Cross-platform **O**pen **S**ecurity **S**tack for **Con**nected Device

D1.3 Validation Criteria Initial Version

Document Identification			
Status	Final	**Due Date**	31/07/2023
Version	1.0	**Submission Date**	31/07/2023

Related WP	WP1	**Document Reference**	D1.3
Related Deliverable(s)	D1.1, D1.2	**Dissemination Level (*)**	PU
Lead Participant	3MDEB	**Lead Author**	Przemysław Sulewski, Maciej Pijanowski
Contributors	3MDEB	**Reviewers**	Hristo Koshutanski, ATOS
			David Purón, Ainara Garcia, BIOT)

Keywords:
Use Case Requirements, Validation Criteria, Validation Scenarios

Document Information

List of Contributors	
Name	**Partner**
Maciej Pijanowski	3MDEB
Rafał Kochanowski	3MDEB
Przemysław Sulewski	3MDEB
Maciej Pijanowski	3MDEB

Document History			
Version	**Date**	**Change editors**	**Changes**
0.1	20/04/2023	Rafał Kochanowski (3MDEB)	filling in the primary fields, ToC.
0.2	21/04/2023	Rafał Kochanowski (3MDEB)	ToC construction completion
0.3	31/05/2023	Przemysław Sulewski (3MDEB)	Add first version of sections: Introduction, Methodology. Add first 15 Validation Scenarios
0.4	14/06/2023	Przemysław Sulewski (3MDEB)	Apply first changes arising from the review
0.5	14/06/2023	Przemysław Sulewski (3MDEB)	Add Validation Scenarios correspond with the Use Cases requirements
0.6	03/07/2023	Maciej Pijanowski (3MDEB)	Include scenarios for UCX-Y requirements and drop the other ones. Add Executive Summary chapter. Add Conclusions chapter. Remove the template information on guidelines for each section.
0.7	12/07/2023	Maciej Pijanowski (3MDEB)	Adjusted Requirements definition (and consequently, the Validation Scenarios) according to the latest version. Added Figure 1, References section, overall improvements to the document editing.
0.8	21/07/2023	Maciej Pijanowski (3MDEB)	Apply review comments from ATOS/BIOT/CYSEC
0.9	31/07/2023	Juan Alonso (ATOS)	Final version based on quality control
1.0	31/07/2023	Hristo Koshutanski (ATOS)	Final version submitted

Quality Control		
Role	**Who (Partner short name)**	**Approval Date**
Deliverable leader	Maciej Pijanowski (3MDEB)	21/07/2023
Quality manager	Juan Alonso (ATOS)	31/07/2023
Project Coordinator	Hristo Koshutanski (ATOS)	31/07/2023

Table of Contents

List of Figures

List of Acronyms

Abbreviation / acronym	Description
D1.3	Deliverable number 1 belonging to WP1
DoA	Description of Action
DUT	Device Under Test
EC	European Commission
MFA	Multi-Factor Authentication
UC	Use Case
WP	Work Package

Executive Summary

This document provides an overview and the definition of the CROSSCON stack Validation Criteria, which serve as a comprehensive guide to assess the fulfillment of project requirements. The Validation Criteria have been based on the project's completed phases and the Use Cases and Requirements formulated in Deliverable D1.1 [1] and D1.2 [2]. It is emphasized that these Criteria may evolve as the project progresses, with potential for expansion to tackle new features or identified problems.

Reading this document will help the reader understand the steps that must be taken to ensure that the CROSSCON stack meets the identified Requirements, the necessary equipment for these checks, and the expected outcomes. Furthermore, the reader will comprehend the significant relationship of this document with other project works and how it impacts the upcoming project stages.

In terms of results, this document presents a comprehensive process for creating validation scenarios from project requirements, which ensures that the Validation Criteria are met. It draws on existing methodologies in the literature for generating validation criteria from use cases and requirements, and presents a novel, simplified two-step analysis process specific to the CROSSCON project.

In summary, the deliverable is a significant contribution to the CROSSCON project, offering a practical guide to validation scenarios that are integral to the project's success. It guarantees that the project complies with the set requirements, and provides a roadmap for future project stages, making it an indispensable tool for all involved parties.

1 Introduction

1.1 Purpose of the document

This document presents the definition of the CROSSCON stack Validation Criteria as the result of the first iteration between application/service providers – BIOT, 3MDEB, and CYSEC, and the academic (UNITN, UWU, UMINHO, TUD) and industrial partners (ATOS, BEYOND) of the project.

Validation Criteria were prepared based on the two project phases completed so far, providing input data in the form of Use Cases (Deliverable D1.1 [1]) and Requirements (Deliverable D1.2 [2]). The primary purpose of introducing the Validation Criteria is to make it possible to determine whether the solutions presented in subsequent design stages meet the initial project assumptions.

