment Engine re-calculates the existing risk interdependencies graph according to the identified vulnerabilities.
2 The graph is sent as an input to the policy recommendation engine that optimises secure attestation policies and orders execution. These policies are sent to the Security Context Broker as responsible for deploying them to the Blockchain.
3 Once deployed, the Blockchain API informs all the registered devices for these new policies.
4 The devices perform a query to the Blockchain API to read those policies that are intended for them and execute the attestation policies.
5 The output is enforced and deployed through ASSURED Blockchain to the edge devices.
6 Each device runs the attestation and in case of a failed attestation, this failed attestation report is sent to the RA.
7 The AVC identifies the type of vulnerability and sends this input to the RAE in order to re-calculate the risk interdependencies graph.
8 The updated graph is sent to the PRE for the re-calculation of the new policies.

9 The "system raw data" from the failed attestation are forwarded to the Attack Validation Component for further processing based on the initial system description that was given by the System Administrator.
10 The Attack Validation component executes internal fuzzing and in the case it identifies no vulnerability, it notifies the RA for recalculating the risk graph and attack path calculation.

4.2.4 DAEM.US.4

As an Internal Operator (city IT operator), I want to have a list of mitigation mechanisms at a device level, in order to implement countermeasures in case of an incident.

User Story Confirmations:

- ✓ *City IT operator should be aware of a set of possible countermeasures, actions and mitigation mechanisms offered in order to protect its system from attacks at the device level. That is, the operator will be in position to instruct the attestation of systems and critical services as part of the risk assessment environment.*

ASSURED Functionalities:

- ✓ *Risk Assessment, Policy Recommendation Engine, Control-Flow Attestation, Configuration Integrity Verification, Swarm Attestation, Runtime Tracing, Blockchain services, TPM-based Wallet*

User Story Implementation:

As this user story (DAEM.US.4) is to be demonstrated at the 2nd iteration of demonstrations, the detailed implementation will be analysed at a later stage.

4.2.5 DAEM.US.5

As an External Member (first responder), I want to have a report on the trustworthiness of data flows in the city systems, in order to support the public safety procedure in any case.

User Story Confirmations:

Non-technical city collaborators (such as first responders) have access to data during public events. From the first responder point of view, there is a need to secure access to the data while the origins of the incoming data need to be validated and confirmed. This is achieved via the sharing of attestation reports and metadata offered along with the operational data of interest though the Blockchain. The trustworthiness of the devices and data is confirmed through the used on the TPM-based wallet and the Blockchain infrastructure. Thus, core enablers for this user story are:

- ✓ *ASSURED Blockchain infrastructure and trusted on- and off-chain data and knowledge management services including user authentication, access authorization and the ASSURED Blockchain Wallets for continuously attesting and assessing the security of all involved devices in a privacy-preserving manner.*
- ✓ *ASSURED lightweight cryptography solutions for data protection (ABE, Searchable encryption)*

ASSURED Functionalities:

- ✓ *Direct Anon. Attestation, Blockchain services, TPM-based Wallet*

User Story Implementation:

As this user story (DAEM.US.5) is to be demonstrated at the 2nd iteration of demonstrations, the detailed implementation will be analysed at a later stage.

4.2.6 DAEM.US.6

As an Internal Operator (security policy officer) I want to have a policy enforcement and policy compliance monitoring tool in order to review and confirm the health state of municipal assets (e.g. Devices) while proposing a set of potential countermeasures against baseline cyber-security attacks.

User Story Confirmations:

- ✓ *The risk assessment dashboard will be available to city security policy officers for monitoring the health state of municipal assets as a whole. In case of baseline cyber-security attacks (e.g., sensor compromise, system unauthorised access), a report is produced after the application of attestation policies as countermeasures. It is crucial for the policy officer to be able to propose different strategies in order to ensure public safety and sustainability of the city under cyber-security attacks. Different attestation schemes are provided to the security policy officer as potential countermeasures.*

ASSURED Functionalities:

- ✓ *Risk Assessment, Policy Recommendation Engine, Attack Validation Component, Control-Flow Attestation, Configuration Integrity Verification, Runtime Tracing, Blockchain services, TPM-based Wallet*

Workflow:

The attack validation component will not be deployed for the 1st release. The workflow refers to the alert for attack while the risk assessment process is considered to be already executed. The scope of this workflow diagram is to depict two main functionalities, namely the ability of the operator to query for different attestation results, in order to monitor the health state of the assets, and also the verification and certification of the correct execution of the attestation process based on the deployed smart contracts.

1 The policy recommendation engine sends the policies to the security context broker that then sends a request (chain code for the attestation policy) for a contract to the smart contract composition engine.
2 The contract is sent to the SCB and is forwarded to the Blockchain peer.
3 The peer deploys the smart contract policies and sends them to the private ledger.
4 When the new policy is deployed, all edge devices are notified for the new contract.
5 It is assumed that one device is the verifier and another the prover. The verifier device that receives a new policy result, verifies it and sends the result of the attestation policy to the Blockchain peer.
6 The prover executes the policy and sends the attestation results to the Blockchain, where the verifier queries the results for the new policy attestation result and receives it.
7 The result of the attestation is sent by the Verifier to the Blockchain Peer. Then, this will be recorded on the private Ledger.

User Story Implementation:

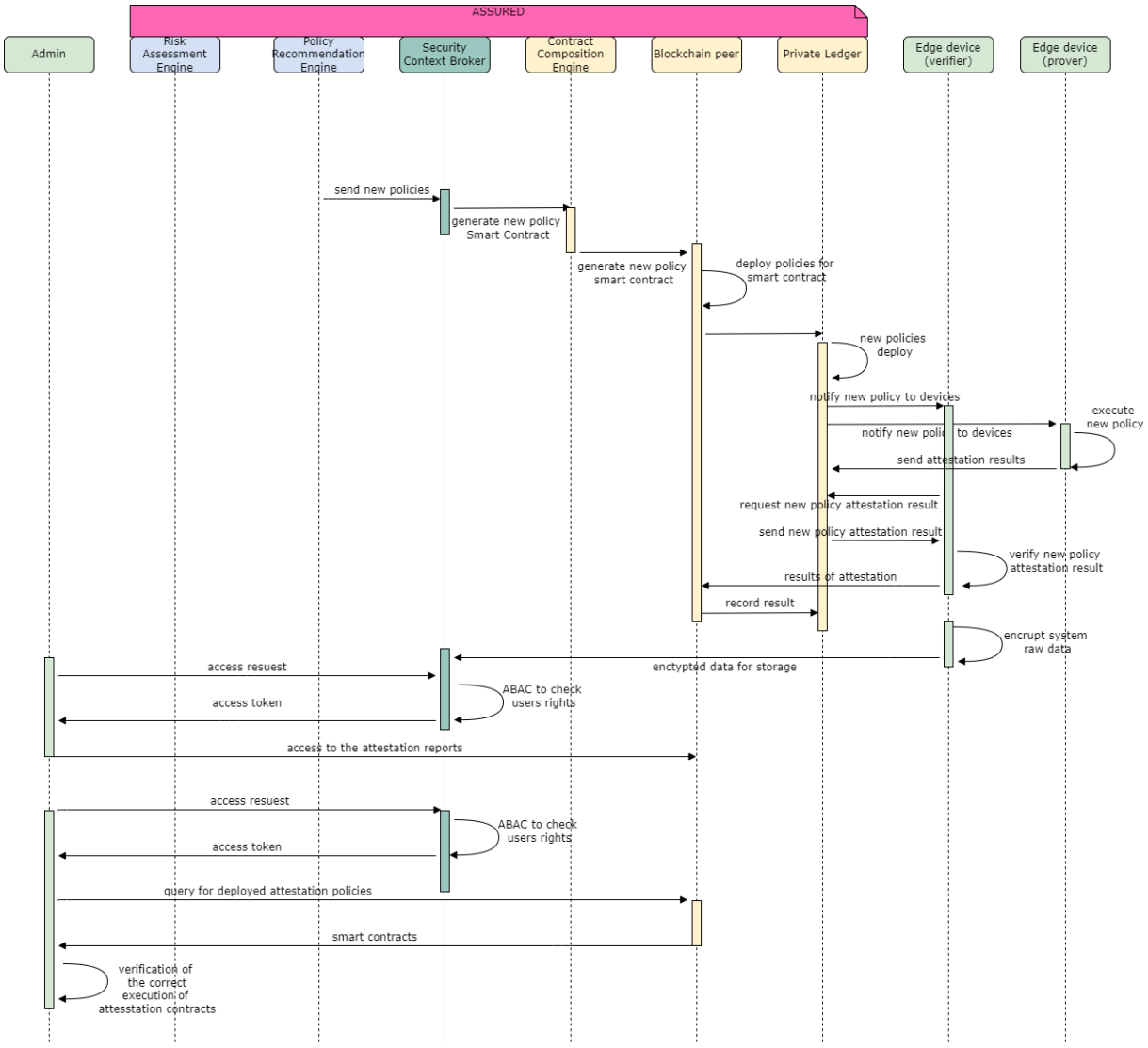

FIGURE 14: DAEM.US.6 SEQUENCE DIAGRAM

8 The policy recommendation engine sends the policies to the security context broker that then sends a request (chain code for the attestation policy) for a contract to the smart contract composition engine.
9 The contract is sent to the SCB and is forwarded to the Blockchain peer.
10 The peer deploys the smart contract policies and sends them to the private ledger.
11 When the new policy is deployed, all edge devices are notified for the new contract.
12 It is assumed that one device is the verifier and another the prover. The verifier device that receives a new policy result, verifies it and sends the result of the attestation policy to the Blockchain peer.
13 The prover executes the policy and sends the attestation results to the Blockchain, where the verifier queries the results for the new policy attestation result and receives it.
14 The result of the attestation is sent by the Verifier to the Blockchain Peer. Then, this will be recorded on the private Ledger.
15 The system raw data is encrypted by the Verifier (using its ABE Key of the TPM Wallet) and forwarded to the SCB for further secure storage in the ASSURED Data Storage Engine.
16 The admin sends an access request to the SCB

17 The SCB performs the ABAC to check its privileges and send back to the admin an access token.
18 This token is then used for accessing and making a query about the health state of specific devices either at the private or public ledger.
19 The admin checks and verifies the correct execution of the smart contracts according to the attestation policies. Thus, the admin queries an access request to the SCB who will return an access token if successful.
20 The admin will then query for the deployed attestation policies as smart contracts through the Blockchain API.
21 Once he/she gets the contracts and the results, it then verifies the correct execution of the attestation contracts.
22 If the above procedure is correct, then the admin certifies the correct enforcement of all attestation policies.

4.2.7 DAEM.US.7

As an Internal Operator (city official), I want to be notified in real-time on security incidents, in order to support on-the-fly mitigation decisions.

User Story Confirmations:

- ✓ *The cyber resilience of a smart city system depends on its effective real-time responses against cyber-security incidents. To that end, a trustworthy notification system is necessary for city officials supported by verification mechanisms for the trustworthiness and legitimacy of the notification message. The risk assessment engine operates during run-time and acquires information upon the validation of attack incidents and compromised devises as a result of failed attestation on critical devices and services.*

ASSURED Functionalities:

- ✓ *Risk Assessment, Policy Recommendation Engine, Control-Flow Attestation, Configuration Integrity Verification, Swarm Attestation, Runtime Tracing, Blockchain services, TPM-based Wallet*

User Story Implementation:

As this user story (DAEM.US.7) is to be demonstrated at the 2nd iteration of demonstrations, the detailed implementation will be analysed at a later stage.

4.3 CONDITIONS

The technical setup of the Smart Cities demonstrator includes the entities depicted in the above schema. Initially the edge devices for the face and smoke recognition scenarios generate video-streams and sensor data received by access points of data collection. Specifically, the cameras are a set of four 5G devices with CCTV recorder including a CMS platform. The smoke sensors are MQ-2 Gas Sensors with an application of a gas leak detector integrated through a C process with the RpP[1]. The collected data referring to the recognition of face and the detection of smoke/fire are shared through a Raspberry Pi 3 and switches/routers. The ASSURED framework ensures secure and efficient cryptography and data flows. The

[1] https://www.safefiredirect.co.uk/blog/project-one-raspberry-pi-connected-wireless-smoke-alarm.aspx

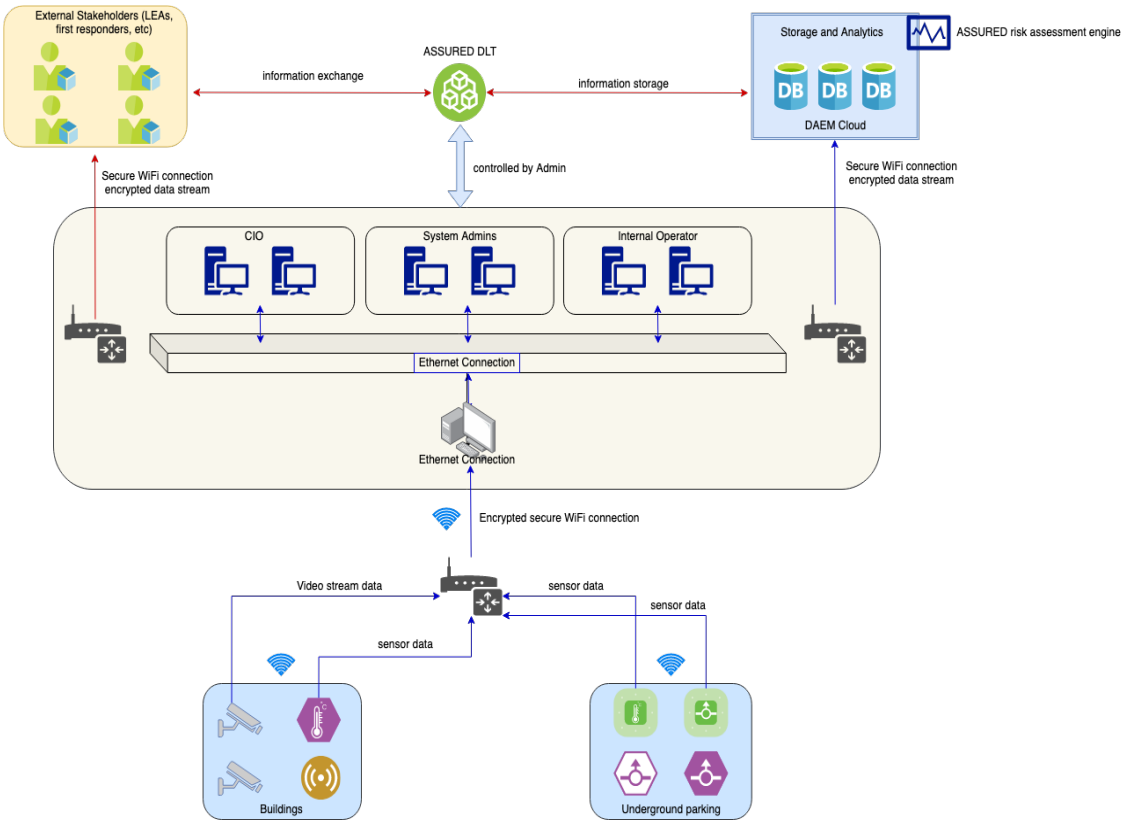

FIGURE 15: *ENHANCED PUBLIC SAFETY ENVISIONED DEMONSTRATOR SETUP*

operational center monitors the data streams received from the edge-devices for risk analysis on potential threats. An interface is shared with external stakeholders such as first responders in case of incidents.

