Net-Zero self-adaptive activation of distributed self-resilient augmented services

D6.1 Definition of the evaluation framework & Pilot specifications

Lead beneficiary	PNET	**Lead author**	Kostas Pournaras
Reviewers	Leonardo Padial (HES-SO), Francesco Paolucci (CNIT)		
Type	R	**Dissemination**	PU
Document version	v1.0	**Due date**	30/06/2025

Co-funded by the European Union

6GSNS

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

UK Research and Innovation

Project information

Project title	Net-Zero self-adaptive activation of distributed self-resilient augmented services
Project acronym	NATWORK
Grant Agreement No	101139285
Type of action	HORIZON JU Research and Innovation Actions
Call	HORIZON-JU-SNS-2023
Topic	HORIZON-JU-SNS-2023-STREAM-B-01-04 Reliable Services and Smart Security
Start date	01/01/2024
Duration	36months

Document information

Associated WP	WP6
Associated task(s)	T6.1
Main Author(s)	Kostas Pournaras, Kostas Lampropoulos (PNET)
Author(s)	Wissem Soussi, Gokcan Cantali, Gurkan Gur (ZHAW), Péter Vörös, Mohammed Alshawki (ELTE), Rana Abu Bakar, Francesco Paolucci, Layal Ismail, Abdul Khan (CNIT), Tom Goethals (IMEC), Antonios Lalas, Asterios Mpatziakas, Alexandros Papadopoulos, Ioanna Angeliki Kapetanidou, Sarantis Kalafatidis, Evgenia Vogiatzi, Eleni Chamou, Donatos Stavropoulos, Thanasis Korakis, Anastasios Drosou (CERTH), Vinh Hoa La, Manh Dung Nguyen, Edgardo Montes de Oca (MONT), Sumeyya Birtane, Mays AL-Naday (UEssex), Jorge Pose Eiroa, Julio Suárez Gómez (GRAD), Maria B. Safianowska (ISRD), Leonardo Padial, Joachim Schmidt (HES-SO)
Reviewers	Leonardo Padial (HES-SO), Francesco Paolucci (CNIT)
Type	R — Document, report
Dissemination level	PU — Public
Due date	M18 (30/06/2025)
Submission date	04/07/2025

Document version history

Version	Date	Changes	Contributor (s)
v0.1	22/01/2025	Draft initial document for ToC validation	Kostas Pournaras, Kostas Lampropoulos (PNET)
v0.2	27/02/2025	First inputs on section 3	All authors
v0.3	03/04/2025	Addition of section, initial inputs for sections 4 & 5	All authors
v0.4	19/05/2025	Further inputs and refinements	All authors
v0.5	07/06/2025	Version ready for internal review	Kostas Pournaras, Kostas Lampropoulos (PNET)
v0.6	15/06/2025	Review complete and feedback to co-authors	Leonardo Padial (HES-SO), Francesco Paolucci (CNIT)
v0.7	16/06/2025	Editor's addressing of review comments	Kostas Pournaras, Kostas Lampropoulos (PNET)
v0.8	18/06/2025	Addressing of leftover review comments, ready for quality review	All authors
v0.8.5	27/06/2025	Quality review complete	Joachim Schmidt (HES-SO)
v0.9	29/06/2025	All comments addressed, latest version delivered to projects' coordinator	Kostas Pournaras, Kostas Lampropoulos (PNET)
v0.9.5	02/07/2025	Final review and refinements	Asterios Mpatziakas, Evangelos V. Kopsacheilis, Antonios Lalas (CERTH)
v1.0	04/07/2025	Final version ready for Submission	Antonios Lalas (CERTH)

Disclaimer

Copyright message

Table of Contents

List of acronyms and abbreviations

Abbreviation	Description
AES	Advanced Encryption Standard
AI	Artificial Intelligence
AMF	Access Mobility Function
AUSF	Authentication Server Function
CIA	Confidentiality, Integrity and Availability security attributes
CNL	Cloud Native Lab
CNF	Cloud-native Network Function
CNN	Convolutional Neural Network
CQI	Channel Quality Indicator
CRD	Custom Resource Definition
CSI	Channel State Information
CTI	Cyber Threat Intelligence
CVSS	Common Vulnerability Scoring System
D-MUTRA	DLT-based mutual remote attestation.
DDoS	Distributed Denial of Services
DFE	Decentralized Feature Extraction
DQN	Deep Q-Learning
DoS	Denial of Services
DoSt	Denial of Sustainability
eNB	Evolved Node B
FDD	Frequency Division Duplex
gNB	Next Generation Node B
IDS	Intrusion Detection System
IoT	Internet of Things
IPC	Instruction Per Cycle
KDR	Key Disagreement Rate
KGR	Key Generation Rate
KPI	Key Performance Indicator
KVI	Key Value Indicator
LLM	Large Language Models
LoS	Line of Sight
MAB	Multi-Armed Bandits
ML	Machine Learning
MIMO	Multiple Input Multiple Output
MMT	Montimage Monitoring Tools
MTD	Moving Target Defense
MTTD	Mean Time to Detect
MTTR	Mean Time to React
NCL	Network Convergence Laboratory

Abbreviation	Description
NIST	National Institute of Standards and Technology
PKG	Physical Key Generation
PMC	Processor Monitoring Counter
PRB	Physical Resource Block
RL	Reinforcement Learning
RBM	Results-Based Management
QoS	Quality of Service
SDN	Software Defined Networking
SDR	Software Defined Radio
SECaaS	Security-as-a-Service
SINR	Signal-to-Interference-and-Noise Ratio
STIX	Structured Threat Information eXpression
TAXII	Trusted Automated eXchange of Intelligence Information
TDD	Time Division Duplex
TPM	Trusted Platform Module
UDM	User Data Management
USRP	Universal Software Radio Peripheral
UE	User Equipment
WAI	Wirespeed AI Offloading
ZSM	Zero-touch network and Service Management
ZSSM	Zero-touch Security network and Service Management

List of figures

List of tables

Executive summary

Deliverable D6.1 “Definition of the Evaluation Framework & Pilot Specifications” outlines the strategic approach for validating the NATWORK framework, focusing on how the project will demonstrate the performance, security, and sustainability of its proposed solutions through real-world testing environments and scenarios.

This deliverable extends the work done in WP2, presenting KPIs, requirements and the projects’ strategy to evaluate the technologies proposed and developed by NATWORK in its testbeds.

The document first introduces the evaluation strategy, highlighting the key validation aspects and the overall plan for evaluating KPIs and requirements. This sets the foundation for assessing how well NATWORK solutions meet their intended goals.

Next, the document presents the final setups for each pilot, including the involved services and architectures. These are accompanied by validation scenarios, each of which is directly linked to relevant KPIs and requirements to ensure that the tests reflect what the project set out to prove.

Both KPIs and requirements are organized in a clear, tabular format, showing their association with specific services, testbeds, baselines (where available), and means of verification. The document also includes the KVIs, each mapped to the most relevant KPIs to support their evaluation and demonstrate NATWORK’s added value.

Finally, a dedicated section outlines how the project plans to handle any obstacles and barriers encountered during validation, offering a practical methodology and concrete examples to ensure smooth integration and testing across all scenarios.

1. Introduction

NATWORK is a forward-looking framework designed to meet the growing need for networks that are not only fast and secure, but also energy-efficient and adaptable. Inspired by how natural systems maintain balance and protect themselves, NATWORK introduces mechanisms that help next 6G networks adjust their performance and security in real time—without wasting resources. It takes this further by using continuous machine learning (ML) to detect and respond to new threats as they appear, building resilience across all parts of the network, from the cloud to the edge. With a focus on practical implementation, NATWORK shows how future networks can strike a meaningful balance between sustainability, performance, and security.

This document is structured into several sections, each addressing important aspects of the final setups of the pilots and the evaluation framework of the NATWORK project. It will give a clear view of the pilot setups, the position of the services and the validation scenarios that will be performed. Moreover, KPIs and requirements are addressed and linked to services and the validation scenarios performed. Additionally, KVIs are mapped to relevant KPIs, in order to showcase their successful evaluation. Finally, a methodology as well as some indicative obstacles and barriers are presented, as a means to depict how NATWORK will deal with any issues encountered during the validation activities.

1.1. Purpose and structure of the document

This document aims to define the evaluation strategy and describe the final testing environments, including the testbed, the architecture and the complete setup that will be used for assessment within NATWORK. It will also illustrate all the validation scenarios that will be performed by each Use Case. Furthermore, KPIs, requirements and KVIs are depicted that will undergo the evaluation process. On top of that, a strategy for obstacles and barriers that may surface during the integration and validation process will be showcased with a few examples as well.

The document's structure after the introduction unfolds as follows:

Section 2 – Evaluation Strategy: Presents validation aspects of the project and the process for evaluating KPIs and requirements.

Section 3 – Use Case Description: Showcases the final pilot setups, the involved services and describes the validation scenarios.

Section 4 – KPI Evaluation: Depicts KPIs' links to services, testbeds and provides the means of verification, target values and baselines.

Section 5 – Requirements Evaluation: Depicts UC and service derived requirements, their mappings to services and the means of verification.

Section 6 – KVIs Evaluation: Presents the project KVIs associated with relevant KPIs.

Section 7 – Obstacles and Barriers: Provides the plan for addressing issues encountered during validation.

Section 8 – Conclusions: Summarises the conclusions stemming from the previous sections.

1.2. Intended Audience

The NATWORK Project's "Definition of the evaluation framework & Pilot specifications" is devised for public use in the context of planning and defining the pilot setups, the validation activities and the evaluation framework of the 6G Use Case Scenarios of the NATWORK consortium, comprising members, project partners, and affiliated stakeholders. This document mainly focuses on the 6G Use Case final testing environments, validation scenarios, evaluation of KPIs, and requirements of the project, thereby serving as a referential tool throughout the activities of WP6 as well as the project's lifespan.

1.3. Interrelations

The NATWORK consortium integrates a multidisciplinary spectrum of competencies and resources from academia, industry, and research sectors, focusing on user-centric service development, robust economic and business models, cutting-edge cybersecurity, seamless interoperability, and comprehensive on-demand services. The project integrates a collaboration of fifteen partners from ten EU member states and associated countries (UK and CH), ensuring a broad representation for addressing security requirements of emerging 6G Smart Networks and Services in Europe and beyond.

NATWORK is categorized as a "Research Innovation Action - RIA" project and is methodically segmented into 7 WPs, further subdivided into tasks. With partners contributing to multiple activities across various WPs, the structure ensures clarity in responsibilities and optimizes communication amongst the consortium's partners, boards, and committees. The interrelation framework within NATWORK offers smooth operation and collaborative innovation across the consortium, ensuring the interconnection of the diverse expertise from the various entities (i.e., Research Institutes, Universities, SMEs, and Large industries) enabling scientific, technological, and security advancements in the realm of 6G.

The D6.1 Definition of the evaluation framework & Pilot specifications document is directly associated with T6.1 "Testing environment definition, UC Requirements for deployment" and serves as a plan for all activities of the NATWORK project related to WP6. Additionally, it is relevant to WP2, WP3, WP4, and WP5, since it receives input from WP2 and maps its content against the developed solutions of WP3, WP4 and WP5.

2. Evaluation Strategy

2.1. Introduction

The evaluation methodology for D6.1 integrates best practices from the referenced literature to provide a robust framework for assessing the NATWORK system. This approach ensures the alignment of validation metrics with research, industry and project-specific requirements. This section initially provides an overview of various evaluation methodologies. Then it proceeds with the definition of the validation aspects, the evaluation planning and the structure for technical evaluation, system coverage analysis, and assessment consolidation.

Evaluation (term) is the systematic process of assessing a project, program, or system to determine its effectiveness, efficiency, and impact. Its purpose is to provide evidence on whether specific goals and requirements have been met and by applying appropriate methodologies, evaluation helps stakeholders make better decisions and improve aspects of the assessed unit, as well as ensure that the end-user is satisfied with the result. It also serves to validate outcomes, ensuring that the unit delivers intended results in a sustainable manner. **Evaluation methodologies** provide structured approaches to assess the performance, quality, and impact of a project, program, or system and they define how the evaluation is conducted. These methodologies are typically classified into two main categories: qualitative and quantitative. Qualitative methods focus on gathering in-depth insights through techniques such as interviews, case studies, and focus groups, making them ideal for understanding complex issues like user satisfaction or stakeholder perspectives. On the other hand, quantitative methods emphasize numerical data collection and statistical analysis, which are crucial for measuring specific performance metrics such as system throughput, response time, or defect rates. Of course, it is possible to use a mixed-methods approach, which is a combination of the two aforementioned methods.

Evaluation types or techniques refer to the focus or purpose of the evaluation. They define what aspect of the project or system is being evaluated and when the evaluation occurs, providing help in structuring the evaluation based on its objective. Examples include (a) Formative Evaluation which is a type of evaluation during the early development phase (b) Summative Evaluation which occurs after the program has been completed (c) Process Evaluation which assesses the implementation and operation of a system according to the plan, exploring how it reaches its short and long-term goals (d) Outcome Evaluation that focuses on the short-term or initial impact and effects on participants or stakeholders and assesses whether the project achieved its intended outcomes (c) Impact Evaluation which assesses the long-term, sustained effects of a project or program on its target population etc.

Further to these, **Evaluation Frameworks** are structured approaches or models used to plan, execute, and interpret evaluations. They are not tied to specific evaluation types but can guide the process for various purposes. Some examples include (a) the Logical Framework Analysis which is a structured approach to planning, managing, and evaluating projects, often referred to as Objectives-Oriented or Goals-Oriented Planning (b) Theory of Change which is a framework used to articulate how and why a program or initiative is expected to bring about change, (c) Outcome Mapping based on a participatory approach designed to monitor and evaluate behaviour changes among stakeholders and their contribution to achieving desired outcomes (d) Results-Based Management (RBM) which is an approach that relies on defining clear objectives, developing a results framework, and continuously tracking progress against measurable indicators to make informed decisions etc. By carefully selecting and applying the right evaluation methodologies and frameworks, organizations can ensure that their projects not only meet their objectives but also deliver meaningful and sustainable results. From the above, the RBM approach seems to be closer to the aspects and processes of EU projects.

2.2. Validation aspects

The validation of the NATWORK system focuses on evaluating the core quality attributes of the system. The assessment ensures that the system meets requirements related to:

1. **Security**: Assessment of confidentiality, integrity, authentication, and accountability features. Metrics include encryption strength, threat detection rates, and non-repudiation mechanisms.
2. **Reliability**: Evaluation of the system's availability and ability to perform consistently without failures under defined conditions, including fault tolerance and recovery capabilities.
3. **Functional Stability**: Verification that the functionalities operate correctly across different scenarios, maintaining consistent behaviour under variable workloads.
4. **Performance Efficiency**: Measurement of resource utilization, response times, throughput, and optimization in system processes.
5. **Compatibility**: Testing the ability of NATWORK components to integrate and operate with external systems without conflicts or performance degradation.
6. **Portability**: Verification that the system can be deployed across different environments with minimal modifications, ensuring flexibility for diverse operational contexts.

Each validation aspect can either be supported by quantitative and/or qualitative evidence gathered during pilot operation and complements the core validation of the project ensuring long-term viability, integration capability, and scalability of the NATWORK solution.

2.3. Technical Evaluation

As a Research and Innovation Action project, NATWORK validation is mostly focused on the research and technical aspects of the framework and solutions, without targeting an evaluation on its business perspectives. The analysis will ensure that the evaluation activities comprehensively address all functionalities and components of the NATWORK framework and involves:

- Mapping pilot activities to system functionalities and requirements.
- Identifying coverage gaps or areas where additional validation is required.
- Consolidating pilot evidence to confirm alignment with project KPIs and requirements.

The objective is to achieve complete and balanced validation across all system dimensions.

2.4. Assessment Structure

The assessment structure organizes the validation process across pilots and evaluation cycles as follows:

- **Pilot Iterations**: Conducting iterative pilot deployments to refine and validate the system progressively.
- **Use of Evaluation Instruments**: Using questionnaires, workshops, and metric-based analysis systematically.
- **Cross-Pilot Consolidation**: Combining results from all pilots/use cases to produce a unified evaluation of the NATWORK solution.
- **Feedback Loop**: Integrating evaluation results into system refinements to ensure continuous improvement.

2.5. Evaluation of NATWORK Key Performance Indicators

The evaluation of the NATWORK system includes the systematic assessment of the Key Performance Indicators (KPIs) defined within the project. These indicators measure both the technical success of the developed solution and its broader impact in terms of user engagement, system applicability etc. The KPIs have been established early in the project and have been further refined through internal consultation with work package leaders. The evaluation during the pilot phase focuses on collecting evidence, analysing results, and verifying whether the defined targets for each indicator are met. The evaluation methodology integrates both quantitative and qualitative data collection methods to ensure a robust assessment of all defined indicators.

For each KPI, the following evaluation process is applied:

- **Definition of Measurement Criteria**: Each KPI is associated with clear, measurable criteria based on pilot execution data.
- **Data Collection**: Metrics are collected through pilot deployments, including logs, performance reports, user feedback, and technical assessments.
- **Thresholds and Targets**: The evaluation verifies whether the measured values meet or exceed the target thresholds established at project outset.
- **Reporting**: The outcomes for each KPI are documented, with supporting evidence included for traceability.

The comprehensive evaluation of KPIs ensures that the NATWORK solution delivers the expected technical performance and fulfils its operational objectives.

2.6. Evaluation of NATWORK Requirements

The evaluation of the NATWORK framework includes a comprehensive verification of the functional and non-functional requirements initially identified in *WP2*. These requirements are critical to ensure that the developed solution meets both the technical specifications and the operational needs. The validation process relies on data collected during pilot activities, structured reporting from technical partners, and direct assessments based on the behaviour and performance of the system components.

NATWORK requirements specify the essential capabilities and behaviours that the NATWORK system must demonstrate. These were identified in previous deliverables and have been mapped to one or more use cases, components or workflows within the pilots.

The evaluation of requirements will follow this process:

- **Requirement Mapping**: Each requirement is associated with specific components or actions within the pilot environments.
- **Verification Activities**: Through pilot deployment and operational testing, evidence is collected to verify whether the functionality is implemented and operates as intended.
- **Measurement and Observation**: Metrics related to performance, reliability, and usability are systematically collected during pilot execution.
- **Qualitative Feedback**: Structured feedback from pilot participants and technical experts is used to improve solutions.
- **Compliance Assessment**: Each requirement is marked as fully met, partially met, or not met based on observed system behaviour and collected evidence.

The evaluation ensures that the NATWORK framework not only delivers the required functionalities but also operates reliably, securely, and efficiently in 6G networks.

3. Use Cases Description

This section provides a brief description of the project's use cases, presenting their objectives, technical implementation, and validation methodology. For each use case a general description is included and then more detailed technical specifications of its sub-use cases. More detailed information regarding the use cases can be found in D2.2.

The description of each use case begins with the Use Case Name and General Description, outlining its primary purpose and scope. This is followed by detailed descriptions of each associated Sub-Use Case, organized into several key topics:

- The description of each sub-Use Case specifying its objectives and functionalities.
- The Architecture detailing the technical design, relevant components, and their interactions.
- Information on the Testbed and Setup describing the deployment environment and configurations used for implementation.

Each sub-use case also identifies the Involved Services and Components, mapping them to the NATWORK services and listing critical system elements. The Validation Scenarios define the test cases and operational conditions under which the sub-Use Case will be evaluated. The scenario descriptions include the Goals, Metrics, and Expected Outcomes, identifying target KPIs (such as latency or throughput) and qualitative success criteria, enabling measurable assessment of performance against objectives.

3.1. Use Case 1: Sustainability and Reliability of 6G Slices and Services

Use Case 1 focuses on enabling sustainable and reliable 6G services by addressing the dual challenge of high energy consumption and increasing security threats. It demonstrates intelligent, intent-aware orchestration (Use Case 1.1), sustainable and secure software deployment via Security as a Servive (SECaaS) (Use Case 1.2), and energy-efficient workload placement using green energy and trusted runtimes (Use Case 1.3). Together, these solutions aim to optimize performance, trust, and energy use across edge-to-cloud infrastructures, ensuring secure, low-carbon, and resilient 6G service delivery.