When analysing the Validation Criteria, it is essential to note that they may evolve as the project develops - depending on the identified problems or new features, the Validation Criteria list might be expanded.

1.2 Ambition of the Validation Criteria

The document's proposed Validation Criteria will investigate if the proposed CROSSCON stack solution meets the declared requirements.

In essence, the produced Validation Criteria should comply with the assessment Requirements specified in D1.2 [2] and guarantee principles of fairness, flexibility, validity, repeatability and reproducibility. The defined criteria will be evaluated in D5.4 and implemented as the stack validation scenarios.

Every defined Validation Criterion roughly describes what steps should be performed to confirm the fulfilment of the dependent Requirement. It also includes a list of equipment needed to complete the check and lists the expected result of the scenario performed.

1.3 Relation to other project work

This document describes the initial version of the Validation Criteria. Deliverable D1.3 provides a detailed definition of the Validation Criteria, which is necessary to determine whether the CROSSCON stack meets the Requirements from D1.2 [2].

The first version of the Validation Criteria provides valuable input for the following work packages and deliverables:

1. **WP2 Design Specification, Safety and Assurance** - CROSSCON stack specification should incorporate the developed Validation Criteria, allowing implementation of the tests based on the Validation Criteria in next project stages.
2. **WP3 Development of CROSSCON stack** - CROSSCON stack development should take into account the developed testing scenarios, allowing to avoid potential bugs at the solution creation stage.
3. **WP5 Integration and Validation** - all delivery of integration, testing and validation results should be based on the prepared validation scenarios. In the case of ***D5.1 - Use Case driven Testbed Environment***, the solutions proposed in this document should be taken into account when designing the testbed. In the case of ***D5.3 - Security Testing and Validation Results of the CROSSCON Stack in Use Cases***, the solutions presented in this document should be considered when creating the test cases and deciding if the CROSSCON stack meets the requirements.

It should be noted that not only D1.3 affects the aforementioned project stages. Also, these stages will impact the subsequent versions of D1.3.

1.4 Structure of the document

This document is structured into five main chapters.

Chapter 1 is the introduction and aims to prepare the reader to understand the scope of the document.

Chapter 2 summarizes approaches identified in the literature to produce validation criteria based on the Use Cases and Requirements. This chapter also presents the path to creating Validation Criteria, which is proposed to adapt for the CROSSCON stack.

Chapter 3 presents the Validation Criteria, prepared based on D1.2 Requirements [2] considering the proposed approach.

The document ends with a conclusion presented in **Chapter 4**.

1.5 Glossary adopted in this document

This glossary provides definitions and explanations of key terms and concepts used in our work on preparing the Validation Criteria. It serves as a reference guide to ensure a common understanding of the terminology used throughout our project. By using this glossary, we aim to promote clarity and consistency in our communication, facilitating effective collaboration and understanding among project stakeholders. Please refer to this glossary to find definitions for terms related to our validation scenarios and other important concepts in our work.

- **Validation Scenario** - A specific test scenario designed to verify the functionality, performance, or compliance of a system, component, or feature.
- **Context** - The relevant background information or conditions that influence the validation scenario, including the system architecture, requirements, and project context.
- **Preconditions** - The necessary conditions that must be met before executing the validation scenario, such as the availability of specific hardware, software, or configuration settings.
- **Actions and Interactions** - The sequence of steps or activities performed during the execution of the validation scenario, involving the system under test, test environment, and any external components or entities.
- **Expected Results -** The anticipated outcomes or behaviours that indicate successful execution of the validation scenario, often expressed as specific conditions, values, or system responses.
- **Alternate Paths** - The alternative sequences or branches that can be followed within the validation scenario if certain conditions or actions deviate from the main path, often involving error handling, exception scenarios, or alternative system behaviour.
- **Validation Environment** - The controlled environment or setup in which the validation scenario is executed, comprising the necessary hardware, software, configurations, and test infrastructure.
- **Device Under Test (DUT)** - The specific system, component, or feature being validated or tested within the validation scenario.
- **Test Results** - The outcome, observations, measurements, or data generated during the execution of the validation scenario, which are recorded and analysed to assess the success or failure of the test.
- **Error Handling** - The set of procedures, mechanisms, or strategies employed within the validation scenario to handle and recover from errors, exceptions, or unexpected conditions encountered during testing.
- **Optimization Measures** - The actions or modifications applied to enhance or optimize the system or its components based on observations or findings from the validation scenario, aiming to improve performance, reliability, or other desired attributes.