In order to provide an overview of the demonstrator to be implemented in terms of edge devices the following schema presents the testing venue of Serafeio Complex indicating the position of entrances to the facilities and buildings included in the Complex for the public and the employees, as well as the underground parking area. In the first position are the estimated location of the cameras while in the second are located the gas sensors, also depicted in the schema by the respective icon.

FIGURE 16: *SERAFEIO COMPLEX TESTING VENUE*

4.4 KPIS AND ACCEPTANCE CRITERIA

4.4.1 Quantitative Metrics

ID	Metric	Target Value	Acceptance Criteria	(M)andatory / (G)ood to Have / (O)ptional	1st Release / 2nd Release
DAEM-QUAN-01	Risk alert generation time	< 600 ms	< 2 sec	M	1st Release
DAEM-QUAN-02	Response time of risk alerts on mitigation action	< 2mins	<5 mins (depending on the complexity of the device output to be checked)	G	1st Release
DAEM-QUAN-03	Time of data sharing transaction (via the Blockchain)	< 600 ms (per transaction)	< 3 mins (considering the querying of hashed attestation data stored on the ledger)	M	2nd Release
DAEM-QUAN-04	# of cyber-attacks handled against assets	> 85% of sw-based attacks on the deployed sensors	> 70% of sw-based attacks	M	1st Release
DAEM-QUAN-05	Time from risk alert reception after a failed attestation	< 2 mins (including also the identification of the exact attack path)	< 6 mins	M	1st Release
DAEM-QUAN-06	Support different roles with different attributes	<10	< 8 (depicting the current number of the various stakeholders requesting access to the data)	G	2nd Release
DAEM-QUAN-07	Performance evaluation of the privacy-preserving platform authentication (Enhanced DAA JOIN phase)	800 ms	2000 ms	G	2nd Release
DAEM-QUAN-08	Prevention of impersonation attacks, hence mimicking the authenticated and enrolled edge devices by other platforms (via TC usage)	100%	100%	M	1st Release

DAEM-QUAN-09	Successful attempts at breaching confidentiality/gaining unauthorised access to recorded data	0%	0%	M	1st Release
DAEM-QUAN-10	Improved perception of individual users' trust to third parties handling their data	80%	65%	G	2nd Release
DAEM-QUAN-11	# of unauthorised accesses tackled	100%	100%	M	1st Release

TABLE 11: *SMART CITIES REFERENCE SCENARIO – QUANTITATIVE METRICS OF SUCCESS*

4.4.2 Qualitative Metrics

ID	Metric	Target Value	(M)andatory / (G)ood to Have / (O)ptional	1st Release / 2nd Release
DAEM-QUAL-01	Improve anonymity of the operational data collection and aggregation interfaces	Supported	M	2nd Release
DAEM-QUAL-02	Trust guarantees on the IDS (Interoperable Data Sources)	Supported	M	2nd Release
DAEM-QUAL-03	ASSURED scalability on smart cities	Supported (70%)	G	2nd Release
DAEM-QUAL-04	ASSURED Usability	Supported (100%)	M	2nd Release
DAEM-QUAL-05	Level of acceptance from city officials	Supported (100%)	G	2nd Release
DAEM-QUAL-06	Prediction of attacks on devices given identified vulnerabilities	Supported	M	1st Release
DAEM-QUAL-07	Risk assessment and mitigation mechanisms	Supported	M	1st Release (only risk assessment)

TABLE 12: *SMART CITIES REFERENCE SCENARIO – QUALITATIVE METRICS OF SUCCESS*

5 SECURE AND SAFE AIRCRAFT UPGRADABILITY AND MAINTENANCE DEMONSTRATOR

5.1 SECURE AND SAFE AIRCRAFT UPGRADABILITY AND MAINTENANCE

Currently, in the context of this use case there is **no possibility to remotely connect to the Secure Server Router (SSR) or to establish a remote connection between the SSR and the Ground Station Server (GSS).** As a matter of fact, any time any update or maintenance is required on any SSR device, an **entrusted and authenticated engineer has to physically go on the airplane and perform what is needed on site.** Furthermore, the whole procedure has to be validated and signed internally, before the engineer can start operating on the device. Bearing in mind the initial approval chain of the procedure, the engineer travel and operational time and the coordination to have the airplane on ground, this will usually take a substantial amount of time, in the order of days or even weeks. Considering this picture, **it is critical that every remote operation involving any device is done in a safe and secure environment, despite the time that it requires to ensure such security, as long as the time will not become a liability for other threats, such as cyber-threats on the channel itself.** It is also imperative that the whole chain of approval and any operations performed on the device are securely registered on a ledger to ensure traceability of all that has been done on each device.

The table below (Table 13) summarizes the main focus of each of the defined scenarios for this specific use case regarding the security, privacy and trustworthiness requirements as well as the set of ASSURED components to be demonstrated and evaluated.

User Story	Security Properties	Functionalities
UTRC.US.1	Secure and authenticated remote update transfer and install on the SSR. Attestation is required before and after the update to ensure that it is in the correct state ➢ Operational Assurance ➢ Data Integrity ➢ Confidentiality of net. Com.	✓ Risk Assessment ✓ Policy Recom. Engine ✓ Control-Flow Attestation ✓ Config. Integrity Verif. ✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet
UTRC.US.2	Secure and authenticated access to the current health status of the SSR ➢ Confidentiality of net. Com. ➢ Continuous authentic. & Authorisation	✓ Blockchain services ✓ TPM-based Wallet
UTRC.US.3	Secure and authenticated access to the attestation chain performed on the SSR ➢ Operational Assurance ➢ Data Integrity ➢ Confidentiality of net. Com. ➢ Continuous authentic. & Authorisation	✓ Control-Flow Attestation ✓ Config. Integrity Verif. ✓ Swarm Attestation ✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet

UTRC.US.4	Secure and authenticated transfer of data between SSR and GSS, based on the correct instantiation of the security keys ➢ Data Integrity ➢ Confidentiality of net. Com.	✓ Blockchain services ✓ TPM-based Wallet
UTRC.US.5	Secure and up to date risk assessment contracts enforced on all the devices registered to the relevant Blockchain ➢ Operational Assurance ➢ Data Integrity ➢ Confidentiality of net. Com. ➢ Continuous authentic. & Authorisation	✓ Risk Assessment ✓ Policy Recom. Engine ✓ Control-Flow Attestation ✓ Config. Integrity Verif. ✓ Swarm Attestation ✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet

TABLE 13: *SMART AEROSPACE REFERENCE SCENARIO OVERVIEW*

5.1.1 Planning

The Secure and Safe Aircraft Upgradability and Maintenance testbed has identified five user stories in D1.1 [2] and an elicitation for the first release has been completed according to the feasibility of the 1st version on the pilot site and the technical advancements of ASSURED. **User stories UTRC.US.2, and UTRC.US.5 will be tested in the second release while user stories UTRC.US.1, UTRC.US.3, and UTRC.US.4 in the first release**.

For the first round of experimentation, the following user stories will be validated.(**Table 14**):

ID	User Stories	Validations
UTRC.US.1	As a System Administrator I want **to securely log in physically or remotely to the device**, in order to perform authenticated system updates.	The update must be transferred securely on the Blockchain, checking its integrity before it is used. The SSR must be attested before and after the update installation to ensure that it is in a correct state and that the update did not impact its functionalities, respectively. The Configuration Integrity Verification attestation results are accessible for verification.
UTRC.US.3	As a system administrator I want **to ensure that all the device configuration and execution log traces are monitored** efficiently and are securely transmitted to the backend infrastructure in order to correctly perform the attestation process.	The system administrator must be registered to the Blockchain CA and properly authenticated before he can request to have access to the attestation results of his interest. The attestation can be verified locally and compared only if the system administrator has the right privileges.

UTRC.US.4	As an Internal Operator, I want **to ensure that my device transmitted data are protected against baseline communication attacks** in order to securely reach the backend infrastructure for further processing.	All devices must have generated proper keys, attestation and ABE, to ensure that all the data are secure. The devices must be properly registered to the appropriate certificate authority.

TABLE 14: *SMART AEROSPACE REFERENCE SCENARIO FIRST RELEASE DEMONSTRATOR SUMMARY*

In the following section focus is given to the user stories of the 1st release where sequence diagrams, workflows and detailed descriptions are included. The rest are mentioned in summary.

5.1.2 Description and User Stories

The user stories presented in the Secure and Safe Aircraft Upgradability and Maintenance use case are focused on the secure remote maintenance and data transmission between devices in the aerospace environment. The main components of these scenarios are the Secure Server Router (SSR), which is going to be located on the cockpit of the airplane, and the Ground Station Server (GSS), located on the ground in the control tower. The latter will be operated by authenticated personnel, such as ground operators and system administrators. Each operator has its own expertise and tasks, which requires the system to identify their privileges and verify them before any critical action can be executed on the system. All the user stories require secure communication channels, sound authentication and attestation mechanisms, and traceability of all the actions performed by any entity of the infrastructure. All these functionalities are provided by various components within the ASSURED framework. For example, the Blockchain would help with the traceability aspect, while the Attribute-Based Access Control (ABAC) and Attribute-Based Encryption (ABE) will help in verifying the privileges of the operators and securing the data transmitted between the entities of the scenarios, respectively. Furthermore, the same ASSURED components will help our user stories to be protected against various malicious attacks, like corrupt or malicious updates, unauthorized access, and corrupted communication channels. More details on the threats can be found in D1.3.

5.2 DETAILED SCENARIOS

5.2.1 UTRC.US.1

As a System Administrator I want to securely log in physically or remotely to the device, in order to perform authenticated system updates.

User Story Confirmations:

- ✓ *System administrator correctly authenticated by the framework to initiate the update process*
- ✓ *Valid attestations on the devices using CFA and/or CIV are performed.*
- ✓ *SSR updated and attested when requested.*

ASSURED Functionalities:

- ✓ Risk Assessment, Policy Recommendation Engine, Control-Flow Attestation, Configuration Integrity Verification, Runtime Tracing, Blockchain services, TPM-based Wallet

User Story Implementation:

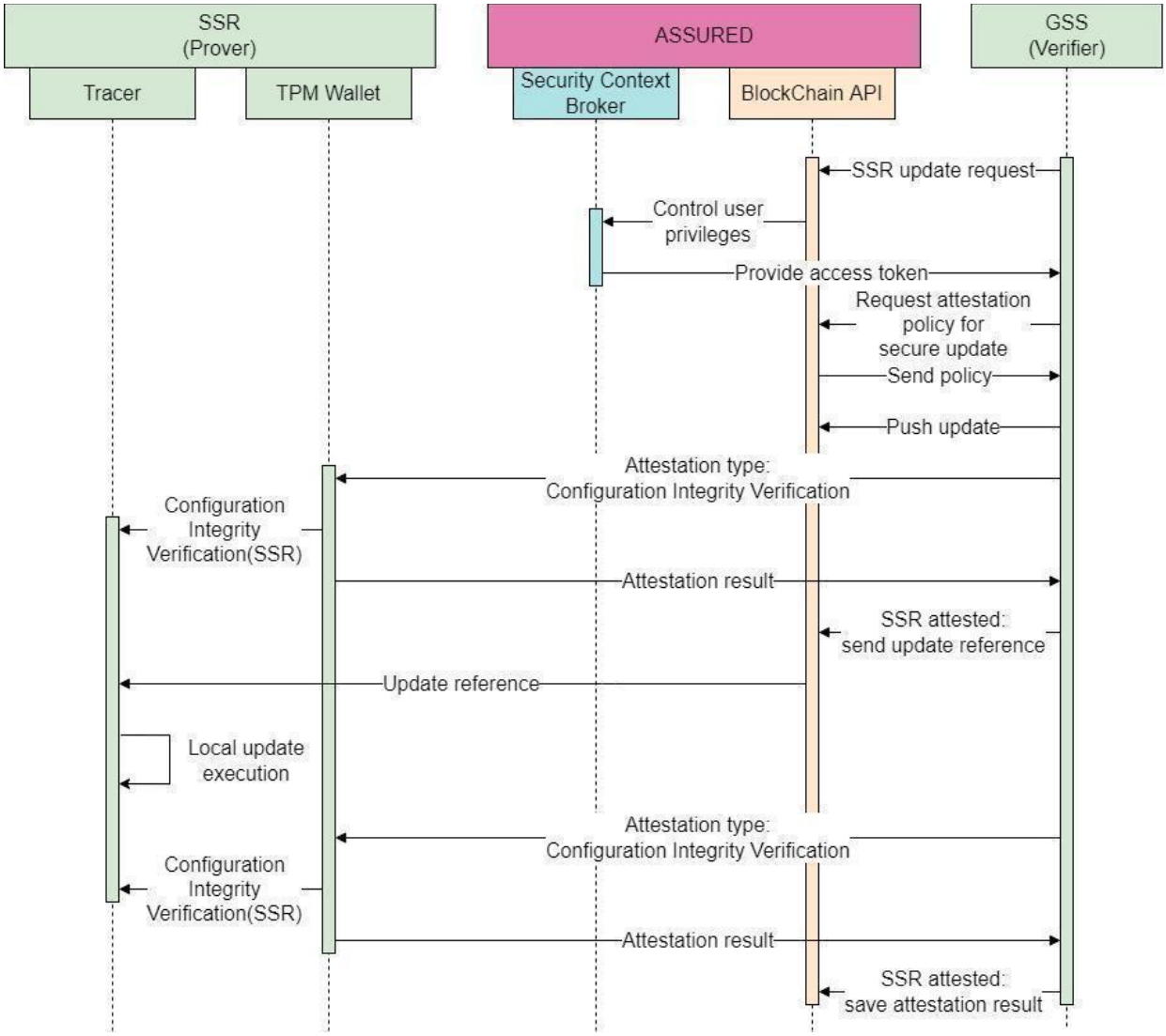

FIGURE 17: *UTRC.US.1 SEQUENCE DIAGRAM*

Workflow:

1. When the airplane is on the ground, once a new update is available, e.g., a message exchange library used by the SSR has to be upgraded, the system administrator logs on the GSS and starts the procedure to execute a remote update on the SSR.
2. First, the system administrator, e.g., airline engineer, needs to be authenticated by ASSURED, which controls his privileges through the ABAC component and returns his access token.
3. The token allows the system administrator to request the policy associated with the remote update. The policy received by the GSS contains a list of all the actions required to perform a sound update on the SSR, which in this case include:
    a. Securely uploading the update on the Blockchain.

   b. A Configuration Integrity Verification (CIV) attestation of the SSR before installing the update to ensure that the SSR is in a valid state.
   c. A CIV attestation of the SSR to verify that the update did not disrupt any functionalities of the SSR.
4. After a valid attestation, the SSR can download the update and install it using the reference received by the Blockchain.
5. As per policy, the system administrator will request a new CIV of the SSR to ensure that the update has gone through correctly.
6. The attestation result is then saved on the Blockchain to ensure traceability.