3.1.1. Use case 1.1 Decentralised Management and Orchestration Service for Intent-compliant end-to-end Service Resiliency and Continuity

3.1.1.1. Description

Use Case 1.1 focuses on showcasing decentralized orchestration and management of 6G slices, tackling the critical issues of energy exhaustion in the cloud and its impact on service resiliency and sustainable continuity within edge-to-cloud networks. It highlights the NATWORK edge-cloud orchestration capabilities, through the simulation of Denial of Sustainability (DoSt) attacks on 6G slices and the consequent slice (re)configuration to mitigate the attack and continue compliance with resiliency requirements of the slice. Two key components are involved in the use case: the 6G-core decentralized orchestrator, which considers cluster risk factors when placing CNFs and establish relationships between them; and the CTI support system, which enables real-time sharing of threat intelligence across clusters and actively informs orchestration choices based on vulnerability evaluations. The initial phase entails implementing the FORK orchestrator and CTI solution on UESSEX's edge-cloud testbed, while the subsequent phase concentrates on expanding, refining, and assessing the system's performance in meeting use case needs. Ultimately, this use case seeks to confirm the security, sustainability, and dependability of 6G networks, illustrating a secure-by-design approach to slice orchestration and management.

3.1.1.2. Architecture, Testbed and Setup

The architecture for Use Case 1.1 is designed to support decentralized management and orchestration of 6G slices across an edge-to-cloud computing continuum, leveraging UEssex's NCL testbed infrastructure. NCL, located at the University of Essex (UK), is a state-of-the-art edge-cloud research data centre featuring over 200+ CPUs, 200+ TB of storage, and a programmable SDN/P4 network with 180 Gbps SDN and 100 Gbps P4 capabilities. This setup mimics a 6G edge-to-cloud environment, with compute and storage clusters managed by Kubernetes, and network control facilitated by ONOS. The testbed currently integrates the FORK orchestrator for 6G core management, and the Cyber Threat Intelligence (CTI) solution, enabling real-time communication and coordination across distributed infrastructure elements. The FORK [1] Solution will be used as baseline to demonstrate the DoSt attack as shown in Figure 1, while extending it to a NATWORK orchestrator will showcase slice resiliency. The demonstration will be conducted on the UEssex testbed infrastructure.

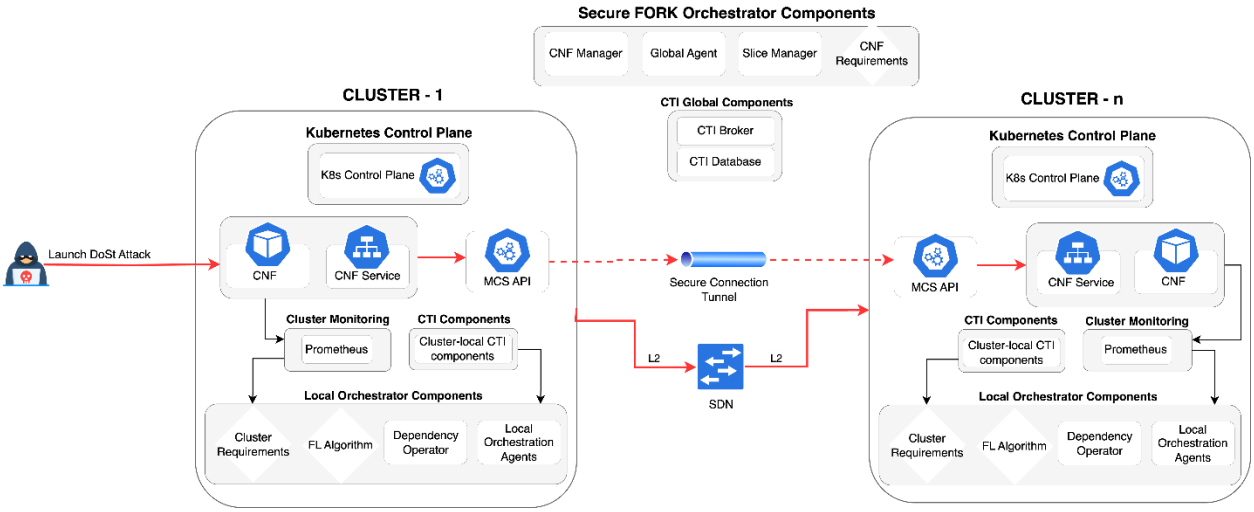

Figure 1: DoSt Demonstration Diagram with FORK

The architecture incorporates a middleware CTI framework as shown in Figure 1 that enables adaptive, STIX/TAXII-compliant threat intelligence exchange between clusters, dynamically adjusting shared data based on vulnerability context and security needs. Telemetry is collected via Prometheus and Kubernetes interfaces, supporting real-time monitoring and attack response.

3.1.1.3. Involved Services and Components

Use Case 1.1 integrates several services and components. These components align with the ecosystem's goals of secure, sustainable, and resilient 6G network management:

- **Secure-by-design Orchestrator**: A state-of-the-art federated orchestration solution serving as the baseline for secure-by-design management of 6G slices. Positioned as the core orchestration engine, it coordinates CNFs across the edge-to-cloud continuum, guiding DoSt attack mitigation and optimising resource allocation.
- **CTI Solution**: A decentralised middleware framework for real-time Cyber Threat Intelligence exchange between clusters. It processes vulnerability data from security tools, enables effective and secure CTI sharing between CNFs and influences orchestration decisions. It plays a pivotal role in the ecosystem by enhancing security-driven orchestration and providing cluster hygiene insights.
- **Prometheus Telemetry**: Collects real-time performance and attack-related data.
- **AI-Driven Security Modules**: AI-based services for anomaly detection, payload protection, and network security, enhancing the CTI solution and FORK orchestrator by identifying threats and optimising responses.

3.1.1.4. Validation Scenarios

The validation of Use Case 1.1 involves two distinct phases with specific scenarios to assess the resiliency, sustainability, and reliability of the 6G slice management solution. These scenarios are designed to test the functionality of the secure-by-design orchestrator, CTI solution, and overall system resilience against DoSt attacks, while meeting predefined KPIs.

- **Phase 1: DoSt Attack Demonstration and Initial Response**
  - **Scenario**: Launch an HTTP-based DoSt attack using random request generators, causing continuous scaling of Kubernetes containers in a 6G slice. The attack simulates oscillating demand to disrupt sustainability.
  - **Validation Goals**:
    - Demonstrate the impact of DoSt on energy consumption (KPI 1.1).
    - Validate initial secure-by-design orchestration by deploying the Secure FORK orchestrator.
    - Test basic CTI exchange between clusters, sharing vulnerability data to inform orchestration decisions.
  - **Metrics**: CPU utilisation, and initial cluster hygiene scores (KPI 1.2).
  - **Expected Outcome**: Establish a baseline for orchestration and CTI functionality, with telemetry confirming attack effects and mitigation feasibility.
- **Phase 2: Scaled Evaluation and Optimization**
  - **Scenario**: Scale up the DoSt attack across an expanded NCL testbed and deploy NATWORK orchestrator (extending FORK) and CTI solutions. Introduce AI-driven anomaly detection and mitigation strategies.
  - **Validation Goals**:
    - Assess energy efficiency and sustainability under attack conditions (KPI 1.1).
    - Evaluate CTI solution performance, including adaptive information sharing (A-KPI 1.6, 1.7: Exposed/Hidden info ratios) and its influence on orchestration (e.g., placing high-security apps in trusted clusters).
    - Validate cluster hygiene scores (A-KPI 1.5) and their role in improving security posture and resilience.
    - Demonstrate service continuity and Net-Zero compliance via optimised orchestration.
  - **Metrics**: CPU utilisation, cluster hygiene scores, CTI data exchange ratios, mitigation response time, and visual KPI representations.
  - **Expected Outcome**: Confirm the solution's scalability, security enhancements, and energy optimisation, with comprehensive documentation of results and system effectiveness.

Both scenarios leverage the UESSEX NCL testbed, with Phase 2 building on Phase 1 insights to refine the system and meet NATWORK's broader objectives

3.1.2. Use case 1.2. SECaaS for CIA-hardening

3.1.2.1. Description

Use Case 1.2 showcases the SECaaS hardening against Confidentiality, Integrity and Availability (CIA) attacks, applied on both x86 compiled and Web Assembly payloads. Additionally, the use case shows the benefits of D-MUTRA blockchain-based remote attestation for hardened payloads.

3.1.2.2. Architecture, Testbed and Setup

The used testbed is integrated into TSS's premises. A general presentation of the use case workflow and testbed components is given in Figure 2. The testbed includes the following components:

- TSS's SECaaS modifies x86 and WASM payloads for their CIA hardening. Moreover, the SECaaS itself is a blockchain node and takes an active part in D-MUTRA mutual remote attestation. In D-MUTRA operation, the SECaaS generates reference quotes of the hardened payloads and serves as the seed of trust. As shown in Figure 2 the SECaaS interplays at both Build and Deploy phases as it hardens payloads before their deployment first and second takes an active part of the remote attestation of deployed payloads.
- D-MUTRA blockchain-based remote attestation and an ad hoc smart contract orchestrating the remote attestation, constructed over hyperledger fabric for performance and scalability.
- A set of hyperledger nodes, either hosting and executing the payloads or alternatively independent from the payloads execution environments.
- For WASM payloads, a specifically modified runtime installed on the execution hosts, with source level changes on WASMTIME open source interpreter from Bytecode Alliance. D-MUTRA remote attestation will be used to validate this modified WASM interpreter.

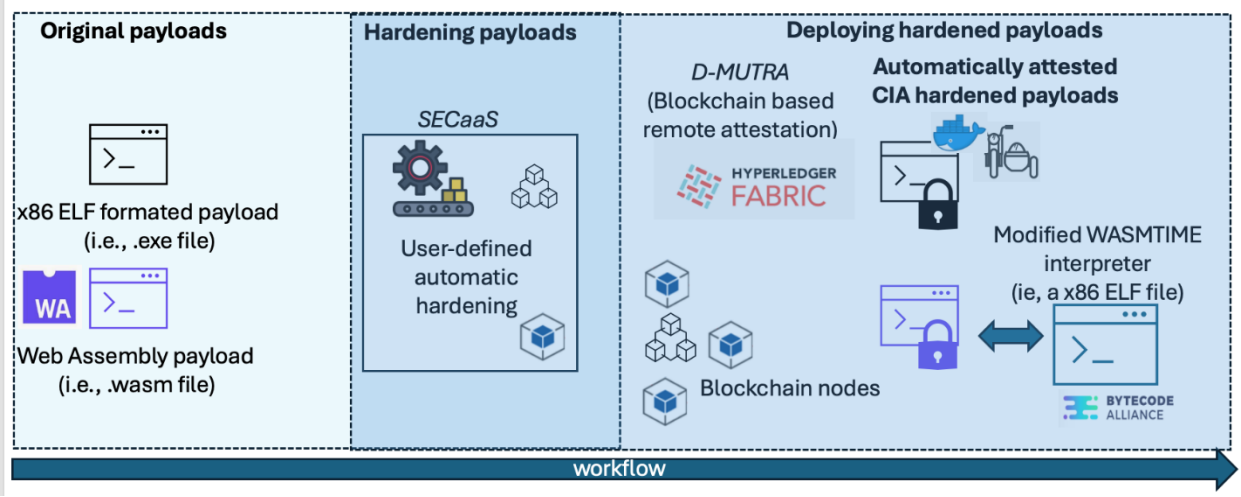

Figure 2: Use case 1.2 workflow and testbed components

The SECaaS implementation will be considered with the benefits of splitting the SECaaS into several functional entities, notably for the sake of reducing the workflow step of payload preparation stage for seamless deployment and especially for container packaged x86 payloads. In that direction, the service of remote attestation and of continuous integrity verification can be worked out by adding a sidecar-mounted container which constructs quotes and exchanges with D-MUTRA. The side car implementation main merit is to preclude to the SECaaS payload wrapping step, hence enabling original payloads (i.e., containers) to be deployed. However, this scheme is functionally restricted since confidentiality and availability hardening cannot be achieved without payload modification prior deployment.

3.1.2.3. Involved Services and Components

The use case implements the service of SECaaS and D-MUTRA, hardening the software payloads against CIA attacks. Its service components or security functions are given below.

CIA-hardening of x86 payloads:

- Confidentiality: x86 executable (i.e., .exe file, structured with Executable Linux Format) text section (i.e., payload instructions) are encrypted with an AES key. The key will be provisioned separately by D-MUTRA to decrypt the text section once the remote attestation has gone through a positive check.
- Integrity: x86 executable files are prepared to be D-MUTRA-ready for the service of mutual remote attestation once they are deployed.
- Availability: x86 executables are prepared to be self-monitored during their execution. Self-performance relies on control-flow inserted probes revealing that the payload execution runs correctly on its execution environment. This availability attribute is furthered in use case UC 4.6.

CIA-hardening of WASM payloads:

WASM hardening is operated through modification of WASM bytecode, directly associated by the modifications implemented on NATWORK's WASM interpreter.

- Confidentiality: WASM modules (i.e., .wasm file) are entirely encrypted. Their decryption is carried out by the modified WASM interpreter.
- Integrity: WASM modules are prepared for being un-ambiguously identified and their function and code sections (i.e., sections 3 and 10) are measured by the WASM interpreter during their execution.
- Availability: WASM modules execution effectiveness and performance ratio will be collected by the modified interpreter, applying the self-monitoring as defined above to the x86 ELF formatted interpreter.

3.1.2.4. Validation Scenarios

The goals of UC1.2 will be illustrated using three scenarios:

- **Scenario 1: Confidentiality attack**. Extract the payload for IPR violation, detection of vulnerability and targeted attack preparation.
- **Scenario 2: Integrity attack.** Replace or tamper an original payload before or during execution.
- **Scenario 3: Availability attack.** Resource attrition for the payload interruption or slow down.
- **Validation Goals**:
  - Demonstrate that the x86 and WASM payloads is duly AES 256 encrypted (prior bootstrap, decryption and execution). For that, the encrypted payloads are decrypted using the same AES key and a comparison with the original is produced.
  - Demonstrate that the x86 and WASM payloads tampering attack taking place either before onboarding or during execution is detected, with the generation of a tampering alert state over D-MUTRA. Used metrics: KPI 1.3.1 time for remote attestation. The same metrics is used for integrity verification.
  - Demonstrate that the x86 and WASM payloads interruption or slow down is instantly detected.
  - For these validation goals, common metrics will be also used and defined as KPI 1.3.2 performance degradation at runtime and KPI 1.3.3 Energy waste.

Expected Outcomes:

- Develop a novel runtime integrity verification for WASM payloads
- Develop a novel confidentiality protection for WASM payloads
- Develop a novel availability protection against x86 and WASM payloads

- Further D-MUTRA implementation for cloud native payloads, removing SECaaS wrapping stage when possible.

3.1.3. Use case 1.3 Green-based payload placement

3.1.3.1. Description

Use case 1.3 involves setting up a multi-location compute mesh with trusted computing-enabled hosts and verified sources of green energy information. This meshes the two main IMEC contributions to NATWORK, i.e. remote node attestation and decentralized orchestration based on dynamic node metadata, the latter in the form of green energy metrics. The use case is evaluated across UEssex and IMEC testbeds to illustrate the decentralized nature and compatibility with various devices. Additionally, it shows the technical feasibility of trustworthy net-zero payload placement while ensuring the security, integrity and confidentiality of the workloads and data.

3.1.3.2. Architecture, Testbed and Setup

The test setup involves a Kubernetes/Flocky cluster spanning multiple geographic locations and using Remotely Attested Kubernetes workers to ensure the trustworthiness of the compute. Device trust will be based on a Kubernetes-compatible device enrolment and attestation platform utilizing Trusted Platform Module (TPM) and attested boot functionality on the remote device. Kubernetes is used for attestation management – decentralized orchestration is managed by Flocky.

The main testbed components are the IMEC Virtual Wall, CloudNativeLab, CloudEdgeLab, and UEssex testbed infrastructure, each as required. The UEssex testbed is used to simulate physically remote devices running on infrastructure outside of the control of the workload owner, and provides data on energy use of workloads. A Green Energy Monitor agent will mock green energy availability, alongside (smaller scale) anonymized performance data from a SolarEdge edge location.

The Kubernetes control plane will be set up on CloudNativeLab, with KeyLime on devices from other nodes to enable remote attestation. No Kubernetes agent (kubelet) is required on worker nodes; this role falls to a combination of Flocky and Feather, which rely on the Green Energy Monitor and Kubernetes cluster for orchestration metadata. Each node will be provisioned with only a containerd runtime, to enable optimal monitoring and accuracy of evaluation metrics.

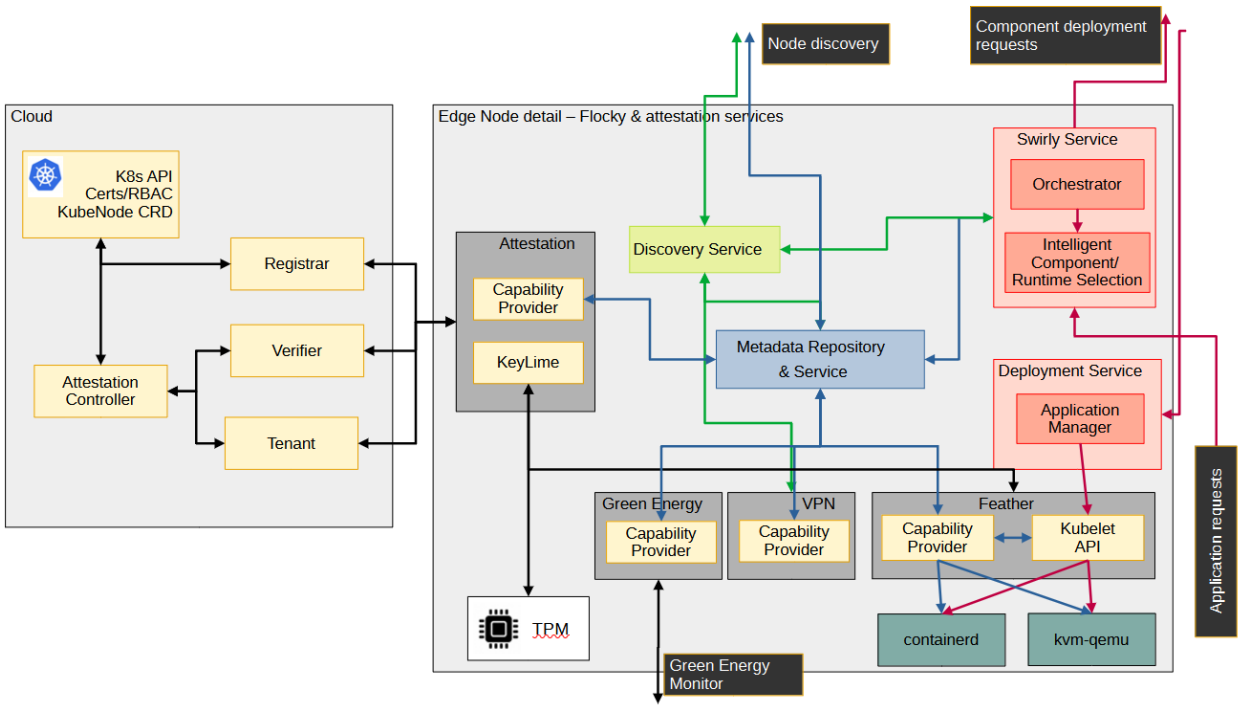

Figure 3: Use case setup including attestation and Feather/Flocky

3.1.3.1. Involved Services and Components

The component placement of all aspects discussed in 3.1.3.2 is shown in Figure 3. This overview illustrates the following services involved in the use case:

Cloud side

- Kubernetes: the TrustEdge attestation framework requires a Kubernetes control plane as a root authority; the main registry and verification logic operates here, and Kubernetes Custom Resource Definitions (CRDs) are used alongside the API to store node credentials.
- TrustEdge: the attestation controller/operator which verifies the integrity of nodes joining the cluster. Note that Kubernetes is only used for attestation; orchestration may also be performed by Flocky (edge component).
- Prometheus monitoring (not illustrated): used for additional metrics gathering on cloud & edge sides.