- **Optimization** - The process of making improvements or adjustments to the system or its components based on the insights gained from the validation scenario, with the goal of enhancing performance, efficiency, or other relevant metrics.
- **Cross-Platform** - Refers to the capability of a system, software, or technology to operate or function seamlessly across multiple different platforms, such as different operating systems or hardware architectures.
- **Firmware** - The software that is permanently stored in read-only memory (ROM) or flash memory on electronic devices, controlling the device's specific functionality and operations.
- **Risk Management** - The process of identifying, assessing, and mitigating risks associated with a project, system, or process to minimize potential negative impacts.
- **Scalability** - The ability of a system, software, or technology to handle increasing workloads or accommodate a growing number of users or devices without significant performance degradation.
- **Remote Update** - The capability of a system or software to be updated or upgraded remotely, often without requiring physical access to the device or system.
- **Attestation** - The process of verifying and providing evidence of the integrity, authenticity, and trustworthiness of a system or component.
- **Attestation Server** - A dedicated component or service that receives, validates, and stores attestation reports generated by the CROSSCON stack. It verifies the integrity and authenticity of connected devices, establishing a trusted communication channel and enforcing system security policies.
- **Testbed** - Represents a designated location equipped with the necessary devices and peripherals to facilitate testing activities for the system.
- **Test Plan** - A comprehensive document outlining the entry requirements and validation steps essential for fulfilling the specified requirement.

2 Methodology

2.1 Overview of the existing approaches

Numerous mechanisms exist in the market for generating Validation Criteria based on Use Cases and Requirements. This chapter will present a selection of these approaches, which we rely on to develop the process for the CROSSCON project.

The book *Managing Software Requirements: A Use Case Approach [3]* provides a comprehensive process for generating test cases based on design requirements. According to the authors, a single test case should be defined for each individual use case. The creation of such test cases involves a sequence of activities, including the identification of Use-Case Scenarios (analysing Use Cases to determine the possible flow during use-case realization and determining the number of test scenarios based on the possible flows), identification of Test Cases (analysing prepared test scenarios to determine input parameters, steps, and expected results according to project guidelines), identification of Test Conditions (analysing the minimum requirements for executing the Test Case), and adding data values to complete the test scenarios (analysing which parameters need to be set to obtain the intended result).

The book *Software Requirements [5]* emphasizes the parallel creation of test documentation during functional analysis. Tests generated at this stage should cover the normal flow of each use case, alternative flows, and take into account exceptions identified during elicitation and analysis. These tests are independent of implementation details, and as development progresses, testers should refine them into specific test procedures.

On the other hand, the book *A Practical Guide to Testing Object-Oriented Software [4]* describes testing as a distinct process from development. The authors introduce various testing concepts, including test case (a single test procedure), test suite (a group of test cases assigned to a specific functionality), and testing ratio (a coefficient determining solution correctness based on the ratio of positive tests to the total number of tests). They also outline a three-step method for creating test cases (analysis, construction, and execution and evaluation) and highlight the importance of risk management during test preparation.

2.2 Description of the CROSSCON project specifics

2.2.1 Assessing Requirement Fulfillment

The determination of whether a specific requirement has been fulfilled serves as the fundamental aspect of test procedures. However, it is imperative to acknowledge that assessing the satisfaction or non-satisfaction of a requirement relies on its scope and constituent elements.

Considering the aforementioned fact and the previously introduced concepts, it is reasonable to posit the following:

A test case can be deemed as PASSED solely if:

The pre-testing requirements defined in the Test case setup have been duly fulfilled.

The test has been executed in accordance with the Test case steps section, and

The outcomes of the aforementioned operations align with the factors specified in the Test case expected results section.

A test suite can be deemed as PASSED exclusively when all subordinate test cases have been designated as PASSED.

A test module can be deemed as PASSED solely when all subordinate test suites have been labelled as PASSED.

A requirement can be considered as PASSED only when all subordinate test suites have been designated as PASSED.

2.2.2 Proposal of the methodology to follow in the next sections

The proposed mechanism for generating validation scenarios is based on the literature previously presented, our previous work, and the identified requirements. It involves a simplified two-step analysis process to ensure comprehensive coverage of the project's requirements and facilitate effective validation.

In the first step, each requirement is analysed individually to identify the main activities, actions, and interactions involved in fulfilling the requirement. This analysis takes into account the specific context in which the requirement is validated. The context describes the relevant conditions, such as the devices or components involved, their capabilities, and the environmental factors that influence the requirement.

Building upon the context and main activities, the second step dives deeper into each requirement to identify potential alternate paths or variations that may affect the outcome. This analysis considers different scenarios or conditions that may arise during the validation of the requirement. By exploring these alternate paths, the project team can anticipate different possibilities and ensure that the validation covers a wide range of potential scenarios.