5.2.2 UTRC.US.2

As an Internal Operator (airline operator), I want to securely have access to the health state information of a remote component, in order to be able to predict any maintenance and management control actions.

User Story Confirmations:

- ✓ *Operator correctly registered to the Blockchain CA so that to have the necessary credentials to perform authenticated and authorised actions.*
- ✓ *Operator correctly authenticated by the framework to initiate the health information acquisition of the devices*
- ✓ *Operator receives and verifies the attestation history of the SSR*

ASSURED Functionalities:

- ✓ Blockchain services, TPM-based Wallet

Workflow:

We assume that the operator has already performed an initial registration to the Blockchain CA.

1. The operator requests the current status of the SSR.
2. The system verifies the privileges and authenticates the operator through the Security Context Broker (SCB) by sending an access token to the operator.
3. The operator is allowed to request the attestation history of the SSR to the Blockchain.
4. For each attestation result received and using the corresponding pointer part of each result, the operator:
   a. Requests the specific raw data to the SCB, which communicates with the Data Storage Engine (DSE) to share the requested data to the operator.
   b. The operator is then able to decrypt all the data received with his ABE key and checks that status of the SSR. Specifically, it is the TPM Wallet component of the operator that certifies each attestation result by verifying their hashes and signatures.

The whole user story holds even if the attestation history of the SSR is currently empty, which could be the case if the operator requests to verify the attestation history of the SSR, but no attestation requests have been performed yet. If this is the case, the operator will receive an empty attestation history and the procedure will end.

User Story Implementation:

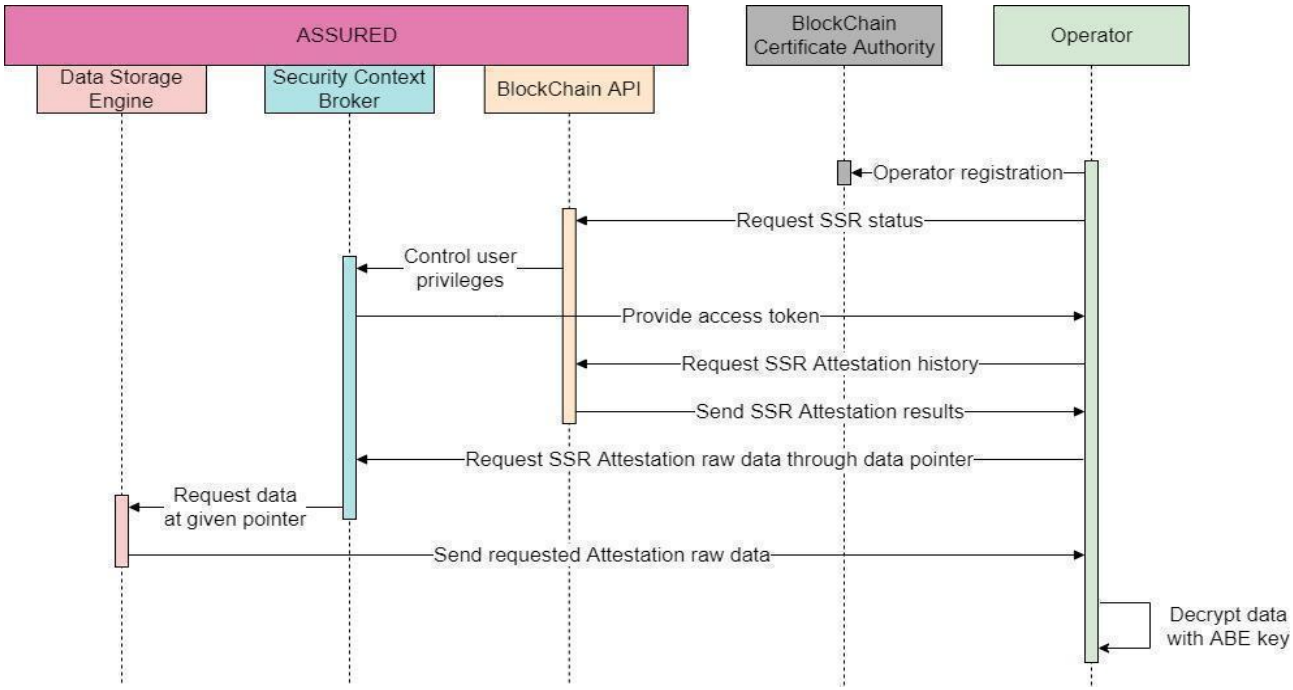

FIGURE 18: *UTRC.US.2 SEQUENCE DIAGRAM*

5.2.3 UTRC.US.3

As a system administrator I want to ensure that all the device configuration and execution log traces are monitored efficiently and are securely transmitted to the backend infrastructure in order to correctly perform the attestation process.

User Story Confirmations:

- ✓ *Operator correctly registered to the Blockchain CA so that to have the necessary credentials to perform authenticated and authorised actions.*
- ✓ *Operator correctly authenticated by the framework to initiate the health information of the devices and receive the attestation chain of the SSR.*
- ✓ *Operator correctly authenticated by the framework to request the raw data associated to the relevant attestation reports and verifies locally the attestation reports of the SSR.*

ASSURED Functionalities:

- ✓ Control-Flow Attestation, Configuration Integrity Verification, Swarm Attestation, Runtime Tracing, Blockchain services, TPM-based Wallet

User Story Implementation:

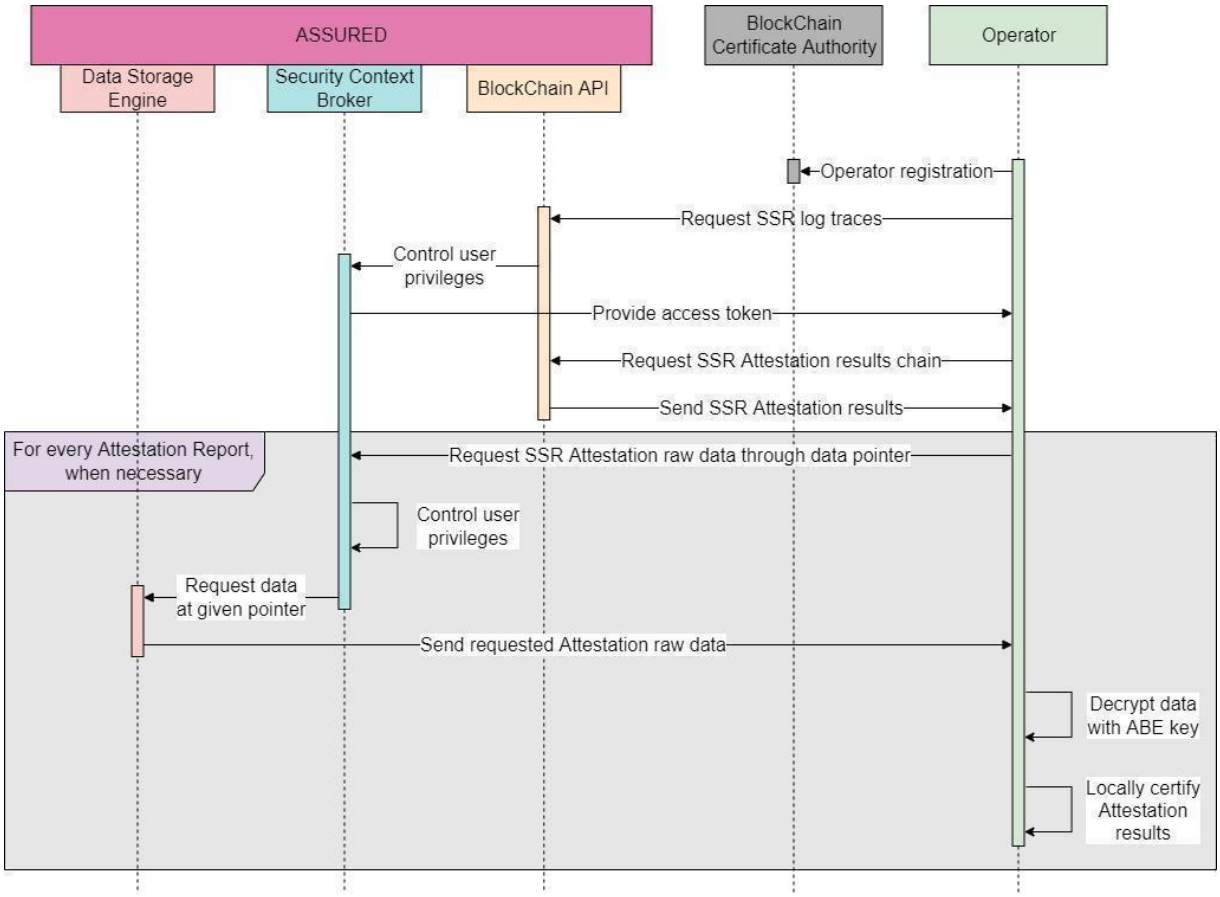

FIGURE 19: *UTRC.US.3 SEQUENCE DIAGRAM*

Workflow:

While not necessary, as described in user story 2, in this user story we assume that at least one full attestation request has been requested to the SSR. This assumption would make it so we can expect to have at least one signed attestation report and the related encrypted attestation raw data stored on the Blockchain and on the DSE respectively. We also assume that the operator has already performed an initial registration to the Blockchain CA.

1. The operator requests the SSR attestation results chain.
2. The system checks the privileges of the operator and replies with an access token if the operator has the right privileges.
3. For each attestation result, the operator can perform a local certification to verify their integrity, which will require extra privileges:
    a. The operator requests the raw data at the pointer received with the attestation result to the SCB.
    b. The system checks if the operator has the correct privileges to request this information.
    c. If properly authenticated, the SCB will retrieve the data from the DSE and send it to the operator.
    d. The operator can then locally decrypt the data with his ABE key and verify the attestation result.

Furthermore, the operator has the possibility to register to the Blockchain for the events related to the SSR and get notified when a new attestation has been performed, allowing the operator to verify the attestation results as soon as they are uploaded.

5.2.4 UTRC.US.4

As an Internal Operator, I want to ensure that my device transmitted data are protected against baseline communication attacks in order to securely reach the backend infrastructure for further processing.

User Story Confirmations:

- ✓ *SSR receives a validity token for its registration and enrolment. Based on the token SSR receives ABE policies and configuration digest. The TPM-based block chain wallet is used for the establishment of secure communication among the parties to guarantee against communication attacks. Valid attestation results and Encrypted data traces are stored on the Blockchain and on the Data Storage Engine.*

ASSURED Functionalities:

- ✓ Blockchain services, TPM-based Wallet

Workflow:

To ensure the security of transmitted data between the devices, e.g., the data collected during the flight that has to be transferred to the GSS once the airplane lands, we need to ensure that the SSR has secure keys to be used when needed. The SSR needs an ABE key and an attestation key that will be used when encrypting its traces and when signing the attestation results, respectively.

1. First, the SSR needs to securely register to the Privacy CA to certify its TPM Wallet and enrol to the Blockchain CA.
2. Once registered, the SSR is allowed to request the policy to generate its ABE and attestation keys.
3. The SSR will then follow each policy and create its keys accordingly.
4. The ABE and attestation keys will respectively be used when encrypting its raw data traces and when signing the attestation results. These keys will ensure that all the data transmitted to other components is safe and secure.

User Story Implementation:

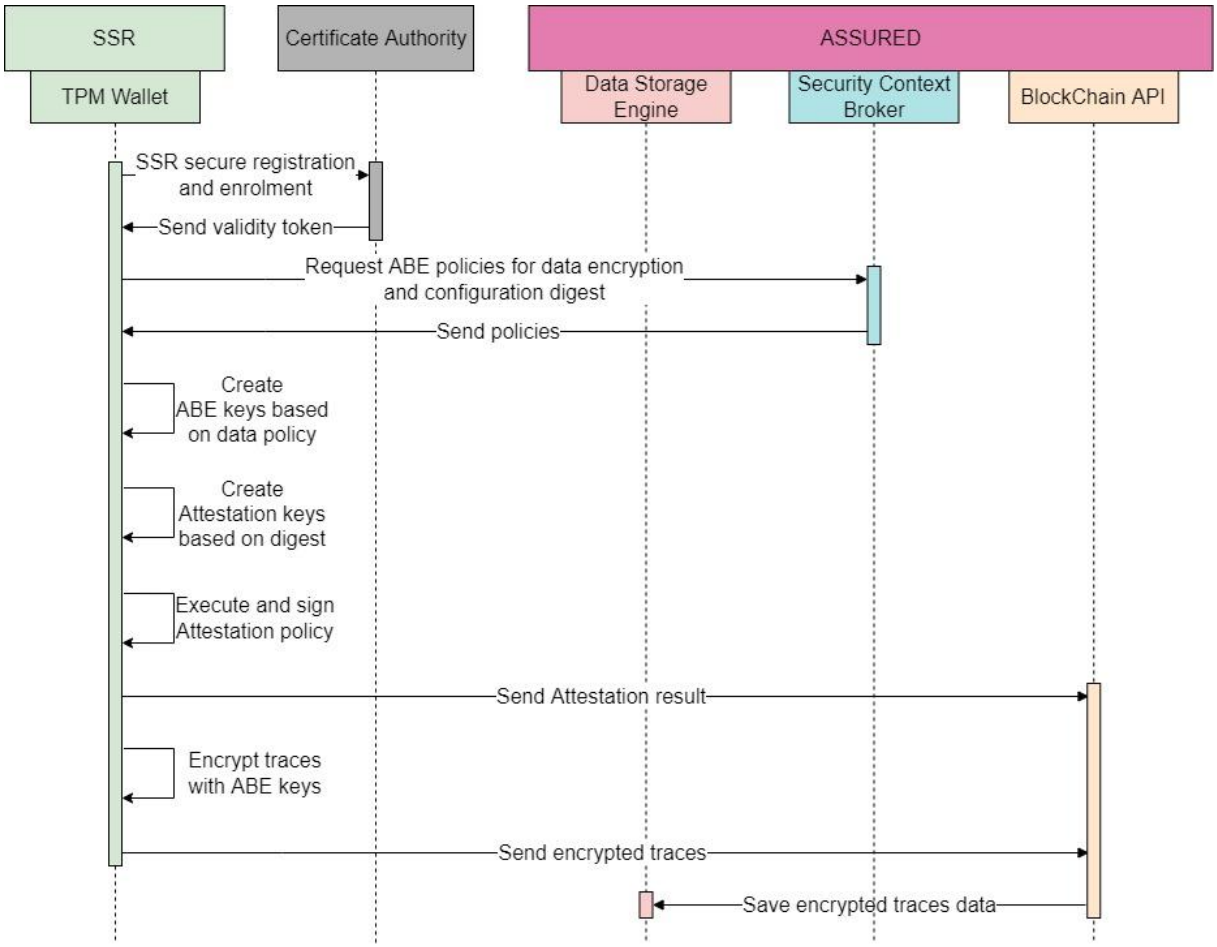

FIGURE 20: *UTRC.US.4 SEQUENCE DIAGRAM*

5.2.5 UTRC.US.5

As an OEM (Ground Station Operator), I want to correctly map my security solutions based on up-to-date security risk assessment in order to always employ the optimised security policies based on the latest events monitored and attacks identified.