Edge devices

- TrustEdge: agent component of the attestation framework running on individual devices, monitoring device status and Trusted Platform Module (TPM).
- Feather: edge-designed payload deployment engine, similar to the Kubernetes kubelet. Detects various workload runtimes e.g. containerd or KVM-qemu.

- Flocky (discovery/metadata/swirly service): decentralized orchestration framework allowing edge devices to make localized decisions to offload workloads. Interfaces with various other software to detect device capabilities, and Feather for payload deployment.
- Green energy monitoring/services: green energy monitoring may be provided by either physical hardware or simulation services, providing additional data for UC1.3 energy efficient orchestration.
- Capability providers:
    - VPN (optional): a VPN may be integrated as Flocky capability provider for secure connections with other sites.
    - Green energy (optional): green energy may be explicitly integrated as a capability if required by the Flocky orchestration algorithm.

3.1.3.2. Validation Scenarios

The goals of UC1.3 will be illustrated using three scenarios:

- **Scenario 1**: Cross-site integration
- **Scenario 2**: Cross-site cloud-edge remote attestation
- **Scenario 3**: Green-based decentralized edge task scheduling
- **Validation Goals**:
    - Scenario 1 illustrates the ability of intent-based orchestration (Flocky) to model various node and software capabilities, with respect to requirements "**Intent-based**" and "**Hardware & infrastructure support".**
    - Scenario 2 extends the intent-based orchestration with attestation and full cross-functionality, as per requirement "**Cross-site orchestrator compatibility**".
    - Scenario 3 integrates green energy metrics and advanced orchestration methods, enabling "**Green energy awareness**". This scenario also includes intelligent node/runtime selection for both security (e.g. attestation) and payload-based energy optimization (e.g. efficient runtimes).
- **Metrics**:
    - Scenario 1 & 2:
        - Accuracy of (supported) capability detection and dissemination through cluster
        - Accuracy of attestation/trust mechanism (**A-KPI 1.8**)
        - Node setup times, communication latency (**A-KPI 1.9**)
    - Scenario 3:
        - Latency from workload scheduling to running
        - Energy savings between normal scheduling and energy-efficient scheduling (Wh or CO2Eq, if feasible).

- **Expected Outcome**:

  Development of an integrated framework capable of orchestration using custom metrics (i.e. green energy) while improving security aspects such as attestation w.r.t. state of the art, across different sites and (private) networks.

3.2. Use Case 2: Anti-Jamming Technologies for AVS

Autonomous vehicles (AVs) will heavily rely on 6G networks to communicate with other vehicles, infrastructure, and the cloud. However, the wireless links used by AVs are susceptible to various types of interference and jamming attacks, which can compromise the safety and reliability of the vehicle. Machine learning and AI can be used to detect, classify, and mitigate jamming attacks in real-time, by analysing signal patterns, adapting to changing signal environments and identifying anomalous behaviour. By leveraging the power of 6G networks and cutting-edge machine learning techniques, a safer and reliable future for AVs could be guaranteed. This use case will explore how advanced anti-jamming technologies can ensure reliable and secure communication for AV networks, enabling safer and more efficient transportation systems. Four UCs will be explored in this demonstrator, already mentioned in D2.2 and D2.3:

- UC#2.1: Enabling multi-antenna systems for resilience against jamming attacks.
- UC#2.2: Empowering AI-based jamming detection and mitigation for multi-path routing in 6G networks.
- UC#2.3: Adaptive modulation techniques for anti-jamming autonomous recovery.
- UC#2.4: Improving 6G security in 6G spectrum bands.

3.2.1. Sub-Use Case 2.1: Enabling Multi-antenna for resilience

3.2.1.1. Description

This sub-use case comprises three services: AI-based RIS configuration, ML-based MIMO, and AI-based anti-jamming. Their joint operation enhances physical layer security in V2X communication links by enabling timely jamming detection, real-time mitigation, and software-controlled EM propagation through RIS. The implemented tools, components and workflow can be easily expanded in other B5G/6G networks.

3.2.1.2. Architecture, Testbed and Setup

The architecture of UC2.1 is shown in Figure 4, detailing the services, their interconnections, and the necessary information flow. The AI-based anti-jamming mechanism includes a detection module, JASMIN, which operates solely on time-domain (I,Q) signal representations. JASMIN requires no jamming data during training and relies on two functions: persistent detection of the modulation used by the base station and comparison of normal versus current noise profiles. It delivers high-throughput, accurate decisions (up to 3500 per second) across all jamming types—

constant, periodic, and reactive. Upon detection, it activates the ML-based MIMO service that is dedicated to the jamming identification properties.

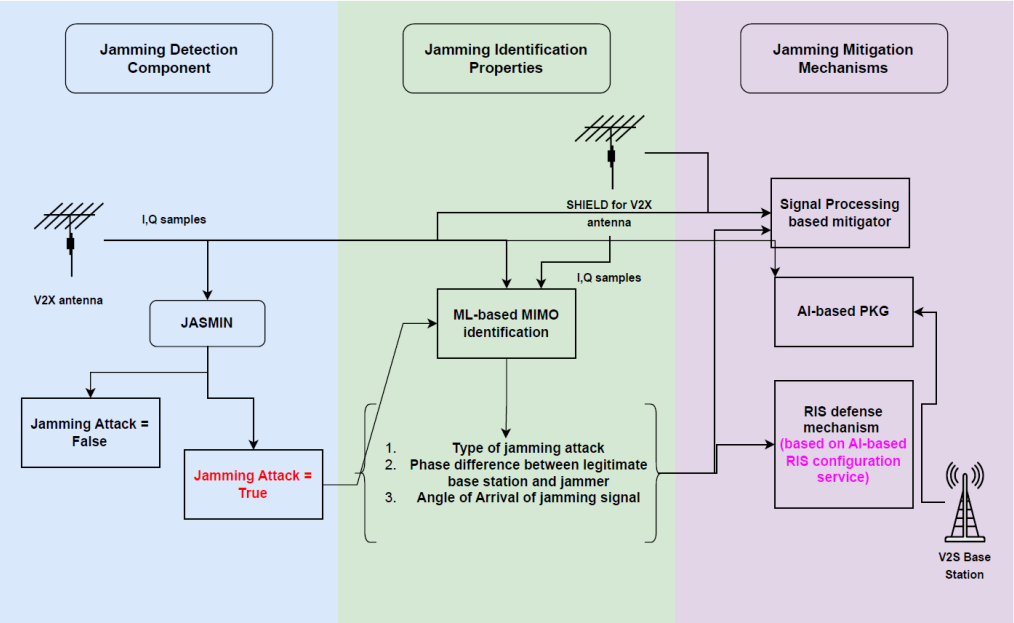

Figure 4: Architecture of UC2.1 components

This component requires the same (I,Q) input as JASMIN, augmented with data from an auxiliary antenna acting as a shield, forming a MISO configuration at the receiver. The correlation between the two antennas enables extraction of key jamming characteristics: attack type, interference-to-signal phase difference, and an estimate of the jamming signal's angle of arrival.

The extracted data, along with the (I,Q) inputs from both V2X antenna and its shield, are forwarded to the mitigation module. For the signal processing-based mitigator, input includes signal representation, jamming type, and interference-to-signal phase. The physical-layer key generation—targeting spoofing—requires pilot signals from both legitimate and base station. RIS-based mitigation relies on jamming angle of arrival and vehicle position, enabling software-defined EM control against jammers and eavesdroppers. RIS defence mechanism effectiveness depends on precise computation of its configuration relative to the base station and vehicle, provided by the AI-based RIS configuration service.

CERTH's testbed for service evaluation, illustrated in Figure 5, integrates:

- **Three USRP B210 SDRs [2]**: Covering 70 MHz to 6 GHz with up to 56 MHz real-time bandwidth.
- **Three NVIDIA Jetson Orin modules [3]**: Handling signal processing from SDR data.
- **Various antennas**: Including directional and omnidirectional types.

- **TMYTEK's XRifle Dynamic RIS unit [4]**: Enhancing 5G FR1 coverage with precise control over reflective angles.
- **CERTH AV.**

This setup facilitates JASMIN evaluation and AI-based RIS configuration implementation. Other components will initially undergo simulation-based development and, if feasible, transition to the SDR-based environment. Key challenges include legal constraints related to jamming attack deployment for AV receiver evaluation and limited synchronization capabilities among available SDR models.

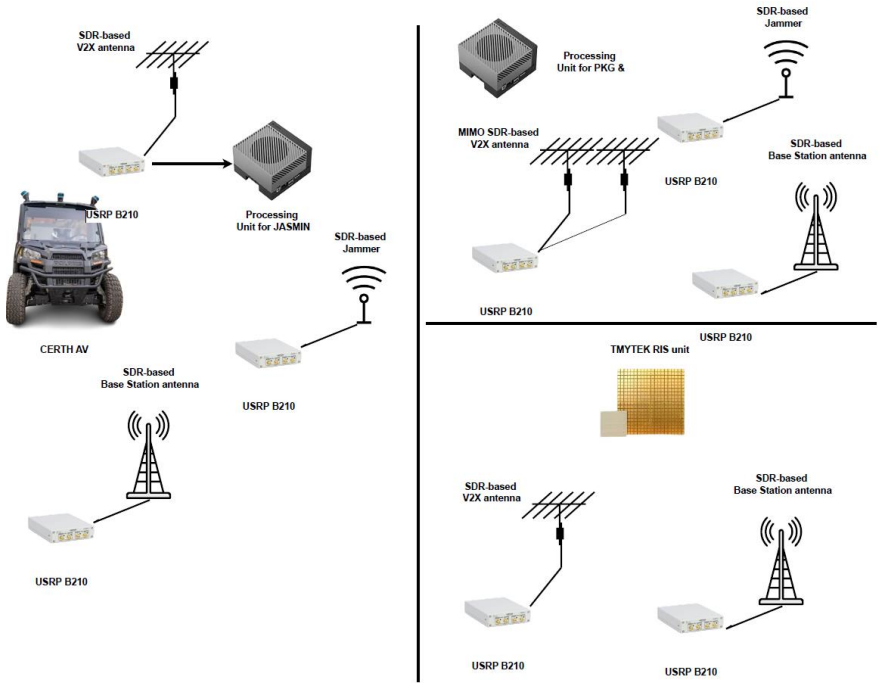

Figure 5: CERTH Testbed

3.2.1.3. Involved Services and Components

The involved services are three; the AI-based RIS configuration, the ML-based MIMO and the AI-based anti-jamming. Mainly the services with each component are:

- AI-based Anti-jamming: JASMIN for detection and Signal Processing based mitigator.
- ML-based MIMO: Jamming Identification. Also, AI-based PKG can be classified in this service.
- RIS as defence mechanism: AI-based RIS configuration.

3.2.1.4. Validation Scenarios

The validation will follow a dual approach: a simulation with parameters aligned to real conditions, and an SDR-based setup deployed in the CERTH lab. Three factors determine the feasibility of porting each service to the SDR setup:

1. Availability of all required components for complete evaluation.
2. Synchronization constraints versus what the SDR setup can support.
3. Legal limitations, primarily concerning jamming attack scenarios.

The validation procedure for each service in UC 2.1 is divided into two main phases, followed by their final integration.

AI-based Anti-jamming Service

Phase 1: JASMIN deployment & Preliminary Results

a. **Scenario**: A V2X base station transmits messages to a legitimate receiver embedded in an AV using the IEEE 802.11p protocol. Simultaneously, a jamming component emits interference on the same frequency band as the base station.
b. **Validation Goals**:
    i. High detection accuracy in different jamming signal levels and topology of the network (KPI 2.1, REQ 2.1-1).
    ii. On time detection of the jamming attack (KPI 2.2).
c. **Metrics**: Accuracy detection (%) and latency (s).
d. **Expected Outcome**: The detection model should achieve >99% accuracy across all jamming types and network topologies, with latency significantly lower than the protocol's sampling period.

Phase 2: JASMIN evaluation in SDR-based setup

e. **Scenario**: The first phase setup is accurately replicated in the SDR-based environment. The feasibility of transferring the final evaluation to real conditions, using the local 5G network and CERTH's AV, will be assessed against the identified limiting factors.
f. **Validation Goals**:
    i. High detection accuracy in different jamming signal levels and topology of the network (KPI 2.1, REQ 2.1-1).
    ii. On time detection of the jamming attack (KPI 2.2).
g. **Metrics**: Accuracy detection (%) and latency (s).
h. **Expected Outcome**: The detection model should achieve >99% accuracy across all jamming types, network topologies & (if the real-conditions evaluation is feasible to be done) AV speed, with latency significantly lower than the protocol's sampling period (KPI 2.5).

ML-based MIMO Service

Phase 1: ML-based MIMO components deployment

a. **Scenario**: At the AV antenna side, a MIMO setup includes an additional antenna serving as a shield. Besides jamming, spoofing attacks must also be addressed. An ML-based model identifies jammer properties—jamming type, phase difference with the V2X base station, and angle of arrival. This enables two mitigation mechanisms: a signal-processing jamming suppression filter and a physical-layer key generation scheme (PKG) to counter spoofing, ensuring network safety.
b. **Validation Goals**:
    i. **ML-based jamming properties identification**: High accuracy in respect of all the attributes prediction (KPIs 2.1,2.5, REQs 2.1-1,2.1-3).
    ii. **Signal-processing based Jamming mitigation**: Reconstruction of the signal removing the impact of the jamming attack (KPIs 2.1,2.5, REQ 2.1-2).
    iii. **PKG:** Definition of the framework that can be aligned with V2X needs.
c. **Metrics**: Accuracy detection (%), SNR and BER improvement in (%).
d. **Expected Outcome**: Error in identification prediction lower than 10%, jamming mitigation more than 50%, publication item for the framework of V2X PKG (KPI 2.5).

Phase 2: Synergy of the components in a unified framework

e. **Scenario**: The mentioned components from the detection up to the mitigation are smoothly cooperating creating a unified framework that can ensure the safety in V2X networks.
f. **Expected Outcome**: A low-latency, high-accuracy unified tool achieving consistent unimodal performance metrics (KPI 2.5).

RIS as a Defence Mechanism Service

Phase 1: AI-based RIS configuration

a. **Scenario**: The Line-of-Sight (LoS) between the V2X base station (BS) and the AV's receiver (Rx) is blocked due to a physical obstacle. An RIS unit with binary ON/OFF pin diode configuration is used in order to reconstruct it.
b. **Validation Goals**:
    i. **Reconstruction of LoS:** The path BS-RIS-Rx is created via the optimal configuration of the RIS (REQ 2.1-4).
    ii. **RIS configuration overhead**: A synergy of physical optics, metaheuristics, and AI-based pattern recognition is implemented to minimize the

computation time for RIS configuration, referred to as codebook compilation (REQ 2.1-4).

iii. **Physics-based codebook compilation**: The results and key post-processing insights will be analysed to define a step-by-step algorithm for physics-based codebook compilation (REQ 2.1-4).

c. **Metrics**: Enhancement of signal amplitude/SNR in the receiver, minimization of the required computational time for codebook compilation up to 50%.

d. **Expected Outcome**: A solid physics-informed, AI-based algorithm for codebook compilation in binary RIS configuration.

Phase 2: RIS-assisted network framework for physical layer security

e. **Scenario**: The V2X protocol operates in an open environment with multiple AVs, where potential threats include eavesdropping and link disruption. RIS units, strategically placed across the area, function collaboratively to enable precise electromagnetic wave control, enhancing physical-layer security against all such threats.

f. **Expected Outcome**: A RIS-enabled service, built on the codebook compilation procedure, ensures real-time computational feasibility for simultaneous QoS enhancement for legitimate users and mitigation of eavesdropping and jamming. The service aligns with proactive covert communication principles and operates via a dedicated algorithm (KPI 2.5, REQ 2.1-5).

3.2.2. Sub-Use Case 2.2: Empowering AI-based jamming detection and mitigation for multi path routing

3.2.2.1. Description

In this sub-use we will showcase NATWORK's novel approach that combines jamming detection and selection of countermeasures into a unified process and investigate innovative AI-driven techniques that consider both phases of jamming detection as a comprehensive process, ultimately contributing to the security of 6G networks. Also, the developed algorithms will be able to demonstrate the routing of traffic through multiple paths to avoid jammed channels and ensure that communication is not affected by jamming attacks. Finally, our machine learning-driven anomaly detection approach for pinpointing jamming attacks will be supported by an AI-supported jamming signal identification and characterization process, reinforced by a learning-based decision-making solution for effective jamming mitigation.

3.2.2.2. *Architecture, Testbed and Setup*

The architecture design of UC 2.2 faces the challenge of implementing a real-time solution that dynamically modifies the frequencies used by a communication channel in response to a detected jamming attack. The entire system will be referred to as DetAction. In the detection task there are two different blocks, one from GRADIANT and another one from HES-SO.

The GRADIANT detection process begins by collecting IQ samples from received signals, preprocessing them to separate different frequency bandwidths, and analysing them using a pre-trained deep learning algorithm to detect the presence of jamming. The output of this block will indicate whether the current PRBs are affected by a jamming attack.

In parallel, a second module uses metrics obtained from the 5G signal that represent the CSI (Channel State Information), like CQI (Channel Quality Indicator) or SINR (Signal-to-Interference-and-Noise Ratio) to determine the presence of jamming in the received signal.

Both outputs are then combined at the next block, which uses its output as the input for the reaction phase, where the system will reallocate the affected PRBs to a location free from jamming.

To achieve this, we propose an O-RAN-based architecture, implemented using BubbleRAN [5], featuring an xApp deployed within the Near-Real-Time RIC to handle resource allocation. This xApp will communicate with the detection block, which operates outside the O-RAN architecture and is connected to a USRP to acquire and process IQ samples. The detection phase output will be transmitted to the action phase via a REST API or a similar interface. The xApp will then adjust the PRB allocation using the O-RAN E2 interface, as illustrated in Figure 6 below.