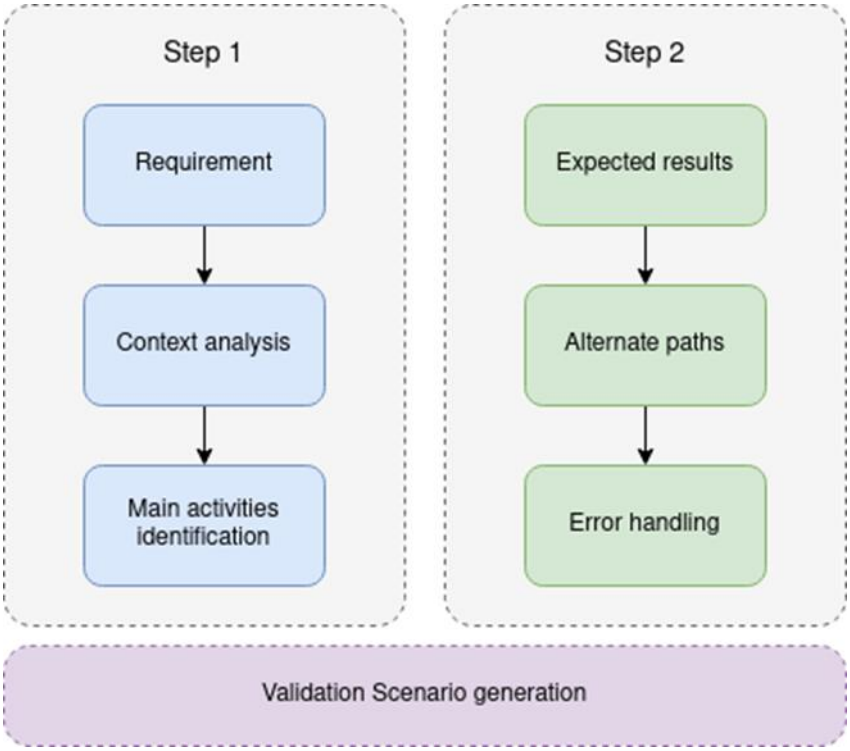

Figure 1: Validation Scenario generation

For each requirement, the expected results are defined, establishing the criteria for determining whether the requirement has been successfully fulfilled. These expected results provide a clear benchmark against which the system's functionality and compliance can be evaluated.

Additionally, alternate paths are considered to account for possible deviations from the expected flow. These sequences represent varying branches within the validation scenario that can be followed if certain conditions or actions diverge from the standard path. This often involves managing error scenarios, exception situations, or unique system behaviour that might present during the execution of the requirement. By pinpointing these alternative sequences, suitable procedures for handling errors can be developed, enabling timely resolution of any complications that might emerge.

By following this simplified mechanism, the project team can effectively generate validation scenarios that provide a systematic approach to validate the system's functionality and compliance with the specified requirements. The process considers the specific context, preconditions, main activities, expected results, and alternate paths, thereby ensuring comprehensive coverage of the requirements and facilitating efficient validation activities throughout the project life cycle.

2.2.3 Proposal of the mechanism for prepared Validation Scenario evaluation

To ensure the uniqueness and correctness of the prepared validation scenarios, it is important to follow a systematic approach.

1) Begin by thoroughly reviewing the project requirements that serve as the foundation for the scenarios.
2) Gain a clear understanding of the expected functionality and behaviour outlined in the requirements.
3) Cross-check the validation scenarios with corresponding use cases or functional specifications, if applicable, to ensure alignment.
4) Validate the logical flow of each scenario, ensuring that the sequence of actions and interactions accurately represents the intended behaviour and fulfils the associated requirement.
5) Verify that the scenarios cover all relevant aspects and account for potential alternate paths or variations.
6) Evaluate the preconditions and expected results specified for each scenario, ensuring that they align with the desired prerequisites and outcomes.
7) Consider any alternate paths described in the scenarios, assessing whether they adequately cover variations or exceptional conditions.
8) Seek peer review and engage in team discussions to gather feedback and validate the uniqueness and correctness of the scenarios.
9) Incorporate any necessary refinements or enhancements based on feedback received.

   By following this systematic approach and regularly reviewing and refining the scenarios, one can ensure their uniqueness, accuracy, and effectiveness in validating the specified requirements.