User Story Confirmations:

- ✓ *The risk assessment will offer a mapping on the assets and the risk levels. Optimised policies will be deployed using the policy recommendation engine and the Blockchain infrastructure.*
- ✓ *Devices will get notified of a new policy and attestation results are stored on the Blockchain after the conduction of the attestation.*
- ✓ *The authenticated operator verifies the attestation results though the Blockchain offerings.*

ASSURED Functionalities:

- ✓ Risk Assessment, Policy Recommendation Engine, Control-Flow Attestation, Configuration Integrity Verification, Swarm Attestation, Runtime Tracing, Blockchain services, TPM-based Wallet

User Story Implementation:

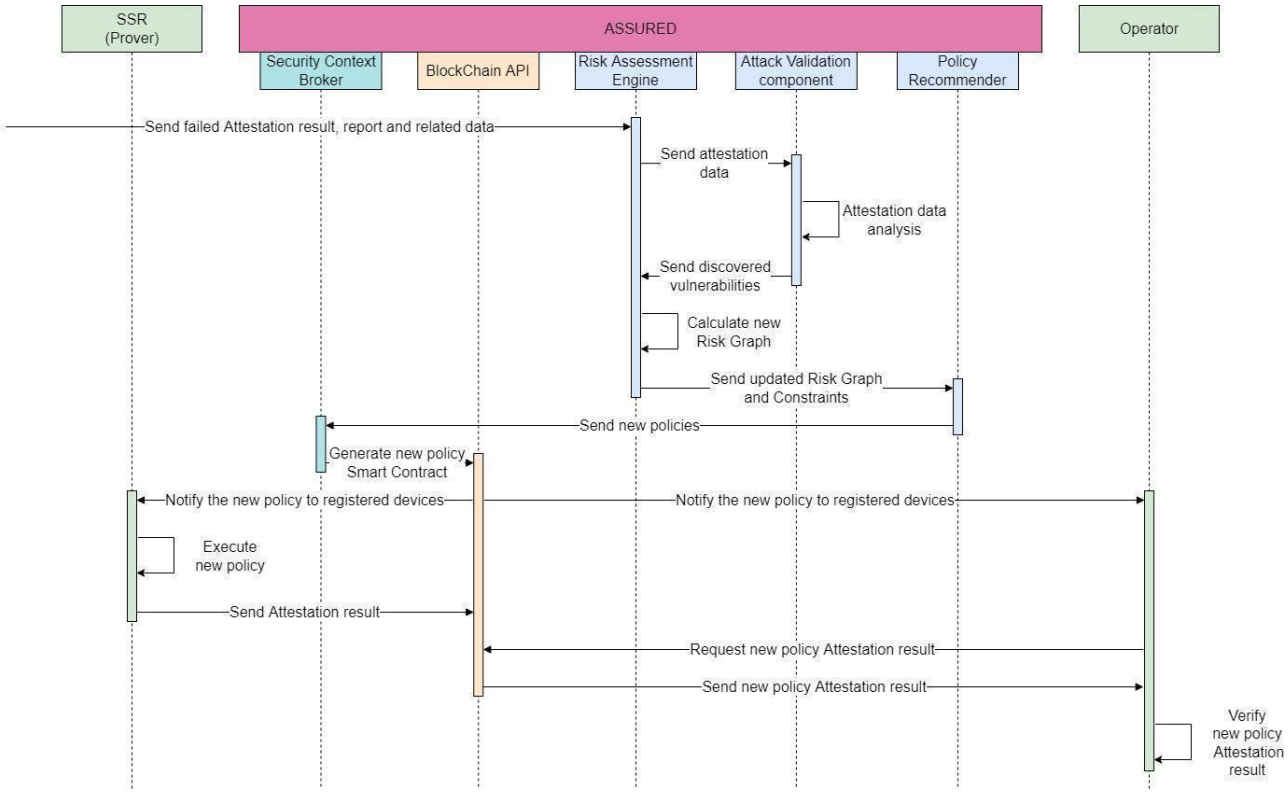

FIGURE 21: *UTRC.US.5 SEQUENCE DIAGRAM*

Workflow:

We assume that at any point in time any device in the framework can fail its attestation check. When this happens, all the data related to such attestation, such as the attestation result, the attestation report, and the related data, will be sent to the Risk Assessment Engine.

1. The Risk Assessment Engine (RAE) receives information of a failed attestation.
2. This information will be shared with the Attack Validation (AV) component which will look for new vulnerabilities.
3. The AV will send back the new vulnerabilities to the RAE.
4. The RAE will generate a new Risk Graph and share it with the Policy Recommender (PR).
5. The PR will analyse the Risk Graph, generate a set of new policies and send it to the SCB.
6. The SCB will translate the new policies into Smart Contracts to enable their enforcement on the Blockchain.
7. The Blockchain will then notify all the devices that a new policy has been created.
8. If needed, the relevant devices will execute the new policy, generate the related attestation report, and the results will be saved on the Blockchain, ready to be accessed by the operator to be verified.

5.3 CONDITIONS

The demonstration will be centred on the SSR, which will be represented by a Xilinx ZCU102 Ultrascale+ platform which will remotely communicate with a couple of Raspberry Pi 3, acting as the GSS and the maintenance server where the update is located. All the devices will run on Linux-based operating systems, which will allow the use of the ASSURED components, such as the tracer and the TPM wallet, crucial elements for the needs of our use case. The binaries that are going to be used for the demonstration will be implemented in C/C++, allowing us to operate at hardware level on the Xilinx platform while also allowing the attestation mechanisms to analyse their behaviours.

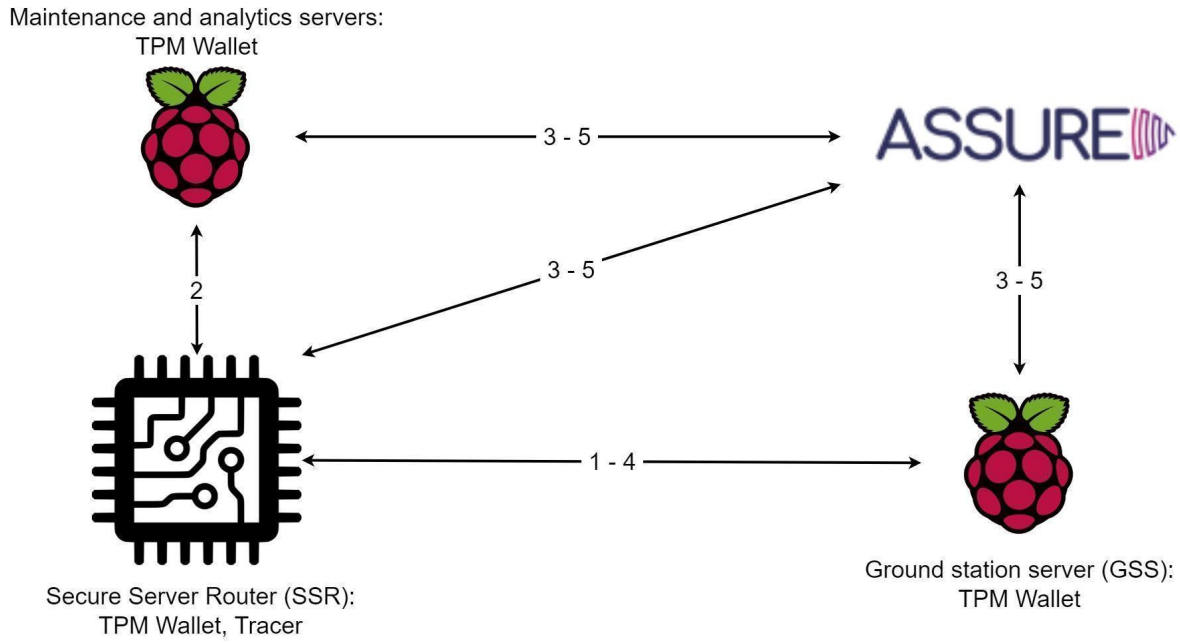

FIGURE 22: *SMART AEROSPACE ENVISIONED DEMONSTRATION SETUP*

5.4 KPIS AND ACCEPTANCE CRITERIA

5.4.1 Quantitative Metrics

ID	Metric	Target Value	Acceptance Criteria	(M)andatory / (G)ood to Have / (O)ptional	1st Release / 2nd Release
UTRC-QUAN-01	Secure Communication between device and operator should not take long periods of time (Establishment of secure and authenticated channel)	<2 minutes	<5 minutes	M	1st and 2nd Release
UTRC-QUAN-02	The core operational tasks of the device should not be impacted by attestation mechanisms	<10%	<10%	M	1st and 2nd Release

UTRC-QUAN-03	The SSR support functionalities have not been impacted by the attestation mechanisms	<15% exhibit operational delays	<20%	G	1st and 2nd Release
UTRC-QUAN-04	Computational resources usage due to attestation mechanisms do not exhibit significant increase	<10%	<15%	O	2nd Release
UTRC-QUAN-05	Amount of SSRs whose integrity can be monitored through ASSURED	100%	100%	M	1st and 2nd Release
UTRC-QUAN-06	Amount of integrity attacks detected on SSRs	80% (with integrity models)	60% (with standard IMA)	M	1st and 2nd Release
UTRC-QUAN-07	Secure software and/or firmware update process	< 1 min	< 2min	M	1st and 2nd Release
UTRC-QUAN-08	Integrity violation alerts delivery latency	< 300 ms	< 500 ms	M	1st and 2nd Release

TABLE 15: REFERENCE SCENARIO 3 – QUANTITATIVE METRICS OF SUCCESS

5.4.2 Qualitative Metrics

ID	Metric	Target Value	(M)andatory / (G)ood to Have / (O)ptional	1st Release / 2nd Release
UTRC-QUAL-01	Secure communication channels when transferring data/updates between devices	SUPPORTED	M	1st and 2nd Release
UTRC-QUAL-02	Complete attestation results (report, result and associated raw data) are secured by signature-based encryption mechanisms.	SUPPORTED	M	1st and 2nd Release
UTRC-QUAL-03	Devices' health state is secure from non-authenticated devices/users.	SUPPORTED	M	1st and 2nd Release

TABLE 16: *SMART AEROSPACE REFERENCE SCENARIO – QUALITATIVE METRICS OF SUCCESS*

6 DIGITAL SECURITY OF SMART SATELLITES DEMONSTRATOR

6.1 DIGITAL SECURITY OF SMART SATELLITES

The digital security of smart satellites use case testbed consists of CubeSats operating and cooperating to execute specific mission(s) and Ground Station, which monitors, maintains, and controls their operation. Given the communication between the CubeSats and the Ground Station, there is a need for ASSURED to **confirm the integrity of all modules cooperating to execute mission critical functions, enhance confidentiality and integrity and provide resilience of the software components (OS and Software modules) against specific attacks**. Since the device operates for critical use cases (e.g., communication) the above issues can be considered as critical. To that extent consideration should be taken related to the **exchanged data confidentiality and integrity (especially for commands)**.

The table below (Table 17) summarizes the main focus of each of the defined scenarios for this specific use case regarding the security, privacy and trustworthiness requirements as well as the set of ASSURED components to be demonstrated and evaluated.

User Story	Security Properties	Functionalities
SPH.US.1	Secure exchange of data as part of the CubeSats and Ground Station communication. Properties of Interest: ➢ Operational Assurance ➢ Data Integrity ➢ Confidentiality of net. Com.	✓ Risk Assessment ✓ Policy Recom. Engine ✓ Control-Flow Attestation ✓ Config. Integrity Verif. ✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet
SPH.US.2	Secure Execution of critical mission Properties of Interest: Confidentiality and Integrity ➢ Operational Assurance ➢ Data Integrity ➢ Confidentiality of net. Com.	✓ Risk Assessment ✓ Policy Recom. Engine ✓ Config. Integrity Verif. ✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet
SPH.US.3	Secure and efficient sharing with external members. Properties of Interest: ➢ Data Integrity ➢ Confidentiality of net. Com. ➢ Continuous authentic. & Authorisation	✓ Direct Anon. Attestation ✓ Blockchain services ✓ TPM-based Wallet
SPH.US.4	Validate the health state of CubeSats swarm. Properties of Interest: ➢ Operational Assurance ➢ Data Integrity ➢ Confidentiality of net. Com.	✓ Risk Assessment ✓ Policy Recom. Engine ✓ Control-Flow Attestation ✓ Config. Integrity Verif. ✓ Swarm Attestation ✓ Runtime Tracing ✓ Blockchain services ✓ TPM-based Wallet

TABLE 17: *SMART SATELLITES REFERENCE SCENARIO OVERVIEW*

6.1.1 Planning

Within the Digital Security of Smart Satellites demonstrator, four user stories have been identified in D1.1 [2] and an elicitation for the first release has been completed according to the feasibility of the first version on the pilot site and the technical advancements of ASSURED. **User stories SPH.US.1, and SPH.US.2 will be tested in the 1st release while user stories SPH.US.3 and SPH.US.4 will follow in the 2nd release.**

For the first rounds of experiments, the following user stories will be validated (Table 18):

ID	User Story	Validations
SPH.US.1	As an Internal Operator (CubeSat Operator), I want **to ensure that the transmitted data are protected against attacks** targeting the devices involved (trying to compromise the key distribution), in order to ensure data confidentiality and integrity.	Key exchange library and related components are monitored and attested every time a new key needs to be established. By monitoring the Key Management related components of the GS and the CubeSat using ASSURED frameworks tracing capabilities, undefined behaviours of the key establishment execution can be verified.
SPH.US.2	As an Internal Operator (CubeSat Operator), I want **to execute critical mission functions in a secure way**, in order to improve the health state information of the entire data value chain.	Software Version Tracking and Software Update Services involved in the execution of critical mission functions (like the distribution of software updates to CubeSats) are attested before execution. By attesting the CubeSat configuration and these services before performing an update, a malicious update can be avoided.