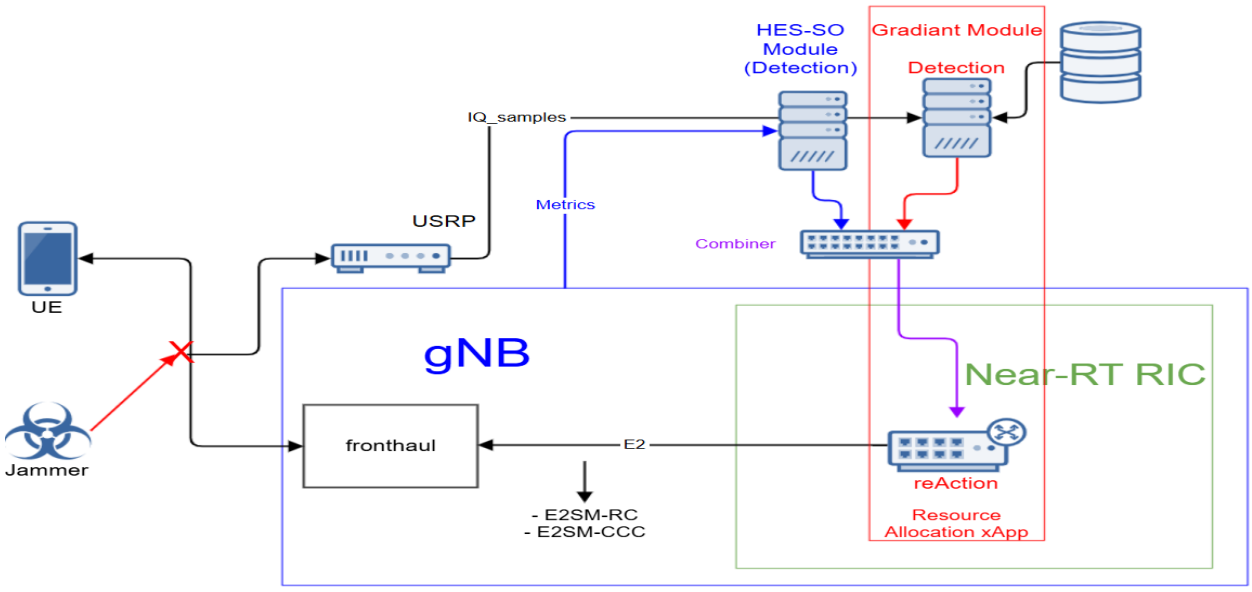

Figure 6: DetAction O-RAN architecture

3.2.2.3. Involved Services and Components

In Table 1 the services and components of the sub-UC 2.2 involved are presented:

Table 1: SubUC 2.2 services and components

Service	Component
AI-based anti-jamming	DetAction: Detection and reAction against jamming attacks (M)

3.2.2.4. Validation Scenarios

In order to validate the sub-use case 2.2, we need to define a scenario to test its KPIs. The DetAction module needs to validate its capabilities of detecting jamming in specific bandwidths and reroute the traffic to another one which is not being attacked. We can divide the process into two steps according to the Detection and the reAction phases, as although the latter cannot occur without the first, some KPIs are dependent on one of the phases only:

- Phase 1: Detection
    - Scenario: launch a jamming attack on a specific bandwidth which is inside the band being used by the cell, while leaving some of the frequencies of the band untouched. The USRP of the Detection block needs to differentiate which frequencies are compromised and which are usable, classifying them using the DL algorithm and communicating it via the interface with the reAction block.
    - Validation goals:
        - Jamming attacks detected (KPI2.1)
    - Metrics:
        - Detection rate
        - Other DL metrics (accuracy, f1 score…)
    - Expected outcome:
        - The detection rate of the correct attacked frequencies should be high, communicating them correctly to the reAction block.
- Phase 2: ReAction
    - Scenario: launch a jamming attack on a specific bandwidth which is inside the band being used by the cell, while leaving some of the frequencies of the band untouched. After the Detection block has identified which frequencies are being attacked, the reAction block should change the used ones to avoid the attack.
    - Validation goals:
        - Time needed to detect and prevent a jamming attack < 5s (KPI2.2)
        - Downtime prevented, less downtime at least 20% (KPI2.4)
        - Throughput enhancement during jamming attack of at least 40% (KPI 2.5)

- Successful establishment of connectivity to avoid jammed channels/paths (A-KPI2.6)

- Metrics:
  - Time needed to detect and prevent a jamming attack
  - Downtime
  - Throughput
- Expected outcome:
  - The reAction block should change the frequencies to avoid the ones being attacked enhancing the connectivity.

3.2.3. Sub-Use Case 2.3: Adaptive modulation techniques for anti- jamming autonomous recovery

3.2.3.1. Description

This use case focuses on recovery mechanisms, which have the capability to regain lost communication caused by jamming attacks without the need for human intervention. By incorporating AI-powered adaptive modulation specifically designed for dynamic jamming environments such as the ones the AVs operating in, machine learning-based channel estimation to enable robust modulation selection, and reinforcement learning-based modulation control, the objective is to enhance anti-jamming performance. Ultimately, this will lead to a more resilient communication system that can effectively withstand and recover from a variety of jamming attacks.

3.2.3.2. Architecture, Testbed and Setup

The architecture of UC2.3 is based on ISRD Liquid RAN and Liquid near-RT RIC which are both the proprietary implementations of the O-RAN Alliance Radio Access Network (RAN) and near - Real Time RAN Intelligent Controller (near-RT RIC). As such the architecture complies with the O-RAN Alliance standards.

The ISRD anti-jamming solution is developed as a near-RT RIC software application (xApp), termed Jamming Detection and Mitigation xApp (JDM-xApp).

This anti-jamming strategy enhances the traditional Adaptive Modulation and Coding (AMC) by incorporating multiple metrics beyond just CQI. It dynamically adjusts the MCS to a more robust setting based on physical layer metrics such as CSI, Reference Signal Received Power (RSRP) and CQI. On the MAC/Link-layer, we consider HARQ feedback with a focus on ACK/NACK patterns and BLER. For instance, if RSRP and CSI remain stable while HARQ NACKs and BLER increase significantly, the system can infer the presence of artificial interference rather than natural channel fading. In such cases, the scheduler proactively lowers the MCS level to strengthen transmission robustness, thereby reducing retransmissions and preventing UE disconnections.

The JDM-xApp relies on the proprietary ISRD KPM-xApp to receive the necessary RAN metrics such as CSI, Reference Signal Received Power (RSRP) and CQI as well as HARQ feedback and BLER.

The use case testbed is based on the available ISRD infrastructure. It consists of the following components:

- RAN network with anti-jamming solution (depicted in Figure 7):
  - UE: Equipment used to deploy the UE will be a USRP or commercial modules such as Huawei, Oppo etc.
  - O-RU: Equipment used to deploy O-RU will be a USRP or commercial module such as Benetel
  - O-DU and O-CU: Equipment used to deploy O-DU and O-CU will be a commercial server running proprietary ISRD Liquid RAN software
  - Near-RT RIC and xApps: It will be a commercial server running proprietary ISRD Liquid near-RT RIC, KPM-xApp and JDM-xApp software
- Jammer
  - Jammer will be deployed using USRP

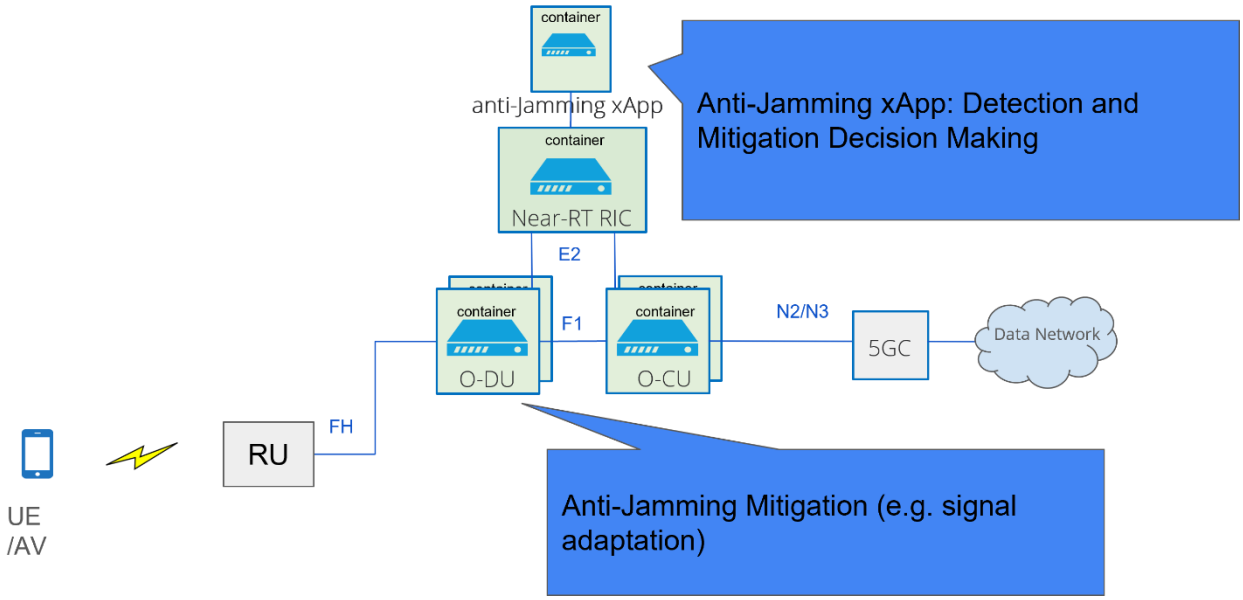

Figure 7: ISRD setup

3.2.3.3. Involved Services and Components

The involved services and components are the following

- Service: RAN
  - Component: Liquid RAN
    - This is the proprietary ISRD implementation of the O-RAN Alliance compliant RAN

- Service: near-RT RIC
  - Component: Liquid Near-RT RIC
    - This is the proprietary ISRD implementation of the O-RAN Alliance compliant near-RT RIC
  - Component: KPM xApp
    - This service provides key signal parameters such as CSI, CQI, RSRP, RSSI to the JDM-xApp
- Service: AI-based anti-jamming
  - Component: JDM-xApp
    - It continuously takes into account signal parameters from the KPM-xApp such as CSI, CQI, RSRP, RSSI as well as BLER to detect jamming in RAN and apply mitigation measures

Table 2: SubUC 2.3 services and components

Service	Component
AI-based anti-jamming	JDM-xApp
RAN	Liquid RAN
Near-RT RIC	Liquid Near-RT RIC
Near-RT RIC	KPM xApp

3.2.3.4. Validation Scenarios

In order to validate UC2.3 we propose the following scenario:

- Scenario: Launch jamming attack using USRP on a given cell. The JDM-xApp continuously monitors signal parameters and adjusts the traditional MCS algorithm to maintain connectivity under the jamming conditions.
- Validation goals:
  - Jamming attacks detected (KPI2.1)
  - Time needed to detect and prevent a jamming attack < 5s (KPI2.2)
  - Downtime prevented, less downtime at least 20% (KPI2.4)
  - Throughput enhancement during jamming attack of at least 40% (KPI 2.5)
- Metrics:
  - Detection rate
  - Detection time
  - Mitigation rate
  - Throughput under jamming
- Expected outcome:
  - Jamming attack is detected, JDM-xApp adapts the original MCS algorithm to maintain the connectivity. The connection to the UE is not dropped.

3.2.4. Sub-Use Case 2.4: Improving 6G security in 6G spectrum

3.2.4.1. Description

This use case focuses on safeguarding 6G spectrum bands, particularly those in the sub-THz range, by leveraging AI-driven PKG techniques that rely on channel reciprocity. These techniques utilize the unique characteristics of the wireless channel to generate secure keys, ensuring robust encryption that is inherently resistant to interception. AI enhances the PKG process by optimizing the generation of secure keys, taking full advantage of the unique and dynamic channel properties between devices. This approach ensures a higher level of security and protection for communications within the sub-THz frequency bands, strengthening the overall security framework of 6G networks.

3.2.4.2. Architecture, Testbed and Setup

The architecture design of UC 2.4 addresses the challenge of implementing an AI-enhanced PKG system tailored for 6G security in the sub-THz spectrum. The process begins by collecting channel metrics from the wireless channel using specialized sub-THz hardware (USRPs). These raw metrics are pre-processed and then fed into a pre-trained AI module designed to exploit the inherent reciprocity and randomness of the channel. The AI module processes the data to generate a symmetric key for both communicating entities, typically referred to as Alice and Bob, ensuring that both keys match while maximizing the Key Generation Rate (A-KPI 2.9).

Security validation is an integral part of the testbed setup. The system is designed to counteract eavesdropping attempts by an adversary (Eve) through a series of rigorous tests. These tests include applying standardized randomness evaluations, such as the NIST test suite, ensuring that the generated keys exhibit the desired level of randomness and are resistant to interception.

To achieve this, we propose a laboratory-based architecture where USRPs are integrated with nodes to enable physical key generation for 6G security. In this system, a dedicated acquisition block connected to the USRPs captures IQ samples from the wireless channel. These raw measurements are then forwarded to a preprocessing module, which normalizes the data and extracts essential channel features. The processed channel characteristics are subsequently transmitted to an AI inference module that exploits the inherent reciprocity and randomness of the channel to generate a symmetric key shared between both nodes. The output of the AI inference is delivered to a reconciliation block, where any discrepancies between the generated keys are corrected. Critical to the system's effectiveness is a robust synchronization mechanism that ensures both nodes perform channel measurements simultaneously, thereby preserving channel reciprocity.

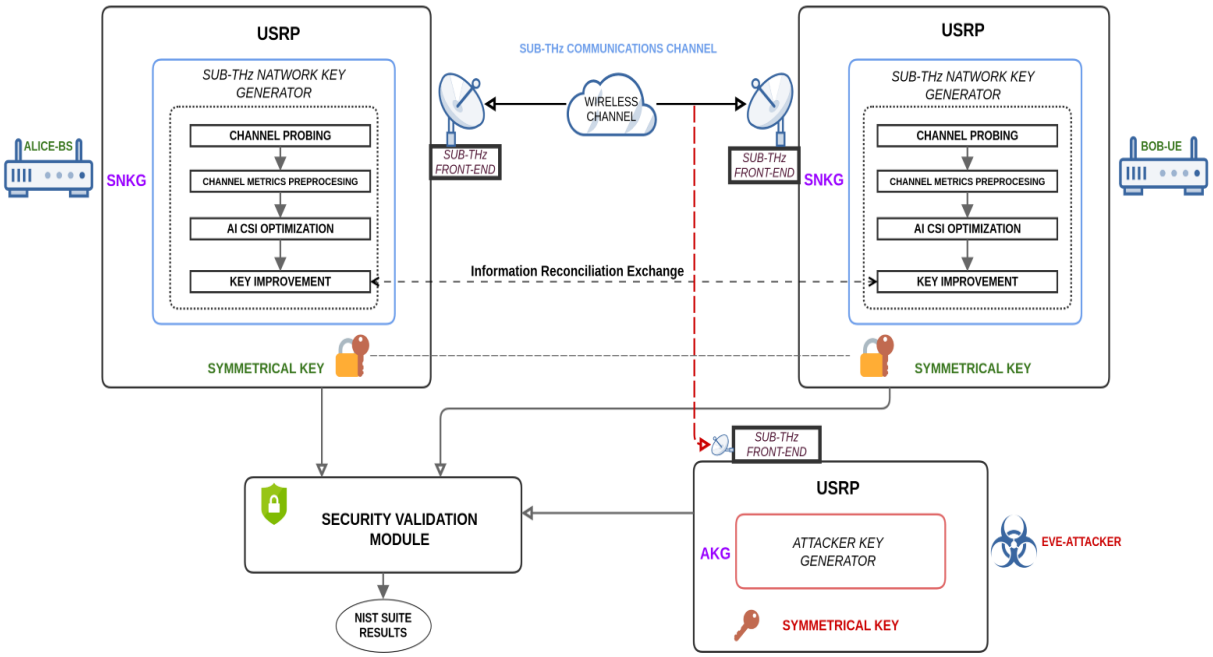

Figure 8: GRAD setup

- **Transmission and Reception Devices (Alice, Bob & Eve):** In our 5GLab GRADIANT testbed, these endpoints are implemented using software-defined radios connected via a shielded, wired RF matrix. This setup ensures an isolated and deterministic environment where both endpoints are calibrated and synchronized to capture high-fidelity IQ samples and detailed channel parameters. Such precision is critical for accurately measuring the sub-THz channel characteristics required for secure key generation. On the other hand, Eve will be equipped with similar RF hardware, introduced into the testbed to simulate eavesdropping attacks. Eve attempts to intercept or reconstruct the generated key, providing a critical measure of the PKG system's robustness. Evaluating the system's performance in the presence of Eve ensures that the key generation process remains secure against sophisticated interception attempts.
- **Sub-THz Wireless Channel:** This will be emulated using specific hardware equipment to simulate the unique characteristics of the sub-THz frequency band.
- **Sub-THz NATWORK Key Generation Module:**
    - **Preprocessing:** At each endpoint, dedicated preprocessing modules filter, normalize, and extract essential channel features from the raw IQ data. This conditioning step ensures that the AI-driven key generation process receives consistent, high-quality input data, thereby reducing noise and mitigating the effects of hardware imperfections.

- **AI CSI Optimization:** In this module a neural network model is trained to predict the downlink channel based on uplink measurements, ensuring effective channel reciprocity even in FDD scenarios. Leveraging machine learning techniques, the model adapts to sub-THz channel conditions to maximize the Key Generation Rate and minimize the Key Disagreement Rate, effectively exploiting the inherent randomness of the sub-THz channel.
- **Key Improvement Module:** The quantificator is in charge of generating the first keys generated through AI output, after that the reconciliation module compares and aligns the keys generated at both endpoints by implementing robust error-correction algorithms. This process corrects discrepancies caused by noise or slight measurement variations, ensuring that both Alice and Bob ultimately derive an identical symmetric key.

- **Security Validation Module:** The final output of the PKG process is a symmetric key used for encrypting communications (via AES-128). This key is generated to meet stringent length and randomness requirements, validated through industry-standard tests such as those outlined by the NIST suite [6]. The Security Validation Module conducts a comprehensive suite of tests, including the NIST randomness evaluations, to assess the key's randomness, length, and overall robustness. It verifies that the generated key complies with the predefined security KPIs and produces detailed validation reports, thereby confirming that the PKG mechanism remains secure even in adversarial conditions.

3.2.4.3. Involved Services and Components

The different services and the associated components with each service for the U.C 2.4 are presented below:

- Characteristics Extraction Service: This service tries to replicate the behaviour of the channel for sub-THz frequency range in our test environment. The components associated with this service are the RF Hardware Components used to emulate the sub-THz conditions.
- Key Generation Service: This service is responsible for deriving a symmetric encryption key from physical channel measurements. It is associated with the AI module component for analysing the channel features and the quantification and reconciliation components to ensure the keys generated by both ends.
- Security Validation Service: The service performs standard security assessments to verify the robustness and randomness of the generated key, all this associated with the Security Validation Module.

3.2.4.4. Validation Scenarios

The validation of use case 2.4 focuses on evaluating how keys generated at the physical layer can reduce the risk of unauthorized listening, which is basically eavesdropping prevention for wireless communications between two nodes. An external attacker attempts to extract or recreate the encryption key. This scenario provides crucial insight into the overall security level that the PKG mechanism can offer under realistic threats.

After an initial setup phase in which Alice and Bob establish a wireless link, attacker Eve is introduced to monitor the exchange. In this case we will evaluate different scenarios and phases:

- **Phase 1: Secure Key Establishment and Communication**
  - Scenario: Deploy two PKG-enabled nodes (transmitter and receiver) and a third node acting as an eavesdropper. Generate a shared key (128 bits) from physical channel measurements and use it to encrypt communications (AES128). Simultaneously, launch an eavesdropping attempt against the encrypted traffic.
  - Validation Goals: Confirm that the PKG-generated key establishes an encrypted channel that significantly reduces the risk of unauthorized interception.
  - Metrics: Eavesdropper success Rate.
  - Expected Outcome: Eavesdropper's success rate being low, thereby demonstrating robust encryption.
- **Phase 2: Key Integrity, Length, and Randomness Verification**
  - Scenario: Simulate a wireless link between the PKG-enabled nodes. Generate keys from channel measurements under various conditions. Evaluate both the key length and its randomness using the NIST test suite in this last case.
  - Validation Goals: Ensure that the key generation process consistently produces a key of exactly 128 bits (A-KPI 2.7) and verify that the generated keys comply with NIST randomness criteria with p-values greater than 0.01 (A-KPI 2.8).
  - Metrics: Key Length and NIST statistical values.
  - Expected Outcome: The PKG mechanism should reliably produce robust keys that are exactly 128 bits long and meet NIST randomness requirements.
- **Phase 3: Key Disagreement, Generation Rate, and Downtime Prevention Analysis**
  - Scenario: Under varying channel conditions, including tests using both FDD (Frequency Division Duplex) and TDD (Time Division Duplex), the transmitter and receiver independently generate keys using the PKG mechanism. During these tests, AI algorithms are applied to optimize the generation keys process, aiming to reduce the Key Disagreement Rate (KDR) and boost the Key Generation Rate (KGR). The process is continuously monitored for discrepancies between the keys and for any downtime or rekeying delays.

- Validation Goals: Reduce KDR comparing FDD and TDD behaviors, boost KGR and validate that the optimized key generation contributes to reduce the downtime (fewer delays during key generation).
- Metrics: KDR, KGR (A-KPI 2.9) and Downtime Prevented (KPI 2.4).
- Expected Outcome: The PKG mechanism, enhanced with AI algorithms, should reliably produce keys in both FDD and TDD configurations with minimal discrepancies between nodes. Improved channel conditions by AI optimizations are expected to increase the KGR and reduce KDR.