3 Validation Scenarios

In this initial version of the deliverable, we are focusing on preparing Validation Scenarios specifically for the Use Case Requirements (UCX-Y). This is because the overall requirements of the project are still being defined, and the Use Case Requirements are more likely to remain valid. Additionally, by concentrating on the Use Case Requirements, we can exercise the approach of defining Validation Criteria and ensure that the core functionalities meet the intended objectives outlined in the Use Cases. The final version of this deliverable will expand the scope to cover additional aspects, allowing for a comprehensive validation process.

3.1 Requirement UC1-1

Requirement:

- The higher-end device (gateway) has to be able to authenticate a constrained device with MFA.

Validation Scenario:

- Context:
  - The terms “higher-end device” and “constrained device” are used as in Section 3.4 Assumptions and Security Properties of the D1.1 [1] document.
  - Traditional first authentication such as credentials (name / password), or cryptography (public / private key, certificate).
  - Both devices have the necessary hardware and software components for authentication.
  - The authentication process involves secure communication and validation of device credentials.
- Preconditions:
  - The gateway and lower-end device are operational and running the CROSSCON stack.
  - Both devices are in the range of the established communication network.
  - The constrained device has been provisioned with the gateway first (so the gateway can is aware of the credentials / key to be expected from the certain device).
- Actions and Interactions:
  - The constrained device sends an authentication request (using traditional first authentication factor first) to the gateway.
  - The gateway receives the authentication request from the constrained device.
  - The gateway verifies the credentials provided by the constrained device.
  - If the credentials are valid, the gateway sends an authentication response to the constrained device, prompting for second authentication factor.
  - The constrained device receives the authentication response from the gateway, and provides the second authentication factor using the CROSSCON stack.
- Expected Results:
  - The constrained device receives the response to authentication using second authentication factor and acknowledges it.
- Alternate Paths:
  - If the credentials (first authentication factor) provided by the lower-end device are invalid, the gateway sends a negative authentication response and does not prompt for the second authentication factor.
  - If the second factor is invalid, the gateway sends a negative authentication response.

3.2 Requirement UC1-2

Requirement:

- Two higher-end devices, like gateways, have to be able to mutually authenticate themselves using MFA.

Validation Scenario:

- Context:
  - The terms “higher-end device” and “constrained device” are used as in Section 3.4 Assumptions and Security Properties of the D1.1 [1] document.
  - Traditional first authentication such as credentials (name / password), or cryptography (public / private key, certificate).
  - Both devices have the necessary hardware and software components for authentication.
  - Mutual authentication involves bidirectional verification of device credentials.
- Preconditions:
  - Both gateways are operational on and running the CROSSCON stack.
  - Both gateways are in the range of the established communication network.
  - The gateways have been mutually provisioned first (so they are aware of the credentials / key to be expected from the second device).
- Actions and Interactions:
  - Authentication of Gateway A by Gateway B:
    - Gateway A initiates the mutual authentication process with Gateway B.
    - Gateway A sends an authentication request to Gateway B, including its credentials.
    - Gateway B receives the authentication request from Gateway A.
    - Gateway B verifies the credentials provided by Gateway A.
    - If the credentials are valid, Gateway B sends its authentication response to Gateway A, prompting for a second authentication factor.
    - Gateway A receives the authentication response from the Gateway B, and provides the second authentication factor using the CROSSCON stack.
    - If the credentials are valid, Gateway B sends an acknowledgement to Gateway A.
  - Authentication of Gateway B by Gateway A:
    - Gateway B initiates the mutual authentication process with Gateway A.
    - Gateway B sends an authentication request to Gateway A, including its credentials.
    - Gateway A receives the authentication request from Gateway B.
    - Gateway A verifies the credentials provided by Gateway B.
    - If the credentials are valid, Gateway A sends its authentication response to Gateway B, prompting for a second authentication factor.
    - Gateway B receives the authentication response from the Gateway A, and provides the second authentication factor using the CROSSCON stack.
    - If the credentials are valid, Gateway A sends an acknowledgement to Gateway B.
- Expected Results:
  - Gateway A receives a valid authentication response from Gateway B.
  - Gateway B receives a valid authentication response from Gateway A.
- Alternate Paths:
  - If the credentials (first authentication factor) provided by either gateway are invalid, the receiving gateway sends a negative authentication response.
  - If the second factor provided be either gateway is invalid, the gateway sends a negative authentication response.

3.3 Requirement UC2-1

Requirement:

- The device has to be able to get a unique identifier (ID) that can be used to identify itself to the server.

Validation Scenario:

- Context:

- The unique ID allows for unambiguous identification of the device within the server infrastructure.
- Validation ensures that the unique ID obtained through the CROSSCON stack is reliable, unique, and consistent.