TABLE 18: *FIRST RELEASE SMART SATELLITES DEMONSTRATOR SUMMARY*

In the following section focus is given to the user stories of the first release where sequence diagrams, workflows and detailed descriptions are included. The rest functionalities are put forth in summary and their details will be included in the second release of this deliverable.

6.1.2 Description and User Stories

In day-to-day operations of a CubeSat the data exchanged are commonly related with the following purposes:

- **Executing mission applications on-demand**. This can include for example the triggering of a mission application which orients an imaging device to the requested coordinates and takes a picture.
- **Automatically sending and receiving health and status information**: Primarily this includes a health and status beacon of the CubeSat.
- **Secure querying of the telemetry database** for specific H/W status information.
- **Secure downloading payload data** files through the file transfer service.

All this data collected is important to be transmitted and shared with external organisations through a secure way verifying their confidentiality and integrity. The main challenges in order to achieve increased levels of security, in terms of confidentiality, integrity, and availability, and a more robust and extensible operation of the use case, require the protection of exchanged data and secure operation and update of critical mission applications.

To that end **Device security attestation mechanisms** will be provided by ASSURED to:

- **Enable the performance of remote security attestation** confirming the integrity of all

- modules cooperating to execute mission critical functions.
- **Enhance the confidentiality and integrity** of the exchanged data.
- **Provide resiliency of the Operating System** and Software Modules of the system, effective against multiple vector attacks.

Given the threats analysed at D1.3 and the respective attacks prioritised (including Key extraction, Code Injection, Runtime Attack & Malicious update) a specific set of ASSURED functionalities has been selected to be demonstrated at this use case. The respective countermeasures provided by ASSURED can be summarised in the list below:

- Runtime attestation for verifying the integrity of key access operations.
- Static attestation and runtime attestation before executing a specific operation.
- Runtime attestation for verifying control flow integrity before executing a specific operation.
- Static attestation for verifying binary signature before distributing updated version of the mission application.

More details are provided in the user stories descriptions at the following parts.

6.2 DETAILED SCENARIOS

6.2.1 SPH.US.1

As an Internal Operator (CubeSat Operator), I want to ensure that the transmitted data are protected against attacks targeting the devices involved (trying to compromise the key distribution), in order to ensure data confidentiality and integrity.

User Story Confirmations:

- ✓ *Ground Station and CubeSats can successfully perform Secure Registration and Enrolment.*
- ✓ *CubeSat Operator can receive data from the CubeSats in a secure way.*
- ✓ *Key Exchange Binary can be successfully attested, including the performance of Key Exchange Protocol and Symmetric Key is accepted.*

ASSURED Functionalities/Components:

- ➢ Risk Assessment, Policy Recommendation Engine, Control-Flow Attestation, Configuration Integrity Verification, Runtime Tracing, Blockchain services, TPM-based Wallet

User Story Implementation:

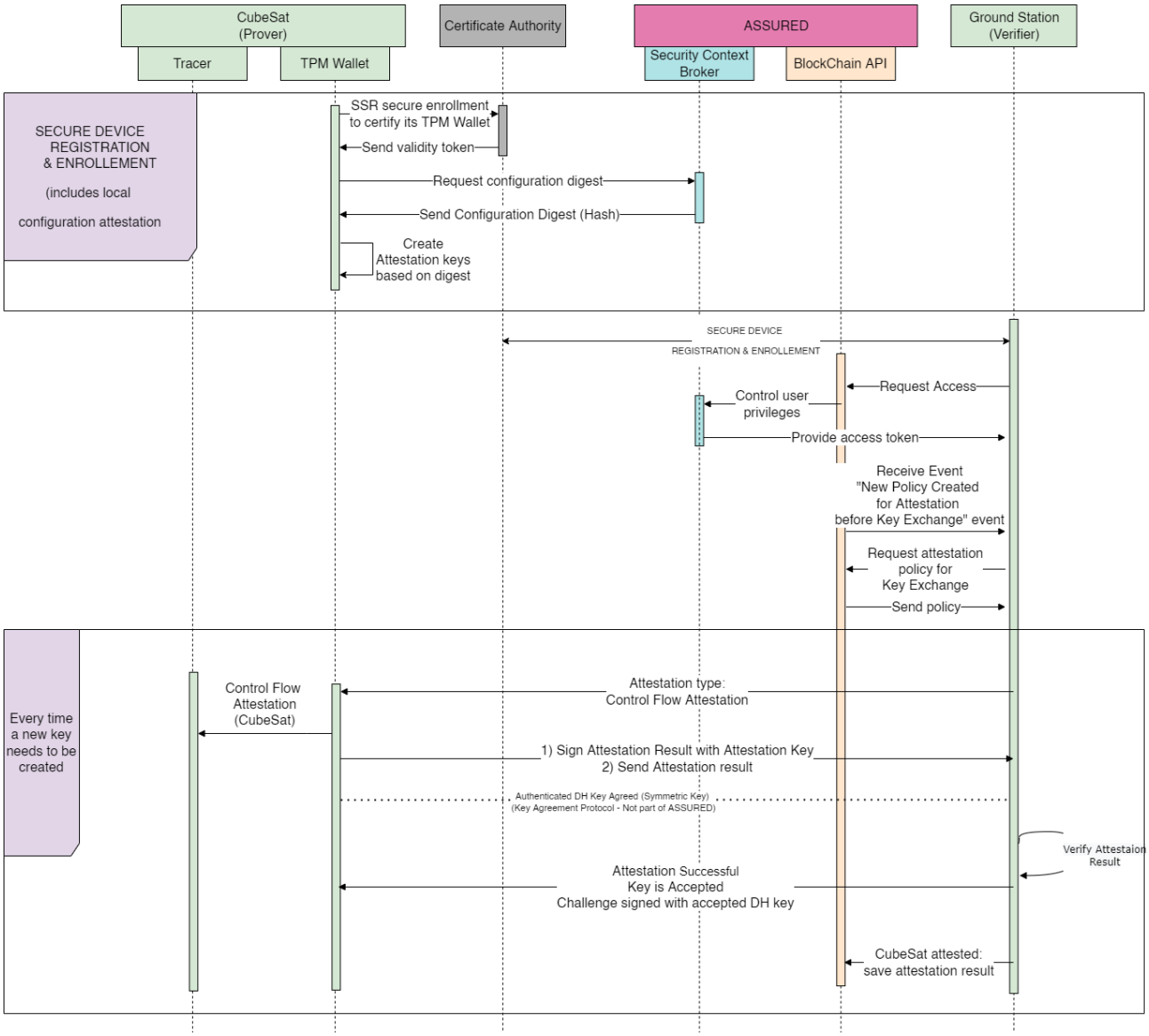

FIGURE 23: *SPH.US.1 SEQUENCE DIAGRAM*

Workflow:

Each time Ground Station and a CubeSat attempt to exchange keys (Run-time attestation for the Key Exchange Protocol) , the Key Exchange Application will be attested to verify that it has not been compromised on both devices. Before the exchange of keys, each device should be able to provide verifiable evidence on its correct configuration state. Devices involved can successfully exchange encrypted data using a security attestation mechanism. As also depicted in the schema below the remote attestation service will be called from both ends (CubeSat and/or Ground Station) and the necessary authentication and verification mechanisms will be used in order for the key exchange process to be approved.

1) A prerequisite of operation is the “Secure Device Registration & Enrolment”.

CubeSat (CS) certifies its TPM Wallet to the Privacy Certification Authority and gets a validity token. Having acquired this token, CS goes to the Blockchain Certification Authority and Performs a Secure enrolment and gets Credentials.

Then a request is made to the Security Context Broker and receives the Configuration Digest (a hash). Using this hash, attestation keys are created for the reports to be signed. The TPM Wallet of CS had made a protected key. Attestation Keys can be used only if configuration is right (verified by Local Configuration Integrity Verification)

2) The TPM Wallet of Ground Station (GS) had made a protected key (Attestation Key) using the exact same process of Secure Registration and Enrolment.
3) Both CS and GS request access. As soon as GS gets access to a private channel it can be notified for all new policies deployed as all registered devices receive these events indicating that a new policy is available.
4) After receiving such an event, GS requests policies available for it from the BC API.
5) According to the policy received, every time a new symmetric key need to be agreed, a Control Flow Attestation is performed (CFA) along with local Integrity verification for both GS and CS so the Attestation Key is used to sign the CFA Results.
6) GS verifies attestation result. Through verification we assure that CFA was done right AND that CS is in the right state (due to local integrity verification attestation). As long as attestation is successful, the key is accepted, and a challenge signed with the accepted DH key is sent verifying that it can be used.
7) GS sends attestation results to the Blockchain API to be stored at the ledger.

Please keep in mind that the above-mentioned flow diagram depicts how Ground Station verifies that DH runs correctly at CubeSat. As an authenticated DH is used from both sides (GS & CS), the exact same workflow should be executed from the CS side. CubeSat (acting as verifier) and GS (Acting as a prover) to Create Keys.

This flow is repeated every time and this policy is implemented (steps 5 – 7) every time a newly generated key needs to be exchanged.

6.2.2 SPH.US.2

As an Internal Operator (CubeSat Operator), I want to execute critical mission functions in a secure way, in order to improve the health state information of the entire data value chain.

User Story Confirmations:

- ✓ *The CubeSat Operator verifies that CubeSat is in the right state, and it is successfully updated.*
- ✓ *Valid attestations (Integrity Verification and Control Flow Attestation) are performed for the CubeSat to be updated and for the services involved.*
- ✓ *Software Update for mission application performed and attested.*

ASSURED Functionalities:

- ✓ Risk Assessment, Policy Recommendation. Engine, Configuration Integrity Verification, Runtime Tracing, Blockchain services, TPM-based Wallet

User Story Implementation:

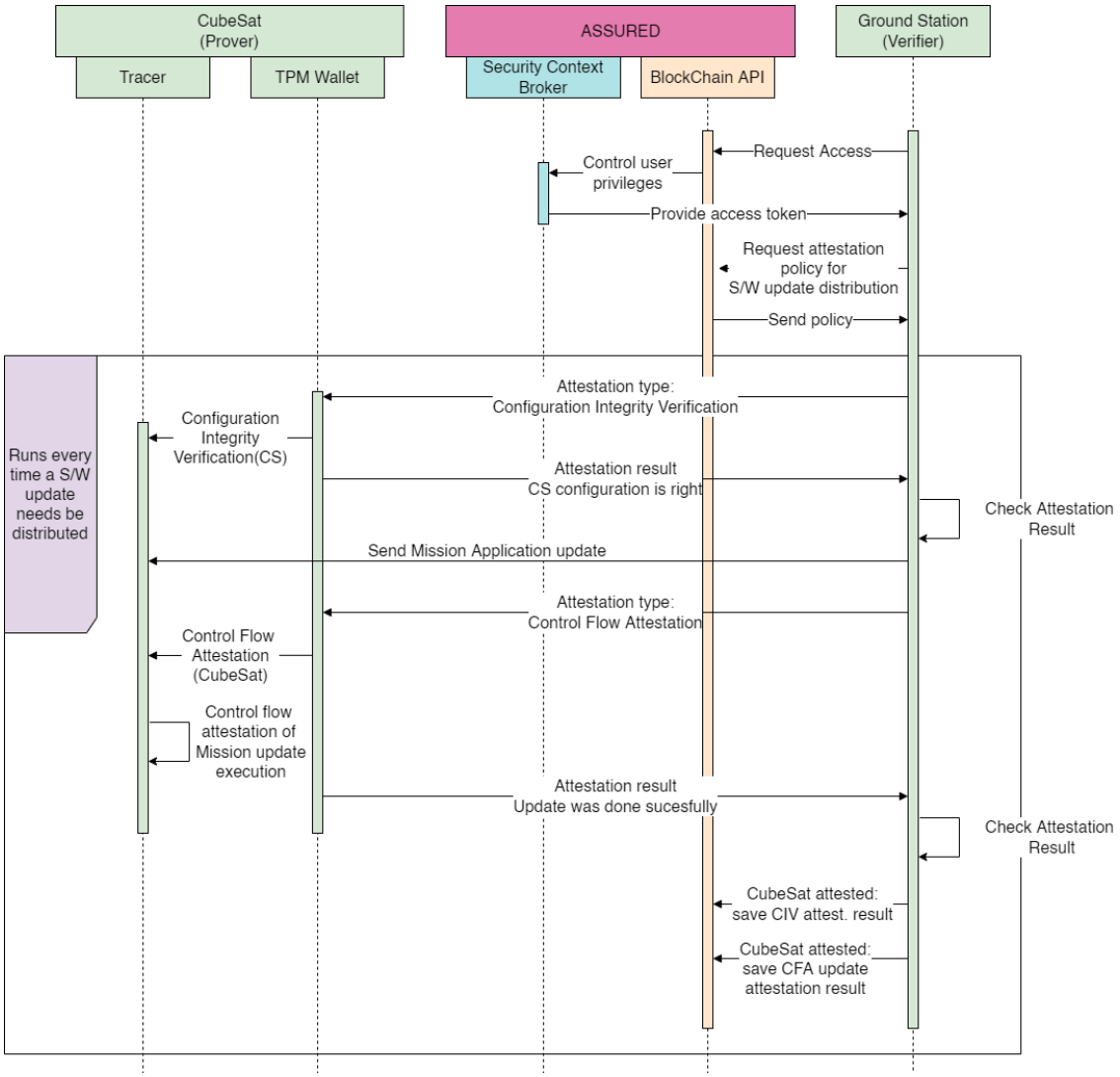

FIGURE 24: *SPH.US.2 SEQUENCE DIAGRAM*

Workflow:

As soon as an updated version of a mission application or an updated version of a specific service is released, it should be distributed to one or more CubeSats. To be protected against specific attacks (e.g., like malicious updates installed) it is important to verify the secure operation of the Software Update distribution to the CubeSat from the Ground Station.

1) As already described at SPH.US.1 a prerequisite of operation is the “Secure Device Registration & Enrolment”. The process is exactly the same and that way we don’t describe it again.