3.3. Use Case 3: IoT Security

The large-scale deployment of IoT devices in 6G networks introduces significant security challenges, such as DDoS attacks, data breaches, and unauthorized access. To address these risks, this use case focuses on developing AI-based intrusion detection systems (IDS) and penetration testing tools to enhance IoT security. Machine learning algorithms, including Deep Neural Networks (DNN), will be used for real-time anomaly detection by analysing network traffic patterns, while reinforcement learning will optimize dynamic threshold settings. AI-driven penetration testing and vulnerability assessments will identify security weaknesses, with NLP models generating targeted social engineering attacks to test defences. Additionally, blockchain technology will be leveraged for decentralized trust management and end-to-end protection. The combination of AI, blockchain, and advanced analytics will provide a comprehensive security framework for safeguarding IoT deployments. Collaboration among MONT, CERTH, ELTE, and ZHAW will ensure the development of robust security mechanisms for 6G IoT ecosystems.

3.3.1. Sub-Use Case 3.1: Enabling anomaly detection using machine learning automated techniques for attack detection

3.3.1.1. Description

The rapid proliferation of Internet of Things (IoT) devices, combined with the emergence of 6G networks, presents both transformative opportunities and critical security challenges. On one hand, this hyper-connected ecosystem enables unprecedented levels of connectivity, automation, and data-driven innovation. On the other hand, it significantly broadens the attack surface, exposing IoT environments to increasingly complex and frequent cyber threats. Sub-Use Case 3.1 is positioned within this context, aiming to leverage advanced Machine Learning (ML) techniques to improve the detection and mitigation of Distributed Denial-of-Service (DDoS) attacks. The fundamental rationale is that traditional security approaches are no longer sufficient to keep pace with the evolving threat landscape. Instead, an intelligent and adaptive system is required—one capable of responding dynamically to emerging attack vectors.

Sub-Use Case 3.1 addresses several critical challenges inherent in securing IoT networks within 6G infrastructures. One of the foremost challenges is scalability. With billions of devices generating vast amounts of data, conventional security frameworks often struggle to maintain performance while scaling. Therefore, the proposed solution must be capable of monitoring and securing extensive, distributed networks without introducing significant latency or overhead.

Another key challenge is real-time detection and response. DDoS attacks can cripple networks within seconds, necessitating immediate threat identification and counteraction. This requires systems that can efficiently process high-volume network data and execute mitigation strategies without delay.

Accuracy in anomaly detection is also vital. Differentiating between normal but unusual behavior (such as network congestion) and actual malicious activity (like a coordinated DDoS attack) can be complex. High precision in detection minimizes false positives—which waste resources—and false negatives—which leave networks exposed to undetected threats.

Moreover, the security approach must be adaptive. Attack methods evolve constantly, rendering static or rule-based systems obsolete. Sub-Use Case 3.1 therefore emphasizes the importance of learning-based models that adapt based on observed patterns and past incidents, maintaining resilience in the face of novel threats.

Finally, seamless integration with existing infrastructures is a core requirement. Many IoT deployments rely on legacy systems, so any new security solution must work alongside current architectures with minimal disruption. The ability to enhance protection without necessitating full-scale system overhauls ensures practical deployment and long-term sustainability.

3.3.1.2. Architecture, Testbed and Setup

This section elaborates on the various components of the testbed configuration, including network topology, security measures, and monitoring and management tools. The testbed will be developed and supported by PNET, CERTH, and MONT. The setup illustrated in Figure 9 integrates key components for monitoring, analyzing, and securing IoT traffic within a virtualized 5G/6G testbed. On the left, a set of phones and IoT devices connect through a base station (eNodeB/gNodeB), which is managed by the OpenAirInterface (OAI) software. This OAI component serves as the interface between radio access and the core network. The Virtualized Evolved Packet Core (vEPC) hosts the main control plane functions: the AMF (Access and Mobility Management Function), SMF (Session Management Function), UPF (User Plane Function), and the HSS (Home Subscriber Server), interconnected using standard Sx, N4, and S1 interfaces.

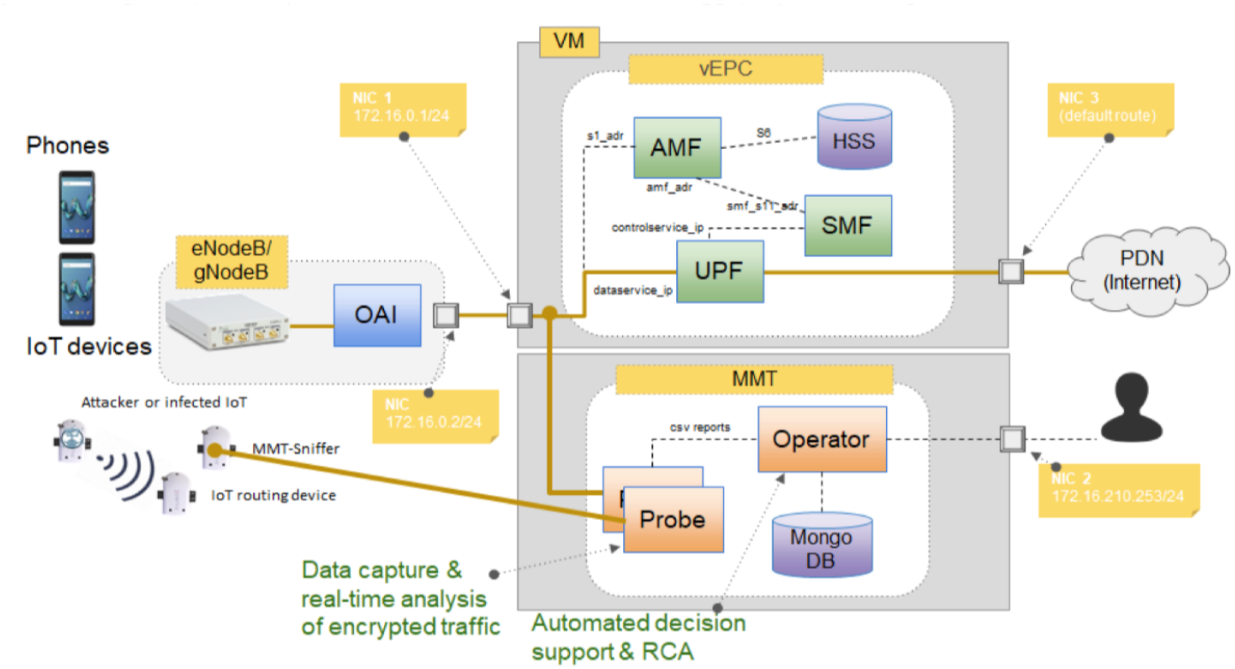

Figure 9: Sub-Use Case 3.1 setup

Below the vEPC, the Montimage Monitoring Tool (MMT) is deployed to enable real-time traffic monitoring and security analysis. It includes three core components: the MMT Probe, which captures and inspects traffic data (including encrypted data); the Operator, which interprets and acts on insights extracted by the probe; and a MongoDB instance that stores structured monitoring data and analysis results. An MMT-Sniffer is connected to an IoT routing device to intercept traffic—including that from potentially compromised or attacker-controlled IoT nodes.

The captured traffic is processed for anomaly detection and Root Cause Analysis (RCA), enabling automated alerts and decisions. The system also supports real-time reporting (e.g., via CSV) and interfaces with external operators or dashboards for visualization and management. The network is connected to the public data network (PDN) via NIC 3, while internal communication between the OAI, vEPC, and MMT occurs over interfaces NIC 1 and NIC 2.

- **IoT / Wireless Sensors Network**

The testbed will include simulated and real IoT devices, sensors, and gateways. This setup is essential for emulating the diverse and complex conditions found in actual IoT deployments. Devices will be configured to generate traffic patterns that represent both normal operations and potential attack scenarios, such as Distributed Denial of Service (DDoS) attacks. The simulation environment will enable controlled testing of machine learning (ML) algorithms designed to detect and mitigate these threats in real-time.

The testbed will incorporate a multi-tier architecture, consisting of edge, fog, and cloud layers: The edge layer will include IoT gateways and edge devices that process data close to the source, minimizing latency and enabling real-time decision-making. The fog layer acts as an intermediary, providing additional processing power and storage closer to the edge, but with more computational resources than the edge layer. The cloud layer will be used for more extensive data processing, storage, and centralized management of the network. This tiered approach will allow for the evaluation of the effectiveness and efficiency of security mechanisms deployed at different levels of the network, particularly in scenarios where computational resources and network conditions vary.

- **Security Measures**

The testbed will include secure, isolated environments specifically designed for testing scenarios that involve sensitive data or potentially untrusted infrastructure providers. These environments will be segmented from the rest of the network to prevent unauthorized access and to contain any potential security incidents. This isolation is particularly important when testing security measures that involve processing confidential information or when evaluating the resilience of the system against insider threats.

In addition, the testbed will be equipped with tools and configurations necessary for conducting penetration testing and vulnerability assessments. These tools will be used to simulate attacks on the network and identify potential weaknesses in security mechanisms. The penetration testing setup will include automated testing tools as well as manual testing procedures to ensure a comprehensive assessment of the security posture of the IoT network.

- **Monitoring and Management**

Effective monitoring and management are essential for maintaining the testbed's performance and integrity and ensuring accurate and reliable testing outcomes. A centralized dashboard (e.g., MMT-Operator) will be implemented to provide real-time monitoring of the testbed’s performance. This dashboard will offer a unified interface for managing test scenarios, tracking key performance indicators (KPIs), and visualizing the results of security tests. The dashboard will enable users to monitor network traffic, detect anomalies, and observe the behavior of ML-based intrusion detection systems in real-time. It will also facilitate the management of the testbed infrastructure, allowing for easy deployment and scaling of test scenarios.

Furthermore, the testbed will incorporate automation frameworks to streamline the deployment and scaling of tests, the collection of results, and the resetting of the environment between tests. Automation will be critical for efficiently managing the complex and repetitive tasks involved in testing multiple scenarios and configurations. These tools will enable the rapid iteration of test

scenarios, ensuring that all relevant use cases are thoroughly evaluated within a consistent and controlled environment.

3.3.1.3. Involved Services and Components

Table 3 lists all the components involved in the validation of Sub-Use Case 3.1.

Table 3: SubUC 3.1 services and components

Service	Component
Security Monitoring	AI-driven security monitoring for anomaly detection and root cause analysis in IoT networks

3.3.1.4. Validation Scenarios

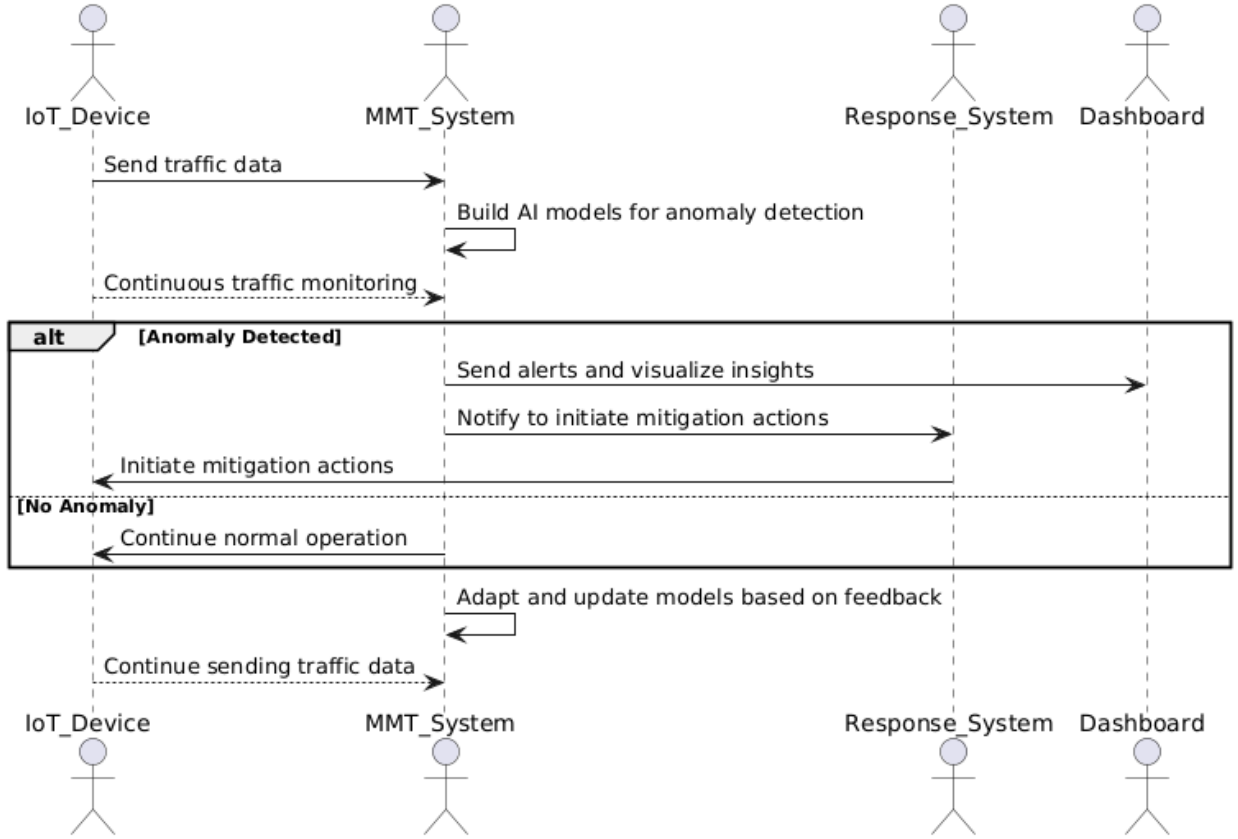

Figure 10: Workflow of anomaly detection system

The provided sequence diagram (Figure 10) illustrates the workflow of the anomaly detection system which will be deployed in Sub-Use Case 3.1. Here's a breakdown of the process:

- IoT devices continuously send traffic data to the MMT (Montimage Monitoring Tool) system. The MMT system receives this data in real-time for further analysis.
- The MMT system uses the collected traffic data to build AI models specifically designed for detecting anomalies. These models are tuned to identify unusual patterns in the

network traffic that could indicate potential threats such as Distributed Denial-of-Service (DDoS) attacks or intrusions.

- IoT devices continue to send traffic data, and the MMT system performs real-time monitoring, using AI models to analyze this data continuously.
- If an anomaly is detected, the system takes the following actions:
- The MMT system sends alerts to a centralized Dashboard to notify operators of potential threats. The system also visualizes the relevant insights and data, allowing the security team to assess the severity of the detected anomaly.
- Simultaneously, the MMT system triggers the Response System to initiate mitigation actions. The Response System takes necessary steps, such as blocking malicious traffic or isolating compromised devices, to minimize the impact of the detected anomaly.
- If no anomaly is detected, the system continues its normal operation, ensuring uninterrupted communication between IoT devices and the MMT system.
- The MMT system continuously adapts and updates its AI models based on feedback from the system, improving its detection accuracy. This feedback loop ensures that the models evolve as the network conditions or threats change, enhancing the system's ability to identify future anomalies.
- IoT devices continue to send traffic data to the MMT system, maintaining the system's vigilance and ensuring that traffic is continuously monitored for any new potential threats.

3.3.2. Sub-Use Case 3.2: Validating AI-driven penetration testing and vulnerability assessment for attack mitigation

3.3.2.1. Description

Use Case 3.2 focuses on leveraging artificial intelligence to develop a sophisticated penetration testing tool that evaluates the security of 6G network infrastructures. This tool simulates advanced cyber threats by integrating AI-driven phishing and Denial of Service (DoS) attack scenarios. The objective is to assess the vulnerabilities of human-operated systems and the overall resilience of network services under adversarial conditions.

In this use case, large language models (LLMs) are utilized to craft persuasive phishing emails targeted specifically at 6G network administrators and operators. These emails are designed to manipulate human behaviour and encourage recipients to interact with malicious content. Each email includes an attachment embedded with a payload that triggers a DoS attack upon execution.

The DoS attack is orchestrated by a reinforcement learning-based AI system, which dynamically adjusts its attack strategy to maximize disruption. This AI-powered technique enables continuous adaptation and optimization, simulating real-world adversarial behaviour. The goal is to degrade the Quality of Service (QoS) and measure the impact on network performance, including delays, outages, and the overall user experience.

Through these simulations, the system evaluates the ability of 6G networks to detect, withstand, and recover from sustained AI-based cyberattacks. It also highlights how LLMs can be exploited to bypass human defences and how AI can be weaponized to disrupt critical infrastructure.

Use Case 3.2 enhances the NATWORK project by introducing an AI-powered attack generation engine, i.e. a penetration testing tool that simulates advanced attack scenarios beyond traditional approaches. By combining Denial of Service (DoS) attacks with protocol-level fuzzing, it will generate custom network packets to uncover vulnerabilities in 5G services that conventional tools may miss. This enables a deeper evaluation of network resilience and protocol security, ultimately strengthening 5G and 6G infrastructures. By modelling realistic, AI-enabled attacks, this use case provides valuable insights into next-gen network security, helping to build more resilient and intelligent defences.

By mimicking real-world threat scenarios, this use case aims to provide valuable insights into the evolving landscape of AI-enabled cyber threats. It supports the development of more robust security measures and defence mechanisms, ultimately contributing to the design of more resilient 6G communication environments.

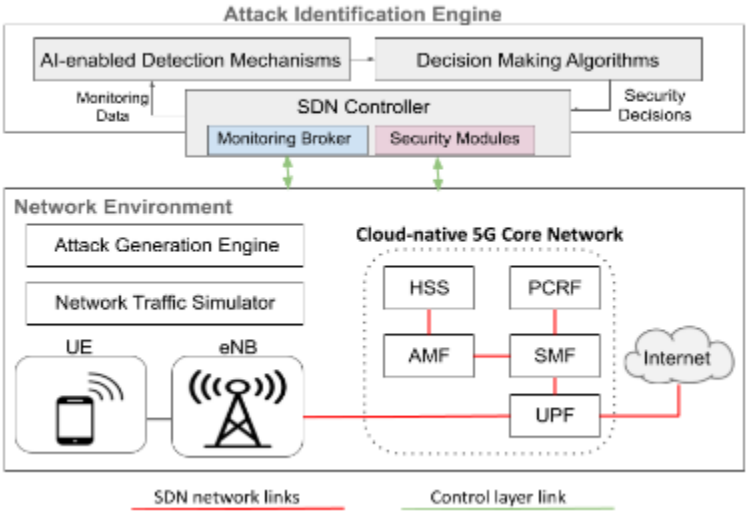

Figure 11: High level overview of the proposed system

3.3.2.2. Architecture, Testbed and Setup

The CERTH testbed experimentation environment is composed of two primary functional components: the Attack Identification Engine, which serves as the central control unit, and the Network Environment, as depicted in Figure 11. The Attack Identification Engine is responsible for: (i) configuring the network appropriately; (ii) monitoring traffic and detecting attacks; and (iii) implementing potential attack mitigation measures i.e. the main components of UC4.3 and UC4.4. The Network Environment includes containerized 5G network elements, User Equipment (UE), and the Attack Generation Engine examined in UC3.2, resides in the Network environment.

The Cloud-native 5G Core Network, which forms the 5G experimentation environment, comprises containerized 5G Network Functions (NFs) that facilitate communication for 5G UEs. It encompasses the essential NFs of a 5G core network. Specifically, after deployment, nine containers are instantiated—two of which run the User Equipment (UE) and Evolved Node B (eNB), while the remaining containers, based on the Free5GC project, host the 5G Core Network Functions. All components are interconnected through Software Defined Networking (SDN) virtual switches. Figure 12 illustrates the 5G network topology as visualized through the Floodlight controller's graphical interface.