- Preconditions:
  - At least two devices running CROSSCON stack are available.
- Actions and Interactions:
  - The devices initiate the process of obtaining a unique ID from the CROSSCON stack.
- Expected Results:
  - The device receives the unique ID from the CROSSCON stack.
  - Generated ID is the same for future generations on the same device.
  - Generated ID is different between two devices.
- Alternate Paths:
  - If the device fails to obtain a unique ID from the CROSSCON stack, appropriate error handling procedures should be followed.

3.4 Requirement UC2-2

Requirement:

- The device has to be able to download the firmware image.

Validation Scenario:

- Context:
  - Firmware image must be downloaded first, prior performing further checks, and installation.
- Preconditions:
  - The device running CROSSCON stack is operational and has the necessary resources for firmware image storage.
- Actions and Interactions:
  - The device is notified that the new firmware update is available.
  - The download process of firmware image is started. It can be initiated by both device or server.
- Expected Results:
  - Local copy of the firmware update image is downloaded successfully.
- Alternate Paths:
  - In case of download failure, the device should attempt to retry until successful. Following download failure reasons are to be considered:
    - network failure during download,
    - storage full during download,
    - network bandwidth decreased, so the firmware update cannot be downloaded before exceeding download timeout.

3.5 Requirement UC2-3

Requirement:

- The device needs to be able to store firmware image in such a way that it can only be accessed by the authorized services.

Validation Scenario:

- Context:
  - By ensuring that only the application intended to be updated has access to the firmware image, we can minimize the risk of unauthorized access or tampering.
- Preconditions:
  - The firmware image is already downloaded.

- The CROSSCON stack is properly configured to manage secure storage.

- Actions and Interactions:
  - The CROSSCON stack securely stores the firmware image in a designated memory location.
  - Access control mechanisms are applied to restrict access to the memory location.
- Expected Results:
  - Access to the stored firmware image is denied to any application other than the updating application.
  - The updating application successfully retrieves the firmware image from the secure memory location.
- Alternate Paths:
  - If unauthorized access is detected or attempted, the CROSSCON stack should block the access and trigger appropriate security measures.

3.6 Requirement UC2-4

Requirement:

- The update should only be applied after ensuring the update's integrity and authenticity.

Validation Scenario:

- Context:
  - The firmware image needs to be validated to ensure it has not been altered or compromised during transmission or storage.
  - It is essential to validate that the firmware image has been authored by the expected entity or source.
  - Ensuring the integrity authenticity of the firmware image is crucial for maintaining the device's security and preventing unauthorized or malicious updates.
- Preconditions:
  - The device has downloaded and stored the firmware image.
  - The expected author or source of the firmware image is known and trusted.
  - The certificate of a trusted party (who will be signing update images) is already provisioned in the device.
- Actions and Interactions:
  - The device retrieves the expected hash or checksum value from the integrity verification data.
  - The CROSSCON stack calculates the cryptographic hash or checksum of the received firmware image.
  - The calculated hash or checksum is compared with the expected value.
  - The CROSSCON stack verifies the authenticity of the digital signature or certificate.
- Expected Results:
  - The calculated hash or checksum matches the expected value from the integrity verification data.
  - The digital signature or certificate is valid and matches the expected author or source.
  - The firmware image passes the integrity and authenticity check and is considered unaltered and authentic.
- Alternate Paths:
  - If the integrity check fails, the CROSSCON stack should trigger further actions to prevent the installation of a compromised firmware image.
  - If the authenticity check fails, the CROSSCON stack should trigger further actions to prevent the installation of an unauthorized or malicious firmware image.

3.7 Requirement UC3-1

Requirement:

- The device has to be able to get a unique identifier (ID) that can be used to identify itself to the server.

Validation Scenario:

- The Unique ID validation has been already discussed in another Validation Scenario. Please refer to the section: Requirement UC2-1.

3.8 Requirement UC3-2

Requirement:

- The device needs to be able to download the provisioning information.

Validation Scenario:

- Context:
  - Confidential provisioning information contains sensitive data, such as device certificates, that are crucial for secure device operation.
- Preconditions:
  - The device running CROSSCON stack is operational and has the necessary resources for provisioning data storage.
  - The device has established a secure connection to the provisioning server.
  - The device is authorized to access the confidential provisioning information.
- Actions and Interactions:
  - The device initiates a request to download the confidential provisioning information from the provisioning server.
  - The provisioning server authenticates the device's request and authorizes access to the confidential information.
  - The provisioning server securely transmits the confidential provisioning information to the device.
  - The device receives the confidential information and employs encryption mechanisms to protect its confidentiality.
- Expected Results:
  - The provisioning information is retrieved by the device.
- Alternate Paths:
  - In case of download failure, the device should attempt to retry until successful. Following download failure reasons are to be considered:
    - network failure during download,
    - storage full during download,
    - network bandwidth decreased, so the provisioning information cannot be downloaded before exceeding download timeout.