2) As soon as a critical mission is to be executed, the Ground Station (GS) requests to receive the respective policy to be implemented. In this case, the policy for distribution of updated software is requested. For the implementation of this policy 2 attestations have to be made.

3) First a remote Configuration Integrity Verification is used to verify the configuration of CubeSat that will receive the software update.

4) As soon as the attestation result is available, indicating the proper status of the CubeSat, it is sent to Ground Station to be checked.

5) After the successful attestation the software update is sent and a Control Flow Attestation (CFA) of the update execution process is performed (second attestation needed).

Local Update Execution is performed updating mission applications. As soon as the local update has been executed and the CFA results are checked, the updates are sent to the Blockchain API. Please note that the updates for both attestations (Integrity Verification and CFA of the update) are sent after the successful completion of the whole process given the critical nature of the task to be executed.

6.2.3 SPH.US.3

As an Internal Operator (CubeSat Operator), I want to share data collected and received from the CubeSats with External Member(s) (including users of external organisations), in order to update them about the status of the CubeSats and Mission in a secure, accountable and in efficient manner.

User Story Confirmations:

- ✓ *Successful store of Attestation Result(s) on the Blockchain*
- ✓ *Successful store of Raw Traces at Storage*
- ✓ *Successful update of respective Attestation Results at Blockchain with pointer reference.*
- ✓ *Successful query and access to results from external members.*

ASSURED Functionalities:

- ✓ Direct Anonymous Attestation, Blockchain services, TPM-based Wallet

Workflow:

The aim of this user story is to validate the proper operation of sharing mechanisms with external stakeholders according to their access rights. More specifically for the context of this demonstrator we envision two types of External Members (EMs). The first one with advanced access privileges refers to External Entities that require direct access to private ledger. This is the case for Regulating Authorities monitoring the status of CubeSats. The second category, including other external entities with more limited access, like possible service providers or service integrators. Below we can see the flow for both types of Ems and how they can get access to monitoring data about the status of the CubeSats in a secure, accountable and efficient way.

As mentioned also in previous User Stories (US) descriptions, a prerequisite of operation is the “Secure Device Registration & Enrolment” for all devices involved (including the external stakeholders. Detailed steps of this process can be found at SPH.US.1 workflow (chapter 6.2.1).

User Story Implementation:

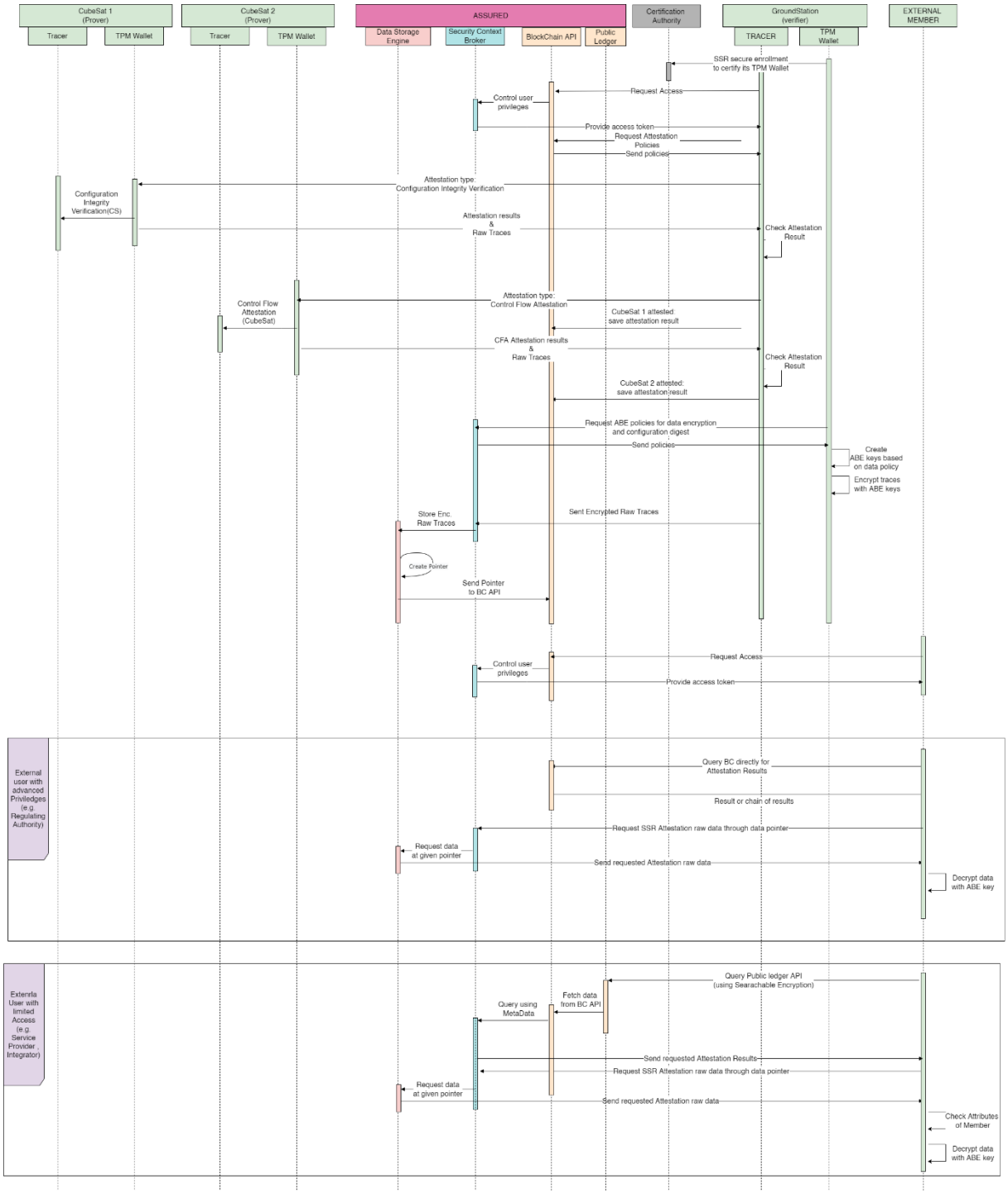

FIGURE 25: *SPH.US.3 SEQUENCE DIAGRAM*

In this specific scenario, Ground Station (GS) is the Verifier and there is the need to:

- Check configuration integrity verification for CubeSat No. 1
- Check the performance of secure software update distributions for CubeSat No. 2.

In line with the description of SPH.US.1 and SPH.US2, GS Read the Attestation Policies through the Blockchain (BC) API and initiates the Attestation Tasks.

GS gets the Results as a verifier from CS1 and CS2. After the results have been checked by GS, they are signed with its Attestation Key.

1) Then the Attestation Results are sent to BC to be stored at Ledger
2) Apart from the attestation results, the related RAW TRACES collected by CSs are also stored. GS, after encrypting them using its ABE keys, sends the encrypted Raw Traces to the Security Context Broker (SCB).
3) SCB makes sure to store them in an offline store (Assured Storage Engine) and create a Pointer and send it to the BC API. This reference (pointer) is added to the record of the respective Attestation Report.
4) After the above steps are completed, this data can be accessible from external members. For this specific use case there are two different types of external members according to their access privileges.
5) To proceed with querying and accessing data, an External Member makes a request to the Security Context Broker. It is a prerequisite for external members to have the required certificate and attributes.
6) SCB Checks the Certificate provided and provides the respective access (Access token). If no certification is available, the entity will be redirected to certification authority to be registered
7) EM, using this access token and depending on the type of privileges provided, can continue with two possible workflows.
8) Scenario 1(1st box): The EMs with advanced access privileges include External Entities with direct access to private ledger. This is the case for Regulating Authorities monitoring the status of CubeSats. After EMs being authenticated, they can Query Directly the BC API to get Attestation Results. Please note that query can return either a result or a chain of results. For example, one query can return a set of results including the results from a CS reporting once per day for 5 days.
9) Scenario 2(2nd box): Other external entities with more limited access can include service providers. As they don't have direct access to BC (Private Ledger API), the Searchable Encryption Mechanisms of ASSURED are used to perform queries over encrypted metadata to the Public Ledger API.
10) If any attestation results (Attestation Knowledge) are found, a query is done with Metadata retrieved from the Public Ledger to SCB.
11) SCB fetches specific attestation results from the BC API and sends them back to the EM requesting the data.
12) After the EM has received the Attestation Results, he can also retrieve the respective Encrypted Raw traces based on which these attestation results are created. This can be done via making a request to the security context broker.
13) EM makes a query to SCB to bring RAW TRACES (based on the POINTER included at ATTESTATION REPORT received)
14) SCB returns encrypted DATA to EM
15) As the traces are encrypted, a check is performed via the TPM Wallet of EM validating Internal Attributes of Member. If Internal attributes of Members are validated, then the decryption keys are reproduced for RAW traces to be decrypted.

6.2.4 SPH.US.4

As an Internal Operator (CubeSat Operator), I want to securely and efficiently communicate with CubeSats and collect data, in order to check the health state of the entire chain of communicating satellites.

User Story Confirmations:

- ✓ *The CubeSat operator can successfully check the health state of the entire chain of communicating satellites.*
- ✓ *Successful store of Swarm Attestation Results on the Blockchain*
- ✓ *Authorised External Members can get access to the Attestation Results.*

ASSURED Functionalities:

- ✓ Risk Assessment, Policy Recommendation Engine, Control-Flow Attestation, Configuration Integrity Verification, Swarm Attestation, Runtime Tracing, Blockchain services, TPM-based Wallet

Workflow:

As soon as the CubeSat Operator wants to check the status and to verify the proper operation of the whole constellation of CubeSats operating, there is a need to perform a swarm attestation and check the whole set in a fast and efficient way.

1) To do so, Ground Station sends an attestation request (as the verifier) and initiates the attestation process via sending an attestation challenge to all CubeSats participating at a constellation (CS1, CS2 ... CSn). This is done in parallel.
2) As soon as the process of Configuration integrity verification is completed, the results are signed using the TPM Wallet of each CS with ASSURED Group/Swarm Attestation Signatures Keys. Then the results are sent to GS.
3) GS collects all the results and stores them in the ledger. Please note that although the GS can verify that all signatures are included, GS does not know which signatures belong to which device.
4) If there is a need, this can be further investigated by using the Tracer mechanism (part of Security Context Broker).
5) External members (like the ones mentioned at SPH.US.3) can also get access and check Swarm Attestation results.
6) If they evaluate the results and there is the need to investigate further (e.g. in case one attestation task has failed and the received results are less than the devices attested), they can make a request to Tracer (part of Security Context Broker Mechanism) to know which device failed. Of course, the appropriate attributes are required in order to access this kind of data.

Please note that as also mentioned in the previous user stories, all devices involved should have been properly registered with their TPM Wallet. This process is performed at the CubeSat initiation phase. Through that phase, TPM Wallet will also create Group Based Signature Keys which are used only for Swarm Attestation Requests.

User Story Implementation:

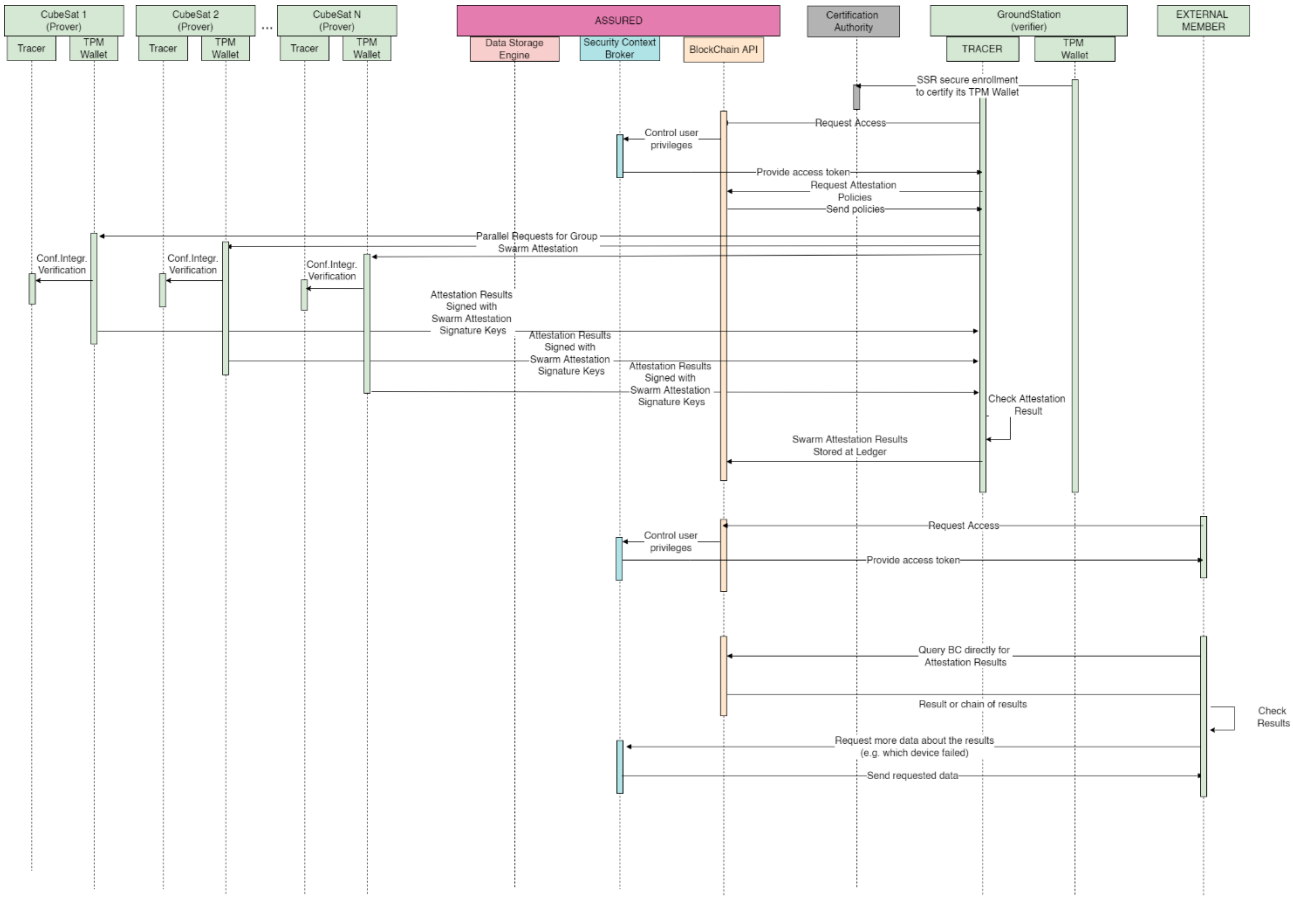

FIGURE 26: *SPH.US.4 SEQUENCE DIAGRAM*

6.3 CONDITIONS

For the Smart Satellites demonstrator, SPH currently provides a testbed including a Ground Station (GS) and 3 CubeSats (CS) - onboard computers (OBC) running KUBOS OS (v1.21.0).