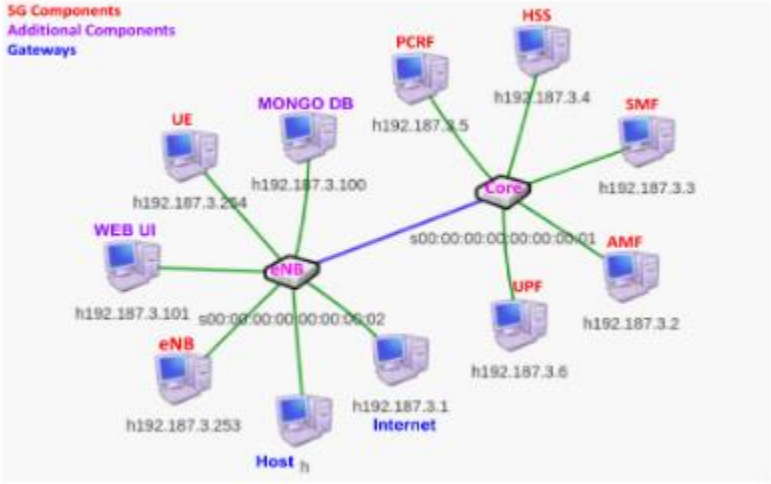

Figure 12: High level overview of the CERTH testbed.

A crucial element of this setup is the SDN Controller, which manages communication between all network functions. The controller comprises two subcomponents: (i) the Monitoring Broker, responsible for collecting real-time network state information to maintain a comprehensive global network view; and (ii) the Security Modules, which define and enforce traffic flow rules, capable of acting swiftly at any network node. Network statistics—such as bandwidth consumption, flow durations, and the number of active flows—are sourced from SDN devices.

Additionally, as the NFs are implemented using Docker containers, compute resource metrics are periodically gathered via the docker stats utility.

From an implementation perspective, the SDN Controller utilizes the **Floodlight** platform, SDN switches are based on **OpenvSwitch** software, and communication within the control plane is facilitated by the **OpenFlow** protocol.

3.3.2.3. Involved Services and Components

Table 4 lists all the components involved in the validation of Sub-Use Case 3.2.

Table 4: SubUC 3.2 services and components

Service	Component
AI driven penetration Testing	AI-enabled DoS attack

3.3.2.4. Validation Scenarios

The validation scenario is structured in three sequential steps, each demonstrating a key capability of the AI-based DoS system in disrupting 5G/6G network communication.

Step 1: Attack Launch and Feedback Loop

The AI initiates a DoS attack targeting the 5G/6G Core (requirement S8-S-C3- AI-enabled DoS attack, as defined in D2.3). As the attack progresses, the system continuously monitors Quality of Service (QoS) metrics (A-KPI 3.6, A-KPI 3.7), receiving feedback from the core to evaluate the impact.

Step 2: Adaptive Optimization

Using the feedback, the AI refines its strategy in real time, adjusting attack parameters to maximize network disruption.

Step 3: Communication Breakdown and Service Denial

The optimized attack disrupts normal communication between the 5G/6G Core and gNodeB. This breakdown halts data exchange with the User Equipment (UE), resulting in a complete denial of service.

This streamlined process highlights the AI's ability to autonomously execute and adapt sophisticated attacks, providing insights into the network's resilience and recovery capabilities.

The validation scenario of UC3.2 is also depicted in Figure 13. The entire process will be presented in automatically created vulnerability report regarding DoS resilience on 5G/6G components (A-KPI 3.8).

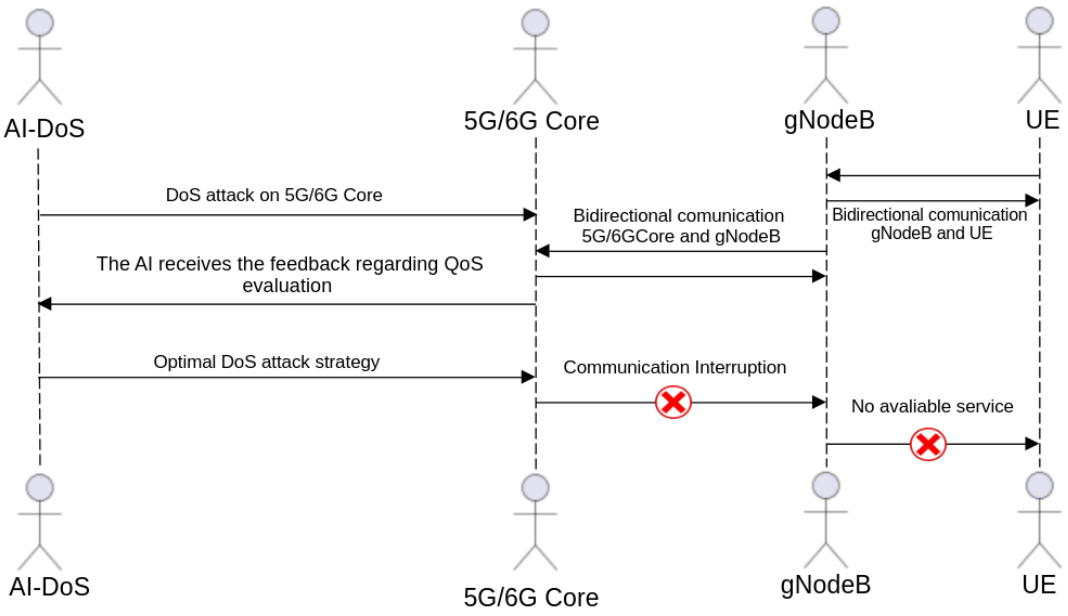

Figure 13: Sequence Diagram for use case UC3.2 depicting the various steps of the validation scenario

3.3.3. Sub-Use Case 3.3: Enhancing blockchain-based security and trust management end-to-end security

3.3.3.1. Description

Use Case 3.3 focuses on establishing a trusted, secure communication between IoT nodes and service providers within a 6G-enabled IoT environment. In highly distributed and dynamic network scenarios, traditional centralized models are often limited in scalability and resilience. This use case addresses those limitations by adopting a decentralized approach to trust management, using blockchain technology to support secure registration, authentication, and access control across the network. The goal is to enable end-to-end trust between the IoT device and the IoT service provider, ensuring strong data protection and system integrity, even across distributed network domains such as edge and cloud. Instead of relying on a centralized authority, the trust framework is supported by blockchain-based storage of key public information, which enhances the transparency, reliability, and immutability of authentication data.

3.3.3.2. Architecture, Testbed and Setup

Key security functions implemented in this use case include:

- IoT devices and services are authenticated using a blockchain-backed framework, which includes storing partial public information from key 5G core components (AMF, AUSF, and

UDM) on the blockchain. This ensures that both the device and the service provider can securely verify each other without depending on a centralized trust anchor.

- Cryptographic mechanisms are applied to verify that the data exchanged between IoT devices and services is authentic and untampered. This is especially important in IoT scenarios where sensitive or critical data is transmitted across various network layers.
- Access policies are enforced to ensure that only authorized users or devices are allowed to access specific services or perform certain actions. This strengthens security while allowing flexible and scalable service management.
- The system design emphasizes protecting user and device data against potential surveillance and data leaks, using secure encryption and decentralized data verification mechanisms.

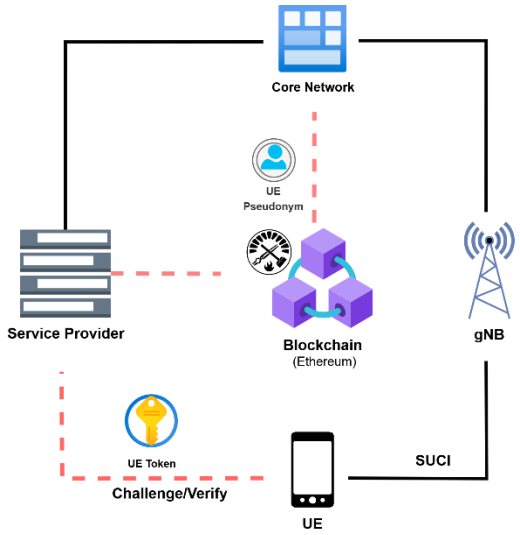

Figure 14: Main components of UC#3.3

By combining blockchain with advanced security protocols, Use Case 3.3 showcases a robust solution for trusted IoT service delivery in 6G environments. Figure 14 shows the relations between the main players. The testbed for Use Case 3.3 is built using both virtualized and physical components to represent a realistic 5G-enabled IoT deployment. The setup includes the following five key components:

- 5G Core: Implemented with Open5GS and provides central control network functions such as AMF, AUSF, and UDM. These NFs handle authentication and mobility management for the IoT devices.
- UPF and DN: Acts as the User Plane Function utilizing Open5GS and connects to the Data Network (DN) utilizing HTTPS server, enabling data routing from the IoT device to the service provider.
- UE: Emulates the User Equipment (IoT device) functionality. In the physical testbed, a Raspberry Pi with UERANSIM is used to represent an actual IoT node.

- gNB: Represents the RAN node to simulate radio access using UERANSIM and establish communication between the UE and 5G Core.
- Blockchain: Represents the distributed ledger, implemented by Foundry, that will be utilized using smart contracts to perform parts of the end-to-end trust establishment process.

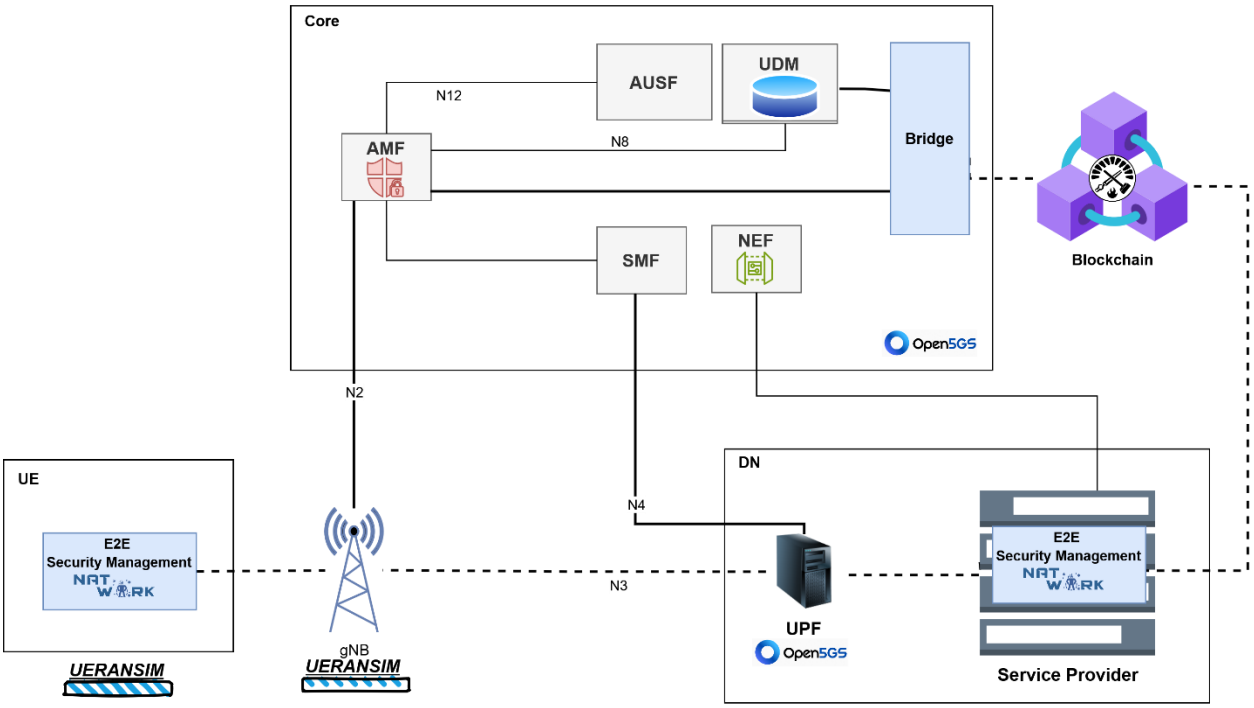

Figure 15: Use case 3.3 workflow and testbed components

The nodes shown in Figure 15 are deployed using VM, utilizing open5GS, UERANSIM, Foundry and HTTPS server allowing flexible configuration and testing. The physical Raspberry Pi nodes serve as the edge devices and are integrated into the testbed to simulate a real-world IoT deployment. This hybrid setup allows for accurate evaluation of network behavior and performance under different configurations. A central component of the architecture is the blockchain layer. During the initial registration process, partial public information from AMF, AUSF, and UDM is stored on the blockchain. This allows any participating node or service provider to verify the legitimacy of devices and services without contacting a central authority, thereby reducing authentication overhead and improving trust decentralization.

3.3.3.3. Involved Services and Components

The implementation of Use Case 3.3 includes a set of key components that collectively ensure secure, decentralized, and privacy-preserving communication in a 6G IoT network. The main involved service is E2E Security Management (S3-S-C2) in Security by Design Orchestration service, which includes the following main modules:

- *Decentralized Authentication (Blockchain):* Manages secure, blockchain-backed user and device authentication. Utilizes stored and secured identity fragments from 5G core components to enable trust validation between IoT nodes and service providers. It replaces the authorization database with an Ethereum-compatible permissioned blockchain. This provides a decentralized, transparent, and integrity-safeguarded mechanism for device authentication management.
- *Data Checker (Bridge):* It plays an essential role as a communication link between the 5G core network and the blockchain. Its main purpose is to monitor the AMF function within the 5G core, identify the pseudonym used during registration, and update the blockchain with relevant security information.

3.3.3.4. Validation Scenarios

The validation is structured into four key phases. Each phase focuses on testing a critical security or trust-related component of the use case.

Phase 1: Trust Generation

A newly deployed IoT device (e.g., Raspberry Pi) is powered on for the first time and initiates the network registration process via the gNB and Open5GS 5G Core. During this process, the AMF and AUSF perform identity verification. A pseudonym (i.e., a privacy-preserving identifier) is generated for the device, and trust-related information is securely stored on the blockchain. This step enables decentralized trust establishment by allowing future validation without repeated identity disclosures.

Validation goals:

- To validate the decentralized trust architecture based on blockchain, including pseudonym generation and secure recording of trust data.
- To ensure that the entire process meets the required performance expectations for trust establishment latency.

Metrics:

- Confirms that the blockchain-based trust model supports pseudonym-based identity abstraction (REQ-3.3-1).
- Measures time taken from device boot to trust metadata being successfully recorded in the blockchain within acceptable latency A-KPI-3.12.

Expected outcome:

- A unique pseudonym is generated and linked to the device's verified identity.
- Trust metadata is written securely to the blockchain.
- The 5G Core completes its involvement after this registration phase.

Phase 2: Trust Verification

Following blockchain registration, the IoT service provider receives a connection request from the device. It queries the blockchain ledger using the pseudonym to verify the trust status of the device. Upon successful verification, a secure communication channel is established directly between the device and the service provider. This communication does not require further interaction with the 5G Core.

Validation goals:

- To validate that the service provider can independently verify the device's trust using the blockchain.
- To confirm that secure communication can proceed without further dependency on the 5G Core.

Metrics:

- Validates the ability of third-party service providers to perform trust checks using blockchain data (REQ-3.3-1).
- Confirms the trust verification and communication setup time remains within defined latency limit (A-KPI-3.12)

Expected outcome:

- The service provider retrieves and validates the device's trust record from the blockchain using the pseudonym.
- Encrypted communication is initiated between the IoT device and the service.
- No additional involvement from the 5G Core is required after trust has been verified.

Phase 3: Unauthorized Access Attempt

In this phase we test the system's ability to detect and block unauthorized entities. An attacker attempts to impersonate a legitimate IoT device by using outdated or randomly generated credentials. The attacker tries to connect to the service provider without proper registration or blockchain record. The service provider consults the blockchain and, finding no valid trust anchor, rejects the request.

Validation goals:

- To assess the system's ability to detect and block unauthorized access attempts.
- To validate that access is permitted only for devices with verified trust records in the blockchain.

Metrics:

- Confirms enforcement of access control based on blockchain trust validation (REQ-3.3-2).
- Measures the speed and accuracy of detecting and rejecting unauthorized access (A-KPI-3.9).

Expected outcome:

- The attacker is denied access immediately.
- No communication is established.
- System performance and availability for legitimate devices are unaffected.

Phase 4: Trust Violation and Further Attacks

This phase simulates advanced threats such as man-in-the-middle (MITM) attacks, impersonation attempts, and data tampering. A proxy device attempts to intercept or modify messages between a valid IoT device and the service provider. The system is tested for its ability to detect and respond to these threats while allowing legitimate devices to function normally.

Validation goals:

- To ensure secure, privacy-preserving aggregation of trust and security data.
- To evaluate detection of real-time trust violations, including data tampering and impersonation.
- To maintain service continuity for verified devices under attack conditions.

Metrics:

- Confirms integrity and privacy of trust/security data under attack (REQ-3.3-3).
- Measures detection capability for tampering in communication (A-KPI-3.10).
- Measures accuracy in identifying impersonation and maintaining continuity for trusted operations (A-KPI-3.11).

Expected outcome:

- Privacy-preserving aggregation of trust and security data is assured.
- MITM and impersonation attacks are detected and blocked.
- Trust violations are responded to appropriately.
- Legitimate communications remain uninterrupted.

Expected outcomes of the phases are a fully functional end-to-end trust establishment mechanism using blockchain and decentralized security. Additionally, demonstrated ability to block unauthorized access and identify integrity violations.

3.4. Use Case 4: Improving variability of network with continuous security

The 6G network architecture will be highly dynamic and heterogeneous, requiring continuous security monitoring to address challenges posed by diverse devices, mobile payloads (e.g., drones, vehicles), and evolving threats. Machine learning (ML) and AI will play a crucial role in real-time security adaptation, threat prediction, and dynamic defense mechanisms. This use case focuses on showing how network variability can be improved while ensuring security through AI-driven strategies, including Moving Target Defense (MTD), software-defined radio (SDR) for agile payload communication, AI-assisted network slicing for efficient resource allocation, and DoS attack detection by payload self-monitoring. Techniques like deep reinforcement learning and federated learning will optimize resource management, detect anomalies, and enhance resilience against emerging threats such as DoS attacks and zero-day exploits.

The use case further explores AI-driven microservices orchestration to maintain QoS during undetectable attacks, using ML to profile normal behavior, detect anomalies, and classify risks. Additionally, it investigates explainable AI for optimizing MTD in the 6G edge-to-cloud continuum, balancing security gains with operational overhead. SDR will enable adaptive payload communication by predicting channel conditions and dynamically allocating resources. By integrating threat intelligence, infrastructure monitoring, and vulnerability assessments, the use case partners, MONT, CERTH, ZHAW, TSS, ELTE and CNIT, aim to create a scalable, secure 6G network capable of autonomous adaptation while providing transparency to security experts through explainable AI techniques.

3.4.1. Sub-Use Case 4.1: Enabling software-defined networking and network function virtualisation by employing security aware dynamic resource allocation and monitoring

3.4.1.1. Description

The combination of Decentralized Feature Extraction (DFE) and Wirespeed AI Offloading (WAI) presents a novel approach to dynamically adapting the behavior of heterogeneous data plane devices, such as switches and smart NICs. This approach enhances security by enabling real-time offloading of computational tasks related to attack detection and mitigation. The key objective is to demonstrate the potential of security computation offloading as a service, where network functions can be dynamically instantiated, monitored, and reconfigured based on evolving security threats. WAI and DFE are orchestrated by a dedicated Security Orchestrator, which ensures that security functions are deployed and optimized across the network infrastructure. While WAI/DFE mechanisms provide protection at the data plane layer, they may not be capable of identifying previously unknown attacks. To address this limitation, selected telemetry features

from the DFE process are transmitted to AI-based collectors, which analyze the data for new attack patterns. Once a novel attack is identified and profiled, the system dynamically enforces updated security mechanisms to mitigate the threat. This may involve repositioning offloaded functions, refining security models, or deploying new detection strategies in real time.

3.4.1.2. Architecture, Testbed and Setup

The CNIT sub-use case architecture consists of data plane and control/mgmt. layers working in tandem to enhance security monitoring and response. The core components include:

1. DFE/WAI-enabled Data Plane Devices: These include smart NICs and programmable switches that support high-speed feature extraction and AI-driven security offloading. The DFE module extracts relevant security telemetry, while WAI applies AI models to detect threats at line rate.
2. AI Collector and Attack Profiler (provided by Montimage): This component receives telemetry data from DFE modules to analyze and profile novel threats. Once a new attack is identified, mitigation strategies are formulated and communicated back to the Security Orchestrator.