3.9 Requirement UC3-3

Requirement:

- The device needs to be able to store provisioning information in such a way that it can only be accessed by the authorized services.

Validation Scenario:

- The secure storage of downloaded data has been already discussed in another Validation Scenario. Please refer to the section: Requirement UC2-3.

3.10 Requirement UC4-1

Requirement:

- The device has to be able to get a unique identifier (ID) that can be used to identify itself to third parties.

Validation Scenario:

- The Unique ID validation has been already discussed in another Validation Scenario. Please refer to the section: Requirement UC2-1. On top of that, it is important to note, that the UC4-1 aims to expend the ID verification process, and this scenario is expected to be expanded in the final version of the deliverable.

3.11 Requirement UC4-2

Requirement:

- The device connects to the remote attestation server using a secure and authenticated communication channel.

Validation Scenario:

- Context:
  - Secure communication between the device and the remote attestation server is essential for transmitting sensitive information.
- Preconditions:
  - Both the device and the attestation server have the required cryptographic keys and certificates for initiating a secure communication.
- Actions and Interactions:
  - The device initiates a secure communication communication channel with the attestation server
- Expected Results:
  - The secure communication channel is properly established, which means that the network traffic is not readable in plain text by a third party.
- Alternate Paths:
  - If the device cannot establish secure communication channel, no sensitive information should be shared with the attestation server.

3.12 Requirement UC4-3

Requirement:

- The user can select which measurements are included within the remote attestation report of the device from a predefined list of possible measurements.

Validation Scenario:

- Context:
  - The selection of which measurements to include in the report provides flexibility and enables users to focus on measurements that are most relevant to their needs of attestation.
- Preconditions:
  - The device is operational and capable of generating the remote attestation report.
  - The user interface provides options for selecting measurements and configuring the remote attestation report.
  - The predefined list of possible measurements is available and up-to-date.
  - The user has the necessary permissions and privileges to modify the configuration of the remote attestation report.
- Actions and Interactions:

- The user accesses the user interface or configuration settings related to the remote attestation report.
- The user is presented with a predefined list of measurements that can be included in the report.
- The user saves the selected measurements as the configuration for the remote attestation report.
- The user triggers remote attestation report generation.

- Expected Results:
  - The user successfully saves the selected measurements as the configuration for the remote attestation report.
  - The generated remote attestation report contains the measurements selected by the user.
  - The generated remote attestation report does not contain the measurements not selected by the user.
- Alternate Paths:
  - If the user does not select any measurements from the predefined list, the system can either generate a default attestation report that includes all measurements or prompt the user to select at least one measurement before saving.
  - If the user attempts to select measurements that are not part of the predefined list, the system should prevent the selection and provide appropriate feedback or error messages to the user.

3.13 Requirement UC4-4

Requirement:

- The device is able to attest the status of its system to a remote verifier. The exact attestation procedure will be determined later on in the project, but shall implement a remote attestation report.

Validation Scenario:

- Context:
  - System status attestation report provides evidence of the device trustworthiness and system integrity to the remote verifier.
- Preconditions:
  - The device is operational.
  - The remote verifier is accessible and available for communication.
  - The device has established a secure communication channel with the remote verifier.
- Actions and Interactions:
  - The device receives a notification indicating that a new attestation process needs to be initiated.
  - The device gathers information about the system and performs measurements to be included in the attestation report, according to the attestation procedure.
  - The device sends the attestation report to the remote verifier over a secure communication channel.
- Expected Results:
  - The verifier receives a remote attestation report.
  - The verifier validates the received system status information against predefined security policies or requirements.
  - The verifier generates a response to the attestation report, indicating the outcome of the verification process.
  - The verifier securely transmits the attestation response back to the device.
- Alternate Paths:
  - If the verification process indicates a violation of security policies or requirements, appropriate actions should be taken by the device.

3.14 Requirement UC4-5

Requirement:

- The device provides an attestation conclusion (accepted or rejected), depending on the response of the remote attestation server to the delivered attestation measurements. Whether or not the attestation conclusion can be overwritten by the user, and if so under which conditions, will be determined later on in the project.