GS monitors each CS, collects data and distributes updated versions of the Mission App or Services running at each one of the CS. For demonstration purposes all devices are connected to a N/W switch through ethernet cables.

For the user stories to be demonstrated at ASSURED, a TPM Wallet will be installed along with the Tracers. A client for key exchange management is installed at the GS and the CSs as well. The N/W switch is connected to a router in order to be accessible to the Internet and ASSURED Services (e.g., Blockchain's API).

The connectivity with ASSURED services will enable the demonstration of secure enrolment and registration of all devices involved.

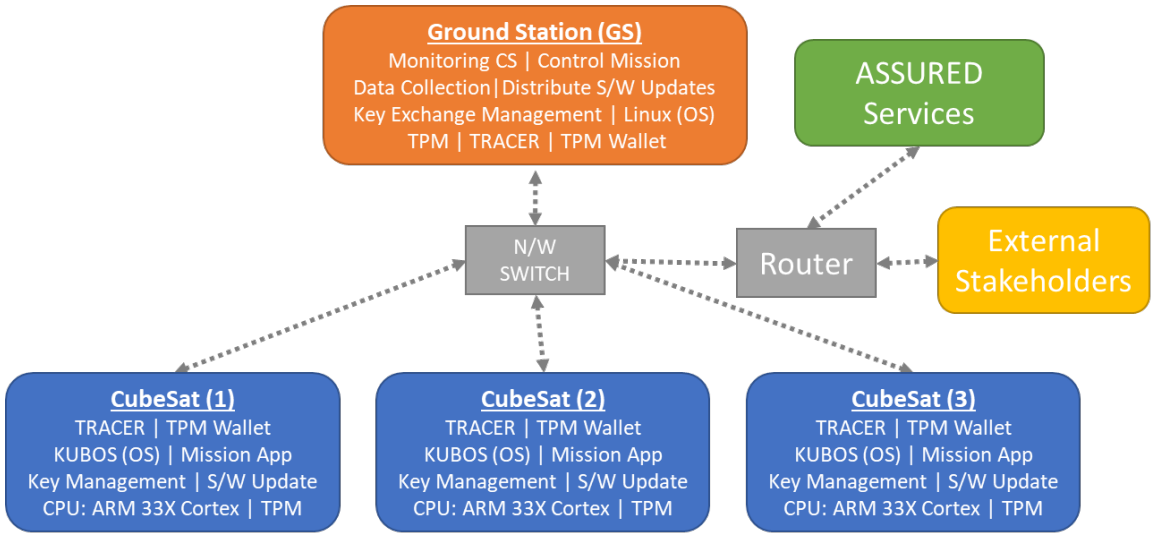

FIGURE 27: *SMART SATELLITE ENVISIONED DEMONSTRATOR SETUP*

6.4 KPIS AND ACCEPTANCE CRITERIA

6.4.1 Quantitative Metrics

ID	Metric	Target Value	Acceptance Criteria	(M)andatory / (G)ood to Have / (O)ptional	1st Release / 2nd Release
SPH-QUAN-01	Processing time for ASSURED attestation (Excluding any related networking and/or transmission actions of the produced attestation reports)	CIV < 800 ms CFA (with ML) < 1 min	1 min	M	1st and 2nd Release
SPH-QUAN-02	Blockchain on-chain data management operations.	< 1 sec	1 sec	G	2nd Release
SPH-QUAN-03	Coverage of processes, running inside CubeSat, whose configuration and execution integrity can be verified.	100% (with integrity models)	90% (standard IMA)	M	2nd Release
SPH-QUAN-04	Interval time between start of the attestation process and secure data transfer.	< 1 min	1 min	M	1st and 2nd Release
SPH-QUAN-05	Attestation both at atomic and swarm level.	At Least Atomic	Atomic Attestation Covered	O	2nd Release

SPH-QUAN-06	Time to securely pass the control of one CubeSat from one ground station to another	3 sec	4 sec	G	2nd Release
SPH-QUAN-07	Setup secure and communication channel with periodic key update	Keys to be exchanged with every mission critical data exchange (< 5 sec)	< 4 sec	G	1st and 2nd Release

TABLE 19: *SMART SATELLITE REFERENCE SCENARIO – QUANTITATIVE METRICS OF SUCCESS*

6.4.2 Qualitative Metrics

ID	Metric	Target Value	(M)andatory / (G)ood to Have / (O)ptional	1st Release / 2nd Release
SPH-QUAL-01	Secure transmission and attested device before transmitting data.	Supported	M	1st and 2nd Release
SPH-QUAL-02	Prevention of OS attacks leading to privilege escalation.	100%	G	2nd Release
SPH-QUAL-03	Runtime Risk assessment.	100%	M	1st and 2nd Release
SPH-QUAL-04	Integrity protection of device configuration and behavioural data.	Supported	M	1st and 2nd Release
SPH-QUAL-05	Reduction of data acquisition	Supported	G	1st and 2nd Release
SPH-QUAL-06	Secure transmission and attested device before transmitting data.	Supported	M	1st and 2nd Release

TABLE 20: *SMART SATELLITE REFERENCE SCENARIO – QUALITATIVE METRICS OF SUCCESS*

7 ASSURED INTEGRATION, TESTING AND EVALUATION PLAN

The main objective of this section is the (non-exhaustive, given that software development runs in parallel) description of the applied unit tests to be performed for the ASSURED integrated framework in the context of the four reference scenarios. **Unit tests are the tool to test the functional modules of a software.** In the context of the ASSURED integrated framework, unit tests will guarantee the quality of the particular layers developed in the corresponding work packages. More precisely, ASSURED applies the unit test in three layers:

- **Layer 1**: Usage of:
  - ✓ **Attestation Enablers** - unique test cases per reference scenario listing the type of attestation enabler to be tested; e.g., *Control-Flow Attestation, Configuration Integrity Verification, Direct Anonymous Attestation, jury-based Attestation, Swarm Attestation* [7];
  - ✓ **Data Sharing Operations** - common test case for all the reference scenarios leveraging the **ASSURED TPM-based Wallet [8] for device authentication, in the Blockchain infrastructure, and secure on-chain interactions** towards querying the recorded attestation data (testing ASSURED Searchable Encryption scheme [9]). Essentially a series of tests for evaluating all the lightweight crypto schemes supported by the designed TPM-based Wallet;
  - ✓ **TPM Functionalities and Operations** – unique test cases per reference scenario for evaluating the different sets of TPM-supported functionalities (e.g., Device Registration and Enrolment [10], Tracer Authentication [11], Attribute-based Encryption [10], etc.) in specific scenarios;
- **Layer 2**: Evaluation of **dynamic management of security (attestation) policies**, calculated by the Policy Recommendation Engine and enforced through the developed smart contracts. This set of tests also includes the **auditable and immutable recording of attestation results in the distributed ledger** towards the creation of an "attestation data hub" capable of supporting all the identified data sharing behaviours and threat intelligence information sharing capturing the required security and privacy requirements (common test cases);
- **Layer 3**: Operations of the supporting environments that are related to the **risk and vulnerability assessment** as well as the **attack simulation and validation** based on the real-time system traces collected from the edge devices (common test cases).

Initially, we describe the common test cases that apply to all the reference scenarios and should also be aligned with the overall goal of the ASSURED to test the efficiency of the newly developed ASSURED variants. We then provide the specific test cases per reference scenario based also on the aforementioned user stories.

7.1 ALL REFERENCE SCENARIOS UNIT TESTING

Test Case ASSURED01	
Reference Code	ASSURED01
Reference Scenario	All
Description	This unit test aims at verifying the correctness of all TPM operations **(ASSURED TPM Wallet)** associated with all key management functionalities, signing operation and encryption/decryption functionalities. For the former, the correctness of the Attestation Key generation is of primary interest whereas for the latter focus will be given on correctness of the TPM key generation and certificate generation associated with that TPM key. Furthermore, the management of Verifiable Credentials (VCs) and Verifiable Proofs (VPs) to be used for the authentication of a device when

	trying to access a specific resource (Blockchain service and/or data bundle) will also be evaluated.

Test Case ASSURED02	
Reference Code	ASSURED02
Reference Scenario	All
Description	This unit test aims at verifying the correctness of the **OLISTIC Risk Assessment (RA) Framework** for calculating the risk graph of all the component's entities envisioned to the reference scenarios and creating a set of adequate security and attestation policies to be deployed to the devices **(Policy Recommendation Engine)**. This applies to both during design- and run-time. ● During design-time, the RA calculates the risks and the risk graphs of all the entities and components and creates a set of initial high-level security policies. ● During run-time, the RA creates, and updates new run times polices based on events of interest (e.g., any abnormal behaviour that might be an indication of an attack) that were detected during a devices' operation.

Test Case ASSURED03	
Reference Code	ASSURED03
Reference Scenario	All
Description	This unit test aims at verifying the correctness of the generation of the Smart Contracts from a new policy to be enforced on the Blockchain. These generations will have to hold during run-time until a new attack, threat or indication of risk has been identified in which case an update in already deployed contracts might occur holding the new security policies. The correct parsing of the necessary attributes and translation of MSPL-based policies into smart contracts is the focal point of this test.

Test Case ASSURED04	
Reference Code	ASSURED04
Reference Scenario	All
Description	This unit test aims at verifying that all the devices registered to the Blockchain will receive notification of new policies.

Test Case ASSURED05	
Reference Code	ASSURED05
Reference Scenario	All
Description	This unit test aims at verifying that the **Policy Recommendation Engine** is able to process the MSPL-based output (from the Risk Assessment) and the optimisation process defines an optimal set of ordered attestation and operational tasks to be executed per device (or sets of devices when swarm attestation needs to be employed).

Test Case ASSURED06	
Reference Code	ASSURED06
Reference Scenario	All
Description	This unit test aims at verifying the correctness of the **Attack Validation** component, focusing on the evaluation of the fuzzing process been able to detect the state of misconfiguration of the technical components emulated

	by the validation component. The fuzzer should be able to imitate all possible states of the device.

Test Case ASSURED07	
Reference Code	ASSURED07
Reference Scenario	All
Description	This unit test aims at verifying the **configuration and operational correctness of the deployed devices based on the calculated set of security and attestation policies.** The policies reflect the Control-Flow Graphs (CFGs for Control-Flow Attestation) and Binary Configuration Traces (for Configuration Integrity Verification) against which we compare and attest the monitored device traces as outputted by the ASSURED SW-based Tracer and based on the sequence of TPM commands of a specific session ID.

Test Case ASSURED08	
Reference Code	ASSURED08
Reference Scenario	All
Description	This unit test aims at verifying the correct operation of the **Direct Anonymous Attestation** process and enables remote authentication of a trusted computer whilst preserving device privacy in terms of anonymity and unlinkability. The testing includes the correct key and base management operations with the TPM, the management of the basename, and the correctness of the Attestation key generation is of primary interest whereas.

Test Case ASSURED09	
Reference Code	ASSURED09
Reference Scenario	All
Description	This unit test aims at verifying the correct operation of the Swarm attestation process based on the use of smart contracts. Validation of the fact that a device can act both as a verifier and prover in the context of swarm attestation.

Test Case ASSURED10	
Reference Code	ASSURED10
Reference Scenario	All
Description	This unit test aims at verifying the correct operation the ASSURED Run-time Tracer. The tracer aims to provide slightly different functionalities per attestation scheme and depending on the system layer aimed to be monitored. Thus, the test aims to validate that the necessary spectrum of system properties can be captured by the tracer.

Test Case ASSURED11	
Reference Code	ASSURED11
Reference Scenario	All
Description	This unit test aims at verifying the correct operation of the data storage used for the off-chain management of data. This storage engine must be able to serve data queries for the acquisition of attestation and business data. **Searchable Encryption and Attribute-based Encryption** are core cryptographic schemes that have dependent functionality with the data storage engine. Thus, the unit test must ensure the unified operation of the aforementioned offerings. searchable encryption shall be verified for

	providing the correct pointer pointing to the off-chain data and ABE shall be testing for the correct generation of the required decryption keys.

Test Case ASSURED12	
Reference Code	ASSURED12
Reference Scenario	All
Description	This unit test aims at verifying the correct operation the Blockchain services including the automatic notification of all registered devices when a new security policy is been deployed. Furthermore, particular focus will be given on the scalability of the designed infrastructure considering the amount of concurrent data access and data recording requests that must be handled.

Test Case ASSURED14	
Reference Code	ASSURED13
Reference Scenario	All
Description	This unit test aims at verifying the correctness of the set of the TPM commands that are core to all the reference scenarios including key Creation, Key Binding, Signing, Encryption, Decryption, Platform Configuration, Sealing and Unsealing.

7.2 SAFE HUMAN ROBOT INTERACTION (HRI) IN AUTOMATED ASSEMBLY LINES

Test Case BIBA.TC.01	
Reference Code	BIBA.TC.01
Reference Scenario	BIBA.US.1
Components	IoT Gateway, Data Aggregator, Program Logic Controller
Description	This unit test aims at verifying the correct execution of RMT, PLMC and CPA services being deployed through Assured's holistic Risk Assessment (RA) Framework. For this, System administrator provides reference scenarios to create a set of adequate security and attestation policies based on manually introduced sw-based vulnerabilities in the these computational tasks.

Test Case BIBA.TC.02	
Reference Code	BIBA.TC.02
Reference Scenario	BIBA.US.2
Components	IoT Gateway, Data Aggregator
Description	This unit test extends the functionality of ASSURED07 and aims at verifying the correct configuration and execution state of all deployed devices, based on the optimal set of attestation policies calculated.

Test Case BIBA.TC.03	
Reference Code	BIBA.TC.03
Reference Scenario	BIBA.US.2
Components	Data Aggregator
Description	This unit test aims at verifying the correctness of the integrity verification of the transaction's history log. The unit test encrypts the history transactional

	logs as well as the monitored system traces leveraging ASSURED ABE scheme.