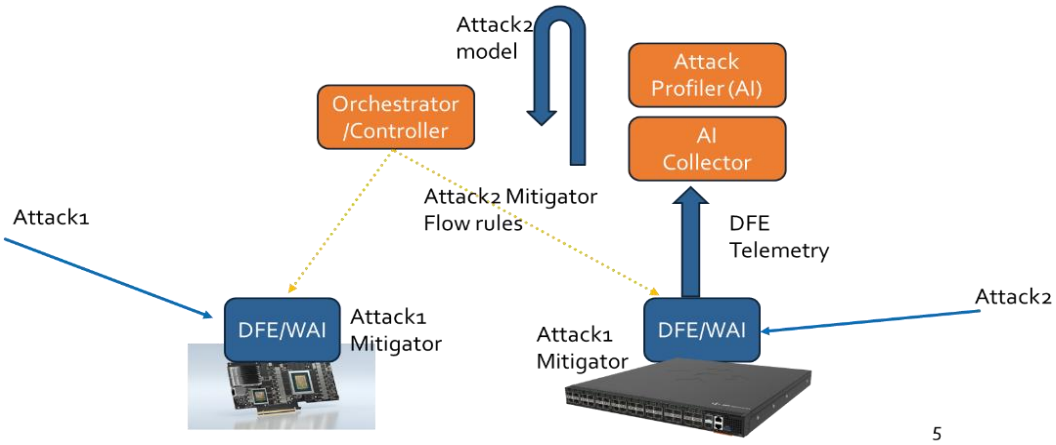

Figure 16: Pilot 4, Use Case 1

The testbed implementation leverages heterogeneous network infrastructure, combining high-performance programmable switches and smart NICs (DPU) with backend AI processing. Real-world attack scenarios are simulated/emulated, and DFE telemetry is collected to refine the AI-driven security functions. The dynamic enforcement mechanism ensures that security policies are continuously adapted, allowing for real-time threat mitigation and optimized placement of offloaded security functions, even in the presence of attacks not 100% profiled. Moreover, evaluation against adversarial attacks attempting the models already deployed in the data plane will be carried out to assess the robustness of such models to perturbation attacks aiming at inducing a wrong classification.

The validation will be assessed in the CNIT ARNO Labs. The testbed employs 100Gb/s connectivity with different available network topologies, resorting to BMv2 switches, Tofino 1 switch, and a number of Dell servers equipped with GPU and inter-connected using NVIDIA Bluefield-2 DPU Dual Port boards at 100Gb/s, employing DOCA libraries for hardware acceleration. Final validation may also include the utilization of the NVIDIA Bluefield-2 with embedded GPU, a special server-oriented device onboarding a GPU.

3.4.1.3. Involved Services and Components

Table 5 reports the needed components required for the validation of the sub-use case 4.1.

Table 5: SubUC 4.1 services and components

Service	Component
AI-based behavioral analysis	WAI and DFE
P4-based Network Analytics	DFE Telemetry
Security Monitoring	AI-driven security monitoring for anomaly detection and root cause analysis in IoT networks
Security by design Orchestration	Security Orchestrator

3.4.1.4. Validation Scenarios

The goals of UC4.1 will be illustrated in the following phases:

- **Phase 1**: **Orchestration/controller deployment and configuration of WAI + Known attack launch mitigated entirely by the data plane**
  - **Scenario**: Discover the edge node capabilities and configure an offloaded WAI function (e.g., DDoS mitigator) in physical node (e.g., switch, DPU). Launch DDoS attack against the cluster.
  - **Validation goals**: Deployment of security functions at the data plane (Requirement 4.1.4) in inter- and intra-edge scenarios (Requirement 4.1.2). The attack should be intercepted and mitigated directly at the network node without reaching the servers (Requirement 4.1.1).
  - **Metrics**: WAI latency (KPI 4.1.4), power consumption reduction (KPI 4.1.3 and KPI 4.1.5), internal DFE processing latency (KPI 4.1.1)
  - **Expected outcome**: Confirm the WAI/DFE solution's deployment, scalability, security enhancements, and energy optimization.
- **Phase 2**: **Not 100% profiled attack (e.g., adversarial attack)**
  - **Scenario**: Launch a second attack, activate DFE telemetry and feed P4-based IDS, proactive intervention of the Orchestrator/Controller to activate a new offloaded network function

- **Validation goals**: The attack should be analyzed in real time and the most suitable countermeasure should be taken: if one offloaded network function blocking the attack is available, a rapid deployment should be enforced. Discover the attack mitigation model and update the P4 DNN weights (Requirement 4.1.3 and 4.1.4).
- **Metrics**: DFE processing latency (KPI 4.1.1), DFE computational efficiency (KPI 4.1.2)
- **Expected outcome**: Confirm the DFE streaming telemetry solution's deployment, efficiency, scalability.

3.4.2. Sub-Use Case 4.2: Including AI-assisted network slicing for efficient resource utilisation and continuous monitoring and analysis

3.4.2.1. Description

The disaggregation and deployment of AI model components across network slices creates an efficient, adaptive, and energy-efficient architecture. By analyzing and disaggregating the AI/ML model based on computational needs and data dependencies, individual components can be placed on programmable data plane devices. This distributed approach reduces reliance on centralized servers, lowers latency, improves responsiveness, and optimizes resource utilization. Continuous real-time monitoring and feedback loops allow for dynamic reconfiguration of slices in response to changing network conditions or application demands, ensuring high performance and adaptability.

3.4.2.2. Architecture, Testbed and Setup

The architecture is designed to support the distributed deployment of sliced AI/ML models across programmable network elements, enabling low-latency, energy-efficient inference with runtime adaptability. The setup reflects a realistic, federated networking environment with heterogeneous hardware and control layers. The primary objective is to validate the proposed AI slicing and deployment methodology under varied and dynamic network conditions.

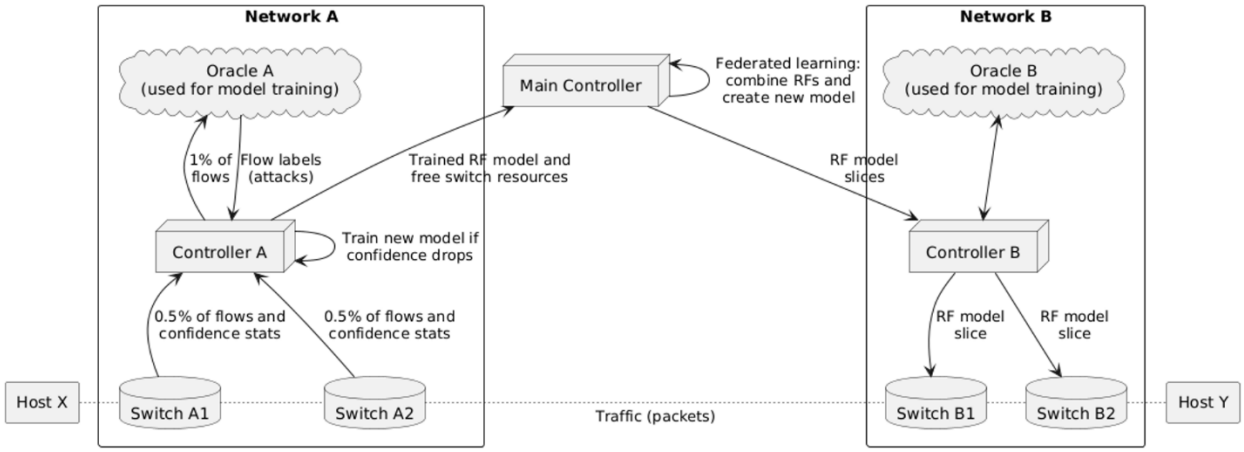

Figure 17: Pilot 4, Use Case 2, ELTE sub use case Architecture

Figure 17 presents the high-level architecture of the testbed used for experimentation and validation. The testbed consists of two interconnected domains (Network A and Network B), each representing a logically separated but collaboratively managed segment of the overall infrastructure. These domains are composed of the following key elements:

Programmable Data Plane Devices: Each network includes multiple programmable switches (e.g., Switch A1, A2, B1, B2) capable of executing lightweight AI inference tasks using preloaded model slices. These switches also collect flow statistics and inference confidence levels for local monitoring and feedback.

Distributed Control Plane: Each network domain features a local controller (Controller A and Controller B) responsible for managing the lifecycle of AI model slices deployed within their domain. Controllers gather telemetry data from their respective switches and interact with an oracle for model training.

Training Oracles: Oracle A and Oracle B serve as trusted sources of ground truth data, enabling supervised training and retraining of AI models when required. These components simulate the availability of labeled data used for refining or updating inference models.

Federated Coordination Layer: A central **Main Controller** acts as the orchestrator of the federated learning process. It aggregates models or slices trained in separate domains and combines them into a global model using ensemble or federated learning techniques (e.g., merging Random Forest classifiers). The resulting model is redistributed in sliced form for further deployment.

Monitoring and Adaptation Mechanism: Each controller monitors the inference confidence levels and flow characteristics. When degradation is detected (e.g., below a predefined confidence threshold), retraining is triggered locally, and updated model slices are pushed back into the network. Model updates are continuously integrated into the federated system via the Main Controller.

Figure 18 below depicts the low-level architecture with the hardware and software requirements.

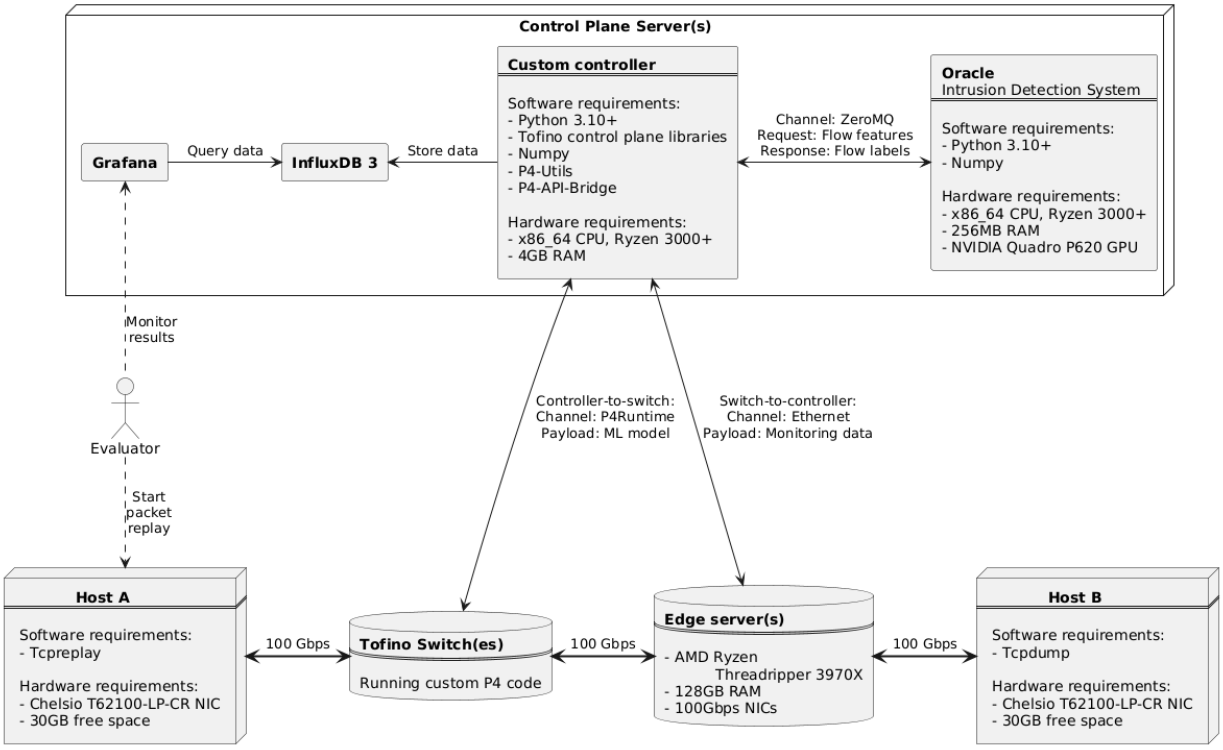

Figure 18: Pilot 4, Use Case 2, ELTE sub use case Low level architecture

3.4.2.3. Involved Services and Components

Programmable Data Plane Devices: Devices such as programmable switches (Intel Tofino), and edge servers that can host AI model slices (P4 or P4-eBPF) and execute them in real-time.

AI/ML disaggregator: Tools and frameworks for slicing large AI/ML models into smaller components and distributing them across the network.

Controllers: Each network domain features a local controller, responsible for their domain. And a central Main Controller acts as the orchestrator of the federated learning process using P4Runtime.

Monitoring and telemetry: Systems that can provide real-time feedback on the performance of the AI slices, including energy consumption, latency, and resource utilization.

3.4.2.4. Validation Scenarios

The goals of UC4.2 will be illustrated in the following phases:

Phase 1: Framework Development and Slicing Design

Scenario: Develop and test a prototype framework capable of slicing AI/ML models into components that can be deployed on network devices.

Validation goals:

- Establish the correctness and feasibility of the model slicing approach.
- Validate basic model behavior after slicing.
- Lay the groundwork for performance improvements related to later KPIs (e.g., energy efficiency, latency).

Metrics:

- The success rate of model slicing.
- Output similarity between full and sliced models.
- Slicing time and deployment overhead.

Expected outcome:

- Slicing is functionally correct and reproducible across models.
- No significant degradation in output quality or behavior.
- Basic performance baselines are established for future comparison.

Phase 2: Testbed Expansion and Early Testing

Scenario: Deploy sliced models on real programmable network devices (e.g., P4 switches) to validate early system functionality and observe key performance indicators.

Validation goals:

- Validate initial **latency reduction** (KPI 4.2.2) by comparing centralized vs. in-network inference.
- Assess if **AI model accuracy is maintained** (KPI 4.2.4) post-slicing and deployment.
- Test framework integration with diverse hardware types (REQ-4.2-1 and REQ-4.2-4).

Metrics:

- End-to-end inference latency (before and after slicing).
- Accuracy metrics (e.g., precision, recall, F1-score) for sliced vs. original models.

Expected outcome:

- Model can be deployed (REQ-4-2.1)
- AI computation is done on edge devices (REQ-4.2-3)
- Sliced models reduce overall latency compared to traditional architectures (KPI 4.2.2).
- No significant loss in model accuracy after deployment (KPI 4.2.4).

Phase 3: Optimization and Dynamic Scaling

Scenario: Refine the slicing framework to optimize runtime performance and introduce dynamic adaptation mechanisms.

Validation goals:

- Improve **energy efficiency** of inference tasks (KPI 4.2.1).
- Enable dynamic reallocation or scaling of slices during runtime based on performance and load. (REQ-4.2-2)
- Validate latency and accuracy stability under fluctuating workloads.

Metrics:

- Energy consumption.
- Runtime slice reallocation time.
- Performance deviation before and after adaptation.

Expected outcome:

- Noticeable improvement in energy efficiency over earlier phases (KPI 4.2.1).
- Slice migration and reallocation processes work with low overhead (REQ-4.2-2).
- System remains responsive and efficient under varying conditions.

Phase 4: Large-Scale Testing and Validation

Scenario: Conduct stress testing and full-system validation under high load and complex network conditions.

Validation goals:

- Demonstrate **resource utilization optimization** across multiple devices (KPI 4.2.3).
- Confirm robust **dynamic adaptation** capabilities under real-time network and traffic changes (KPI 4.2.5).
- Validate that all previously introduced optimizations hold under scale.

Metrics:

- Load distribution (CPU, memory, bandwidth) across the network.
- Inference accuracy, latency, and energy metrics under peak load.
- System uptime and fault tolerance during adaptation.

Expected outcome:

- Resource utilization is balanced and efficient across heterogeneous hardware (KPI 4.2.3).

- The system adapts dynamically with minimal service disruption (KPI 4.2.5). Real-time monitoring of slices work without errors (REQ-4.2-4)

3.4.3. Sub-Use Case 4.3: Employing software-defined radio for agile payload communication

3.4.3.1. Description

Integration of Software-Defined Radio (SDR) can enable agile, adaptive payload communication in next-generation 6G networks. SDR offers a flexible, software-based radio architecture capable of operating across multiple frequency bands and communication protocols. This adaptability is critical in dynamic and heterogeneous network environments. In UC4.3, SDRs are enhanced with machine learning-driven channel prediction, which enables real-time analysis of wireless channel conditions. By forecasting future states based on historical and real-time data, the system can proactively select the most suitable frequency and protocol, ensuring reliable and efficient communication. This approach significantly improves link stability and responsiveness under fluctuating conditions.

To further optimize spectrum usage, AI-powered cognitive radio functionality is introduced. Unlike traditional static frequency allocation, cognitive radio systems dynamically manage spectrum resources, identifying underutilized bands ("white spaces") and reallocating them as needed. This leads to higher spectral efficiency and reduced network congestion. Additionally, reinforcement learning (RL)-based channel switching mechanisms are implemented to maintain communication quality during adverse conditions. RL algorithms such as Q-learning, deep Q-networks (DQN), and multi-armed bandits (MAB) continuously learn from network performance and adapt channel selection strategies. This allows for seamless transitions between frequency bands when degradation is detected, which is essential in high-mobility or mission-critical scenarios.

In summary, this use case integrates SDR with intelligent control mechanisms—including ML-based prediction, cognitive radio, and RL optimization—to create a robust, adaptive communication framework. These technologies will be applied particularly in scenarios addressing adversarial threats and jamming mitigation, as explored in Use Cases 2.1 and 4.4.

3.4.3.2. Architecture, Testbed and Setup

The CERTH testbed is used in UC4.3 uses the same testbed and components as UC2.1, adding SDR Frequency and Protocol AI/ML Switching. The testbed is described in detail in section 3.2.1.2, in Figure 4. The architecture is outlined in Figure 19.

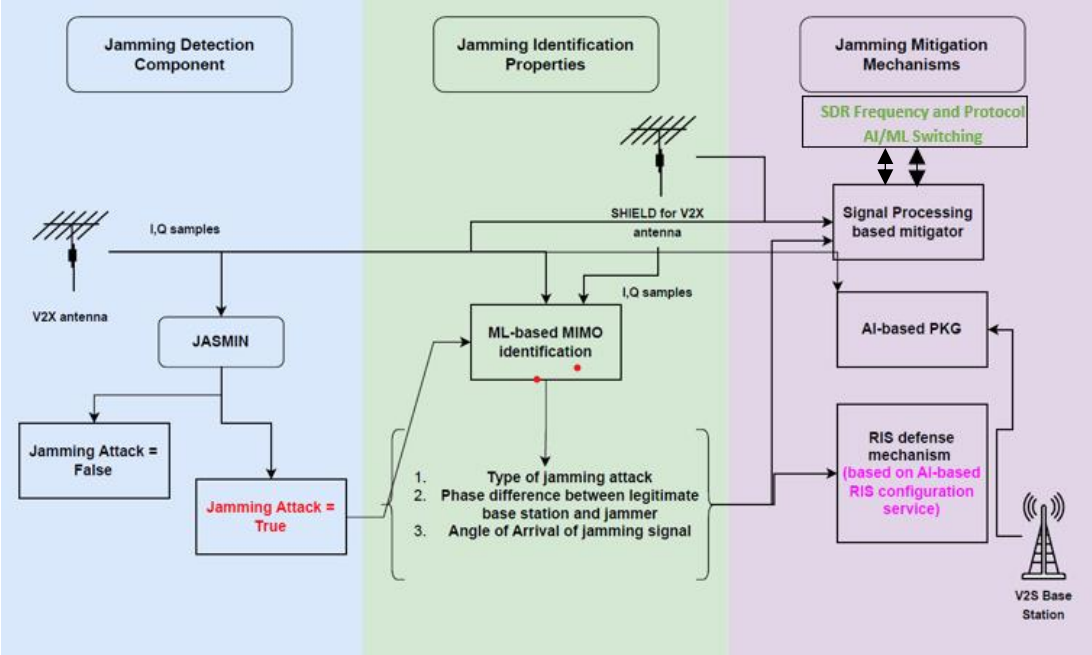

Figure 19: Architecture of UC4.3: This UC utilizes UC2.1 components, adding SDR Frequency and Protocol AI/ML Switching.