Validation Scenario:

- Context:
  - The device needs to provide an attestation conclusion based on the response received from the remote attestation server.
  - The conclusion will be either "accepted" or "rejected" depending on the evaluation of the delivered attestation measurements.
- Preconditions:
  - The device has successfully completed the process of generating remote attestation report.
  - The device is capable of interpreting the response from the remote attestation server and generating the attestation conclusion.
  - The user interface provides options for getting attestation conclusion status, and potential conclusion overwrite.
- Actions and Interactions:
  - The device receives the response from the remote attestation server, containing the evaluation of the delivered attestation measurements.
  - The device interprets the response and generates the attestation conclusion as either "accepted" or "rejected" based on the evaluation.
  - If user overwrite is allowed, the device checks the conditions specified to determine if the attestation conclusion can be overwritten.
  - If the attestation conclusion can be overwritten, user can use user interface to overwrite the attestation conclusion.
- Expected Results:
  - The device generates the attestation conclusion based on the evaluation of the response from server.
  - If user overwrite is allowed and the specified conditions are met, the conclusion from the user overwrites the one evaluated by the device.
- Alternate Paths:
  - If the user overwrite of the attestation conclusion is not allowed, the device does not provide an option for the user to modify or overwrite the attestation conclusion. The generated conclusion remains final.
  - If the specified conditions for attestation conclusion overwrite are not met, the device prevents the user from modifying or overwriting the attestation conclusion and provides appropriate feedback or error messages.

3.15 Requirement UC4-6

Requirement:

- The device can perform a remote attestation while in motion, including when no connection to the remote attestation server can be established.

Validation Scenario:

- Context:
  - The device can securely attest its integrity and system status even in dynamic environments and when network connectivity is limited or unavailable.

- Preconditions:
  - The device is operational and capable of generating the remote attestation report.
  - The remote attestation server is available but may not always have a reliable connection due to network limitations.
- Actions and Interactions:
  - The device initiates the remote attestation process while in motion, without a connection to the remote attestation server.
  - The device collects and securely stores the necessary attestation measurements locally, until a connection becomes available.
  - If a connection to the remote attestation server becomes available, the device attempts to establish a secure connection.
  - The device transmits the locally stored attestation measurements to the remote attestation server once the connection is established.
- Expected Results:
  - The remote attestation server receives the attestation measurements and proceeds with the evaluation and attestation process as in the case of UC4-1.
- Alternate Paths:
  - If a connection to the remote attestation server cannot be established at any point during the attestation process, the device continues to store the attestation measurements securely until a connection becomes available.
  - If the device loses network connectivity during the attestation process, it temporarily suspends the transmission of attestation measurements and securely stores them until network connectivity is restored.

4 Conclusions

The presented document has faced and addressed the challenge of translating Use Cases and Requirements from previous deliverables (D1.1 [1] and D1.2 [2]) into the newly introduced Validation Criteria. It has managed to weave together the outcomes from the past deliverables and form a comprehensive procedure to ensure the CROSSCON stack fulfills the identified Requirements. However, it must be noted that the Validation Criteria may evolve as the project progresses and encounters new challenges or features. The initial version of this deliverable is focused solely on the Use Case Requirements (UCX-Y), while many more requirements are expected to be present in the final version.

The primary result achieved in this document is a robust and adaptable Validation Criteria for the CROSSCON stack. Drawing from existing literature, it provides a systematic and efficient approach for generating validation scenarios. This includes a simplified two-step analysis process specifically tailored for the CROSSCON project, ensuring comprehensive coverage of the project's requirements and effective validation.

As for the project's further development, the Validation Criteria in this deliverable will be critical for the subsequent design stages. It provides valuable input for the WP2 Design Specification, Safety, and Assurance, the WP3 Development of the CROSSCON stack, and the WP5 Integration and Validation. The developed validation scenarios serve as a basis for the implementation of tests cases, the design of the testbed, and the decision-making process in evaluating if the CROSSCON stack solution meets the requirements.

In alignment with the project roadmap, the next steps include the application of the Validation Criteria in the development and integration stages of the project. This will ensure that the CROSSCON stack adheres to the established requirements and operates according to the initial project assumptions. Furthermore, future deliverables will take into account the approach and methodology outlined in this document, ensuring consistent progress towards the project's objectives.

References

[1] “D1.1 Use Cases Definition Initial Version” Deliverable of the CROSSCON project.

[2] “D1.2 Requirements Elicitation” Deliverable of the CROSSCON project.

[3] Dean Leffingwell, Don Widrig,`”Managing Software Requirements: A Use Case Approach” Addison-Wesley Professional, 01.2003, ISBN 978-0321122476

[4] John D. McGregor, David A. Sykes, “A Practical Guide to Testing Object-Oriented Software”, Addison-Wesley Professional, 01.2001, ISBN 978-0201325645

[5] Karl Wiegers, Joy Beatty, “Software Requirements”, Redmond (WA), Third Edition, Microsoft Press, 2013, ISBN 978-0-7356-7966-5