Test Case BIBA.TC 04	
Reference Code	BIBA.TC.04
Reference Scenario	BIBA.US.3
Components	IoT Gateway, Data Aggregator, Programme Logic Controller
Description	This unit test aims at verifying the operational correctness of the real-time location monitoring system that is connected to all of the PLCs deployed in a manufacturing floor. This requires the generation of the appropriate CFGs (leveraging the ASSURED Tracer) reflecting the normal behaviour of all its executional binaries and their verification against the expected behaviour as was defined by the System Administrator. In the case of a failed attestation result, this test will also validate the correct operation of the Attack Validation component for fleshing out the exact attack path followed that led to the failed attestation. Hardcoded software-based vulnerabilities will be injected in the codebase that can lead to buffer overlfow, ROP attacks, etc. so as to test case both the Tracer and the Attack Validation component

Test Case BIBA.TC 05	
Reference Code	BIBA.TC.05
Reference Scenario	BIBA.US.4
Components	Data Aggregator
Description	This unit test aims at verifying the correct registration of all devices and the subsequent generation of the necessary cryptographic material (i.e., Attestation Key) that needs to be binded/sealed under the correct key protection usage policy. Essentially, the TPM should not allow the use of the AK unless the device is at an expected state – based on the state that it was registered during the device enrolment with the Privacy and Blockchain CA. The focus would be at verifying the correctness of the TPM AK and certificate generation. It verifies that AK is created with the given policy and that the generated certificate is associated to that TPM key.

Test Case BIBA.TC 06	
Reference Code	BIBA.TC.06
Reference Scenario	BIBA.US.4
Components	IoT Gateway, Security Context Broker
Description	This unit test aims at verifying the Security Context Broker functionality of the Assured framework by successfully enrolling new trusted devices (such as Data aggregators, IoT Gateways) and establishing a secure communication channel to the smart manufacturing infrastructure.

Test Case BIBA.TC 07	
Reference Code	BIBA.TC.07
Reference Scenario	BIBA.US.5
Components	IoT Gateway, Data Aggregator
Description	This unit test aims at verifying secure communication properties of the ASSURED framework. Upon success, a secured communication channel is established between trusted devices (such as data aggregators) and IoT Gateway (Raspberry Pi) running RMT, PLMC, CPA services for data

	exchanges. The focus would be on testing the creation of ephemeral symmetric keys through the ASSURED TPM-based Wallet.

Test Case BIBA.TC 08	
Reference Code	BIBA.TC.08
Reference Scenario	BIBA.US.6
Components	IoT Gateway, Data Aggregator
Description	This unit test aims to verify the ASSURED framework's zero-touch provisioning capability by successfully enrolling a new trusted device into the smart manufacturing infrastructure without any manual provisioning while protecting the privacy of the devices' configuration against implementation disclosure attacks.

Test Case BIBA.TC 09	
Reference Code	BIBA.TC.09
Reference Scenario	US.7
Components	IoT Gateway
Description	This unit test aims to verify capabilities of the query engine of the ASSURED framework by successfully querying different levels of details in the execution of a device through the Assured framework so the services such as RMT, PLMC, CPA deployed on IoT Gateway can be enhanced.

7.3 SECURE COLLABORATION OF "PLATFORMS-OF-PLATFORMS" FOR ENHANCED PUBLIC SAFETY

Test Case DAEM.TC.01	
Reference Code	DAEM.TC.01
Reference scenarios	DAEM.US.2
Components	Edge device, Stakhodlers/Users of Public Safety Monitoring Tool
Description	This unit test aims at authenticating users and edge devices by the system in order to provide verified access to actions only to those who have the required attributes and system identifiers. This unit test extends ASSURED01 for evaluating the ASSURED TPM-based Wallet's capability to issue the required verifiable proofs disclosing only those attributes needed for getting access to the target resource.

Test Case DAEM.TC.02	
Reference Code	DAEM.TC.02
Reference scenarios	DAEM.US.3
Components	Edge Devices, Gateways
Description	This unit test aims at ensuring the secure data flows and the trustworthiness of the channels so as to avoid data leaks. In case of an attack an alert is issued. The focus here is on the correct execution of the DAA protocol for creating the necessary DAA Key and the subsequence short-term anonymous credentials (pseudonyms) that can be used for anonymously signing all exchanged messages.

Test Case DAEM.TC.03	
Reference Code	DAEM.TC.03
Reference scenarios	DAEM.US.6
Components	Edge Device, System Administrator, Security Context Broker
Description	This unit test aims at querying the health state of an edge-device (e.g., a sensor) by an operator. The operator creates a query and receives the attestation results from the devices TPM.

Test Case DAEM.TC.04	
Reference Code	DAEM.TC.04
Reference scenarios	DAEM.US.6
Components	Edge device, Security Context Broker
Description	This unit test aims at verifying the signature (SIGN phase) of device's bunch of data using the DAA key. This unit receives the data from the device and then checks how the TPM forwards back the signed data, either anonymously or non-anonymously based on the leveraged DAA base-name.

7.4 SECURE & SAFE AIRCRAFT UPGRADABILITY/ MAINTENANCE

Test Case UTRC.TC.01	
Reference Code	UTRC.TC.01
Reference Scenario	ALL
Components	SSR, GSS, Operator, System Administrator, Airline Engineers
Description	Operators and devices properly registered to the relevant CA and authenticated by the system to provide access to specific actions only to those who have the right privileges, e.g., remote update request, remote health state monitor, request and local verification of attestation chain.

Test Case UTRC.TC.02	
Reference Cod	UTRC.TC.02
Reference Scenario	ALL
Components	SSR, GSS, Operator, System Administrator
Description	Ensure that secure and safe channels are properly set up before sharing confidential data between devices, e.g., SSR and GSS.

Test Case UTRC.TC.03	
Reference Cod	UTRC.TC.03
Reference Scenario	UTRC.US.1
Components	SSR
Description	Compare that the functionalities of the SSR have not been impacted by the new update.

Test Case UTRC.TC.04	
Reference Cod	UTRC.TC.04

Test Case UTRC.TC.04	
Reference Scenario	UTRC.US.5
Components	SSR, GSS
Description	Verify the correctness of the generation of the Smart Contracts from a new policy to be enforced by the Blockchain and executed on the relevant devices, e.g., SSR. These generations will have to hold during design-time as well as during run-time.

Test Case UTRC.TC.05	
Reference Cod	UTRC.T.05
Reference Scenario	UTRC.US.5
Components	SSR, GSS
Description	Verify that all the devices registered to the Blockchain will receive notification of new policies, e.g., SSR, GSS.

Test Case UTRC.TC.06	
Reference Cod	UTRC.TC.06
Reference Scenario	ALL
Components	SSR, GSS
Description	Verify the correctness of all TPM functionalities (ASSURED TPM Wallet) associated with all key management functionalities, signing operation and encryption/decryption functionalities. For example, these keys will be needed when generating attestation reports by the SSR and to ensure the validity of the data transferred from the SSR to the GSS.

Test Case UTRC.TC. 07	
Reference Cod	UTRC.T.07
Reference Scenario	UTRC.US.1
Components	SSR
Description	Verify that the SSR is in the correct state before and after performing the remote update requested by the authenticated airline engineering operating through the GSS. The verification will be done through a Configuration Integrity Verification.

Test Case UTRC.TC 08	
Reference Cod	UTRC.T.08
Reference Scenario	UTRC.US.1, UTRC.US.4
Components	SSR, GSS
Description	Verify that the data transferred between the SSR and the GSS have not been tampered with by external malicious actors. For example, verify the integrity of the update and verify the integrity of the flight data.

7.5 DIGITAL SECURITY OF SMART SATELLITES

Test Case SPH.TC.01	
Reference Code	SPH.TC.01
Reference Scenario	SPH.US.1

Components	Ground Station, CubeSat 1, CubeSat2, CubeSat3
Description	This unit test aims at verifying the establishment of symmetric keys among Ground Station and CubeSats. Correctness of Control Flow attestation of ASSURED should be validated and ensure that process can be initiated, performed and completed with the correct Key Exchange Management Operation for the establishment of the symmetric key. Emphasis will be made to processing time for ASSURED Attestation.

Test Case SPH.TC.02	
Reference Code	SPH.TC.02
Reference Scenario	SPH.US.2
Components	Ground Station, CubeSat 1, CubeSat2, CubeSat3
Description	This unit test aims at verifying the proper operation of safety critical procedures (e.g., Distribution of Software Update) Correctness of Attestation Integrity Verification Mechanisms of ASSURED should be validated and ensure that process can be initiated, performed and completed ensuring the operation correctness. Emphasis will be made to processing time for ASSURED Attestation.

Test Case SPH.TC.03	
Reference Code	SPH.TC.03
Reference Scenario	SPH.US.3
Components	Ground Station, CubeSat 1, CubeSat2, CubeSat3
Description	This unit test aims at verifying the proper operation of sharing information with externals. Correctness of Attestation Integrity Verification Mechanisms of ASSURED should be validated and ensure that process can be initiated, performed and completed ensuring the operation correctness for external to search, receive and decrypt data requested. Emphasis will be made to processing time for ASSURED Attestation.

Test Case SPH.TC.04	
Reference Code	SPH.TC.04
Reference Scenario	SPH.US.4
Components	Ground Station, CubeSat 1, CubeSat2, CubeSat3
Description	This unit test aims at verifying the proper operation of Attestation at Swarm Level. Correctness of Attestation Integrity Verification Mechanisms of ASSURED should be validated. The proper receipt from Ground Station and the possibility for results to be stored on the Blockchain.

Test Case SPH.TC.05	
Reference Code	SPH.TC.05
Reference Scenario	SPH.US.3, SPH.US.4
Components	Ground Station, CubeSat 1, CubeSat2, CubeSat3
Description	This unit test aims at verifying the proper operation of Security Context Broker and Data Storage Engine of ASSURED serving requests from external stakeholders. More specifically the proper operation of their mechanisms will be tested in order to Store Raw traces at Storage and Search and Retrieve the required information requested by external.

8 SUMMARY AND CONCLUSIONS

The current deliverable aims to document the activities of Task 6.1 "*Evaluation Framework Definition and Demonstrators Planning*" in the context of WP6. D6.1 describes the testing approach and evaluation plan to be followed, when experimenting with the overall ASSURED integrated framework, in the context of the four envisaged reference scenarios. Therefore, **testing becomes important so as to ensure the quality of the delivery, both at submodule level and as a whole integrated system.** To that end, a specific approach has been described for the technical and business evaluation of the platform.

Towards this direction, D5.1 has already put forth a **technical guideline for the adequate integration of the ASSURED mechanisms and software components**, which follows a "bottom-up-approach". It included an integration and testing plan, so each one of the components should first go through a set of unit tests and satisfy requirements in terms of interfacing and software quality in order to be considered ready for the final integration.

Building upon this technical guideline, D6.1 [6] then proceeds with the documentation of the framework aspects to be tested and evaluated, against a set of **test cases, that have been identified per reference scenario:** in order to verify the proper **functioning and performance of the core components against pre-defined Key Performance Indicators (KPIs)** but also to evaluate the capturing of the **security, privacy and trustworthiness criteria** that have been identified per use case. The four use cases foresee the evaluation of specific functionalities of the ASSURED framework in different application domains, namely *Smart Manufacturing, Smart Aerospace, Smart Cities and Smart Satellites*. More specifically, the **Smart Manufacturing** reference scenario focuses on the **operational assurance of the deployed edge devices and data aggregators in an ecosystem with strict time constraints**, thus, the execution of attestation enablers should not impact the performance of other computational functionalities of the devices; the **Smart Aerospace** reference scenario focuses on the **need for remote SW updates**, of the devices comprising the aircraft, coming from s**ecure and authenticate sources**; the **Smart Cities** reference scenario focuses on the strict **user privacy issues** that need to be met and the requirement for **role-based access control** to groups of stakeholders requesting access to specific operational and attestation data recorded on the Blockchain; and, the **Smart Satellite** reference scenario focuses on **monitoring and the establishment of trust** between satellites and the Ground Station that are part of the same safety-critical mission

All these will be used for the demonstrator's implementation and tests scenarios performance for the evaluation. The specific planning of scenarios to be tested per release contributes to prioritising the technical work and will be used along with the test cases defined as well for the platform assessments execution.

Overall, this document has given an overview of the details experimentation plan to be adopted during the first evaluation round of the ASSURED framework in the context of the defined use cases. The key output has been the definition of the set of test cases for the list of core, integral components plus the technologies to be leveraged towards carrying on with such tests, paying special attention to the evaluation plan.

ABBREVIATIONS

Abbreviation	Description
ABAC	Attribute-based Access Control
ABE	Attribute Based Encryption
AK	Attestation Key
API	Application Programming Interface
BFT	Byzantine Fault Tolerance
BGP	Byzantine Generals Problem
CA	Certification Authority
CP-ABE	Ciphertext Policy Attribute Based Encryption
CRED	AK Credential
DAA	Direct Anonymous Attestation
DApps	Distributed Applications
DLT	Distributed Ledger Technology
DoA	Description of Action
DPos	Delegated Proof of Stake
Dx.x	Deliverable x.xl
ECDSA	Elliptic Curve Digital Signature Algorithm
EK	Endorsement Key
HLF	Hyperledger Fabric
IoT	Internet of Things
KDF	Key Derivation Function
KP-ABE	Key Policy Attribute Based Encryption
KPI	Key Performance Indicator
MSP	Membership Service Provider
PBFT	Practical Byzantine Fault Tolerance
PCR	Platform Configuration Register
PE	Policy Enforcement
PEP	Policy Enforcement Point
PK	Public Key
PoA	Proof of Authority
PoB	Proof of Burn
PoC	Proof of Capacity
PoET	Proof of Elapsed Time
PoS	Proof of Stake
PoW	Proof or Work
RA	Risk Assessment
SCB	Security Context Broker

SE	Searchable Encryption
SGX	Software Guard Extensions
SK	Secret Key
TPM	Trusted Platform Module
US	User Scenario
UT	Unit Test
WPx	Work Package X

REFERENCES

[1] ASSURED Consortium, 2021, “D1.2 Reference Architecture”

[2] ASSURED Consortium, 2021, “D1.1 Use Cases and System Requirements”

[3] ASSURED Consortium, 2020, ASSURED Description of Action

[4] “https://iso25000.com/index.php/en/iso-25000-standards/iso-25010” [Online]

[5] “https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en” [Online]

[6] ASSURED Consortium, 2021, “D5.1 Technical Integration Points, APIs Specification and Testing Plan”

[7] ASSURED Consortium, 2021, “D3.2 ASSURD Layered Attestation and Runtime Verification Enablers Design & Implementation”

[8] ASSURED Consortium, 2022, “D4.5 ASSURED TC-based Functionalities”

[9] ASSURED Consortium, 2022, “D4.3 ASSURED Blockchain-based Control Services and Crypto Functions for Decentralized Data Storage, Sharing and Access Control”

[10] ASSURED Consortium, 2022, “D4.2 ASSURED Secure Distributed Ledger Maintenance & Data Management”

[11] ASSURED Consortium, 2021, “D3.1 ASSURED Attestation Model and Specification”