3.4.3.3. Involved Services and Components

Table 6 lists all the components involved in the validation of Sub-Use Case 4.3. This UC 4.3 is complementary to UC 2.1 and to UC 4.4. The adversary detection and mitigation mechanisms developed there will be systematically evaluated in tandem with an intelligent ML/AI-driven protocol and frequency switching via SDR introduced in UC4.3 as an additional adversary-attack mitigation measure.

Table 6: SubUC 4.3 services and components

Service	Component
AI-based anti-jamming	JASMIN & Filter Mitigation

3.4.3.4. Validation Scenarios

The proposed validation scenario of the detection framework involves with the validation of the sub-modules that compose it: Three interconnected sub-modules, each targeting a specific class of network anomalies or threats within the 5G/6G environment. These sub-modules collectively enable early threat detection and adaptive mitigation strategies using Software-Defined Radio (SDR).

Step 1a: Physical Layer Threat Detection validation

In this step we validate the sub-component that focuses on identifying adversarial attacks at the physical layer, such as jamming, spoofing, or other forms of signal interference. It continuously monitors the radio environment for anomalies indicative of external disruptions. The UC will use

the same physical attack detection and mitigation in UC2.1 components. This step is linked with requirement S6-S-C1 - JASMIN & Filter Mitigation.

Step 1b: Network Traffic Bottleneck Identification

In this step we validate the sub-component that focuses on detection of irregularities in traffic flow that may indicate congestion or malicious interference. The aim is to highlight areas where performance degradation is likely due to resource exhaustion or denial-of-service behaviour. This is linked to A-KPI 4.3-4.9.

Step 1c: QoS Anomaly Detection in Services

In this step we validate the sub-component that focuses on monitoring the Quality of Service (QoS) across network services, detecting deviations from expected performance levels. Anomalies could signal underlying issues such as targeted service disruption or infrastructure misconfiguration. This is linked to A-KPI 4.3-4.10.

Step2: Mitigation of anomalies detected

All detection modules leverage established machine learning techniques to analyse and classify anomalies in real time. Based on the detection outcomes, appropriate mitigation strategies are autonomously selected and executed. These may include frequency and/or protocol switching, primarily driven by reinforcement learning (RL) algorithms. Mitigation is performed via SDR, enabling rapid and flexible adaptation to evolving threat conditions. Both mitigation actions developed in the UC and some developed in UC4.4 will be utilized. This step is linked with requirement S6-S-C1 - JASMIN & Filter Mitigation. This step is linked with A-KPI 4.3-4.6, A-KPI 4.3-4.7, A-KPI 4.3-4.8, A-KPI 4.3-4.9, A-KPI 4.3-4.10.

This framework showcases the capabilities of resilient, self-healing networks by integrating intelligent threat detection with adaptive, software-defined communication controls. Figure 20 graphically presents the various steps of the validation scenario.

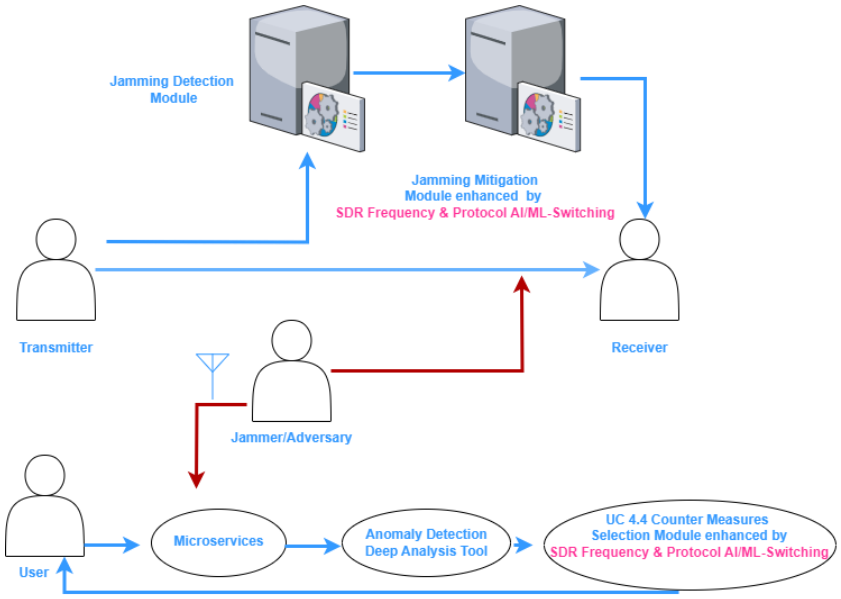

Figure 20: Sequence Diagram for use case UC4.3 depicting the various steps of the validation scenario. Attacks and related mitigations described by other UC are utilized (UC 2.1 upper part and UC 4.4 lower part)

3.4.4. Sub-Use Case 4.4: AI-driven microservices orchestration in 6G networks

3.4.4.1. Description

Use Case 4.4 explores the integration of AI for intelligent orchestration of microservices in 6G networks, focusing on enhancing flexibility, scalability, and resilience. Microservices, due to their modular and decoupled nature, are well-suited for the dynamic demands of 6G environments, where rapid deployment, adaptive resource management, and real-time responsiveness are critical.

This use case leverages AI algorithms to monitor and optimize microservice deployment, scaling, and operation. By incorporating predictive analytics, the system can proactively adjust resource allocation based on changing network conditions. Microservices are designed with distinct resource footprints—some CPU-intensive, others network-heavy—enabling the system to detect anomalies in behaviour under attack scenarios, even when the attack type is unknown.

This use case involves and showcases the following functionalities:

- Microservices Profiling: Pre-process procedures to map the behaviour of microservices under normal traffic and workload conditions.
- Real-time Anomaly Detection: Online procedures for detecting irregular resource usage patterns indicative of potential attacks.

- Risk Classification: Classification of network load based on the degree of risk targeting to the isolation malicious traffic.
- Automated Anomaly Mitigation: Online procedures that try to heal, mitigate or deflect detected anomalies or attacks.

Use Case 4.4 demonstrates how AI can enable adaptive, secure, and autonomous service orchestration, contributing to a more robust and intelligent 6G ecosystem.

3.4.4.2. Architecture, Testbed and Setup

The implementation of Use Case 4.4 as well as any relevant validation and experimentation activities will be carried out on CERTH's 5G-SDN testbed. The testbed is built upon a Software Defined Networking (SDN)-powered 5G infrastructure. More specifically, the core 5G network functions are deployed as containerized microservices. This containerized 5G network stack is achieved through standards such as Open5G and Free5G. The considered microservices include the ones corresponding to the 5G core elements, NGINX and microservices that can be attacked via Metasploit, particularly suitable for evaluating the anomaly and attack detection mechanisms. In this regard, the testbed is equipped with the following services:

- AI-Based Intrusion Detection System (IDS): An AI-powered IDS designed to monitor and analyse traffic in real time, identifying abnormal behaviour and potential threats with minimal latency.
- SDN-Based Microservices Resource Consumption Monitoring Engine: A dedicated monitoring tool to track microservice resource consumption—including CPU, memory, and network usage—to detect anomalies that might indicate network underperformance due to excessive resource consumption or potential security breaches.
- AI-Driven Mitigation Engine: Coupled with the IDS, this component responds to detected anomalies by executing real-time countermeasures to contain threats and maintain service continuity in the microservice environment. The selection and enforcement of pertinent countermeasures is handled by a dedicated module, the so-called Countermeasure Selection Module. This module considers, among others, the insights provided by the SDR Frequency and AI/ML-Switching protocol, as described in UC4.3.

Finally, a suite of attack models concerning mostly DoS attacks on different network protocols that have been developed by CERTH will be utilized in this use case.

The main functional building blocks of the testbed being used in this use case are illustrated in Figure 21.

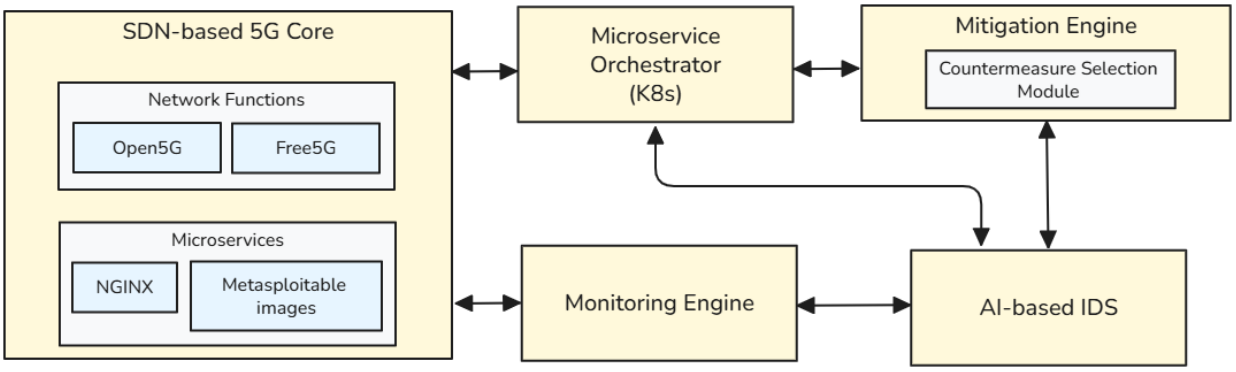

Figure 21: CERTH testbed in UC4.4

3.4.4.3. Involved Services and Components

Sub-Use Case 4.4 leverages the:

- **AI-based Intrusion Detection service:** For real-time detection of DoS attacks based on AI models trained on multimodal features extracted from the network traffic.
- **AI-based behavioural analysis service**: For embedding microservice profiling mechanisms
- **Security by Design Orchestration service**: To perform network slicing considering anomaly detection techniques
- **Security-performance balancer**: To ensure balance between network performance and security

Table 7 lists all the components involved in the validation of Sub-Use Case 4.4.

Table 7: SubUC 4.4 services and components

Service	Component
AI-based Intrusion Detection (1)	Multimodal Fusion Approach for Intrusion Detection System for DoS attacks
AI-based Intrusion Detection (2)	Lightweight SDN-based AI-enabled Intrusion Detection System for cloud-based services
AI-based behavioural analysis	Microservice behavioural analysis for detecting malicious actions
Security by Design Orchestration	Slice orchestration and slice management for beyond 5G networks
Security-performance balancer	Security-performance balancer

3.4.4.4. Validation Scenarios

The goals of UC4.4 will be illustrated using two validation scenarios presented below.

Validation Scenario 1: Microservice scaling

The first validation scenario evaluates the ability of the orchestration system to dynamically scale microservices in response to real-time demands and heavy workloads while maintaining quality of service (QoS). The experimental setup is built upon a containerized microservice architecture with AI-enabled autoscaling capabilities and integrated monitoring tools. These tools continuously track key performance metrics, such as CPU usage and memory consumption, providing the necessary observability for intelligent decision-making.

As a first step, microservice monitoring data is collected under normal operational conditions to achieve microservice profiling under normal conditions is achieved. That way, baseline profiles for microservices to understand typical behavior under standard traffic and workload conditions will be developed and observed metrics baselines are established to be used as benchmarks for detecting anomalies and triggering scaling decisions during stress conditions.

Subsequently, a series of stress scenarios will be considered for testing and validating the system's responsiveness. For instance, one such scenario involves sudden surges in network traffic, designed to increase load and potentially exhaust the resources allocated to specific microservices. In this context, the system is expected to identify resource saturation, automatically trigger scaling actions in a timely manner, and restore the QoS parameters to their target levels.

Overall, this validation scenario will test and validate the ability of the system to (i) automatically perform dynamic scaling of microservices in response to varying network loads, ensuring that resources are allocated optimally to maintain performance and service continuity (ii) recover quickly from disruptions and ensure that critical services remain operational. Therefore, it directly supports and verifies the use case requirements subUC-4.4-1, subUC-4.4-2, and subUC-4.4-3, focusing on dynamic resource management, real-time adaptation and scalability.

Validation Scenario 2: AI-based Attack Detection & Mitigation

The following scenario describes the key stages involved in the validation of the proposed approach on detecting and mitigating cyber threats targeting microservices within a 6G environment. This process combines SDN-based monitoring with AI-driven analytics to ensure timely and effective protection.

Step 1: Traffic Ingestion and Microservices Deployment

The process begins with the introduction of both user and malicious traffic into the network. Microservices are deployed across the 6G infrastructure—including core and edge environments—simulating a realistic service landscape exposed to potential adversarial behaviour.

Step 2: Monitoring and Data Collection

A monitoring broker gathers key data from the SDN network, including traffic statistics and resource consumption metrics related to deployed microservices. This real-time data feed serves as the input for the anomaly detection process.

Step 3: Anomaly Detection and AI-Based Analysis

The collected monitoring data is first evaluated by a statistical analysis module to detect irregularities in traffic patterns or resource behaviour. When anomalies are identified, the system escalates the issue to an AI-based deep analysis tool. This tool performs a detailed inspection to classify the threat, extract relevant indicators (e.g., attack type, affected components, attacker IPs), and determine the scope of the impact. Success will be assessed based on detecting these attacks with a detection rate higher than 80% (KPI 4.4) while maintaining a low rate of false positives (KPI 4.5).

Step 4: Countermeasure Selection and Execution

Based on the results of the AI analysis, the system selects appropriate countermeasures to neutralize the threat. This may include adaptive responses such as resource reallocation, protocol switching, or isolation of affected services. These actions are carried out through the software-defined radio and orchestration framework, ensuring system stability and service continuity.

The two last steps also validate subUC-4.4-4 and subUC-4.4-5 requirements, by ensuring the system's ability to detect and mitigate anomalies by employing advanced security measures within the orchestration process, while ensuring service continuity.

This streamlined workflow highlights the role of intelligent automation in safeguarding microservices within dynamic 6G environments, enabling rapid detection, classification, and mitigation of emerging threats. The process is graphically depicted in Figure 22.

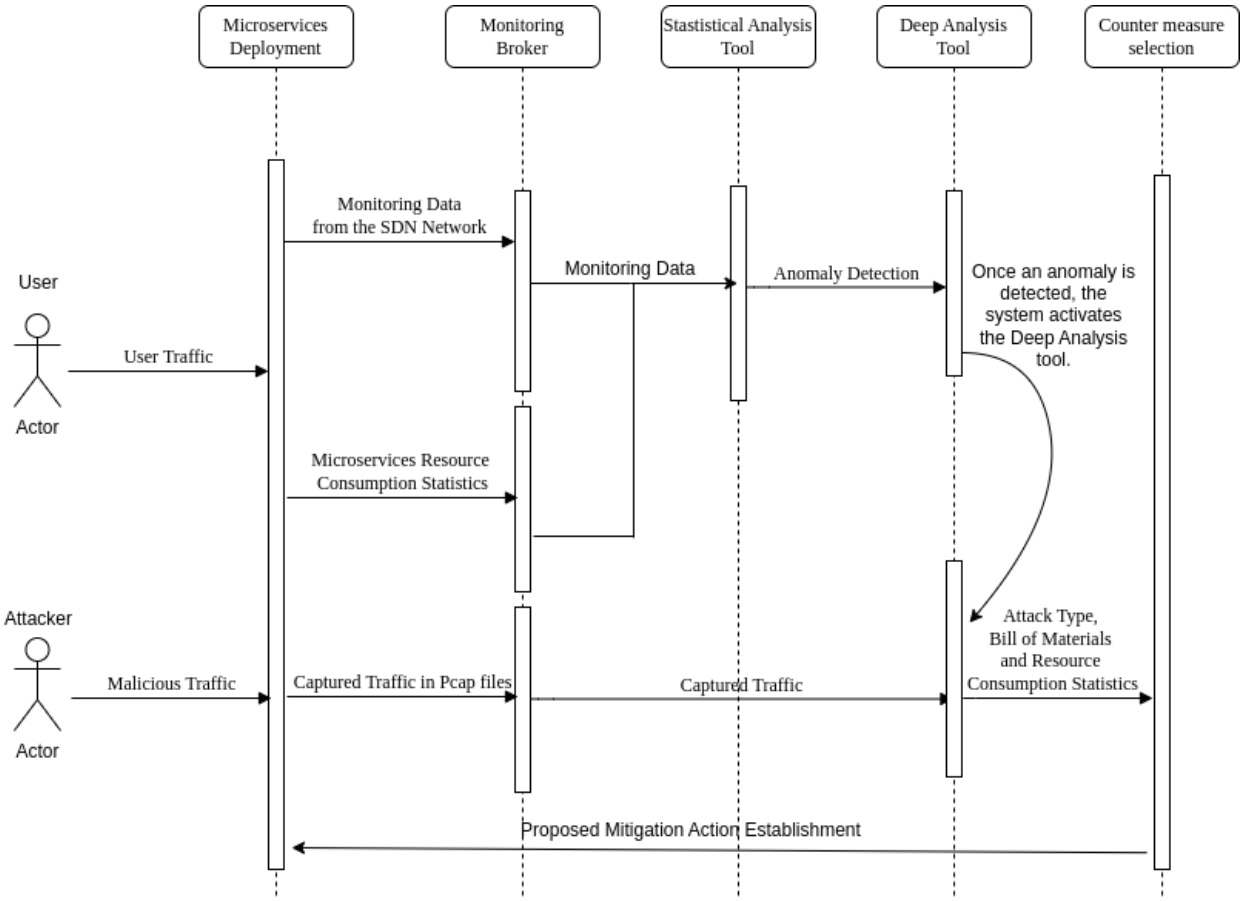

Figure 22 Sequence Diagram for use case UC4.4 depicting the various steps of the validation scenario.

3.4.5. Sub-Use Case 4.5: Enabling optimised and explainable MTD for 6G edge-to-cloud continuum

3.4.5.1. Description

UC4.5 focuses on enhancing the security and efficiency of the 6G edge-to-cloud continuum by leveraging service mobility for Moving Target Defense (MTD) strategies. MTD allows the dynamic and proactive protection of ICT infrastructure. However, it is challenging to facilitate MTD in an optimal and autonomic way. This sub-use case aims to showcase:

1. Orchestration of MTD actions, performing MTD operations such as IP/port shuffling, reinstantiation, multi-domain live migration, and transfer to TEE-environments of Virtualized Network Functions (VNFs) and Cloud-native Network Functions (CNFs).
2. Optimization and automation of MTD strategies, across edge and core domains in a Multi-access Edge Computing (MEC) environment, balancing security, Quality of Service (QoS), and resource consumption.

3. Provide explainability for MTD decisions using Explainable AI (XAI) techniques to ensure transparency and trust in deep ML-driven optimizations.

The use case will integrate various data sources such as infrastructure performance monitoring, vulnerability scans and threat intelligence from the 6G network to devise dynamic policies. Moreover, explainable techniques for AI will be integrated to provide insights into human security experts about the self-driven MTD operation.

UC4.5 bridges adaptive cybersecurity, autonomous network orchestration, and explainable AI, making it a critical enabler for resilient and intelligent 6G infrastructures.

3.4.5.2. Architecture, Testbed and Setup

The testbed used in UC4.5 is integrated between PNET 5G testbed, Patras, in Greece, and ZHAW TEE testbed, in Switzerland. The testbed includes the following components:

(In PNET 5G testbed)

- A 5G network with a MEC-setup, comprising a core domain and two edge domains implemented with Open5Gs.
- A distributed architecture running the UPFs in the edge domains.
- An NFV Orchestrator implemented with OSM.

(In ZHAW testbed)

- A server equipped with an AMD Epyc 4th gen CPU enabling TEE isolation with SEV-SNP (Secure Encrypted Virtualization-Secure Nested Paging) technology.
- A Kubernetes cluster operating CNFs possibly transferred from the PNET testbed to the TEE enclave.

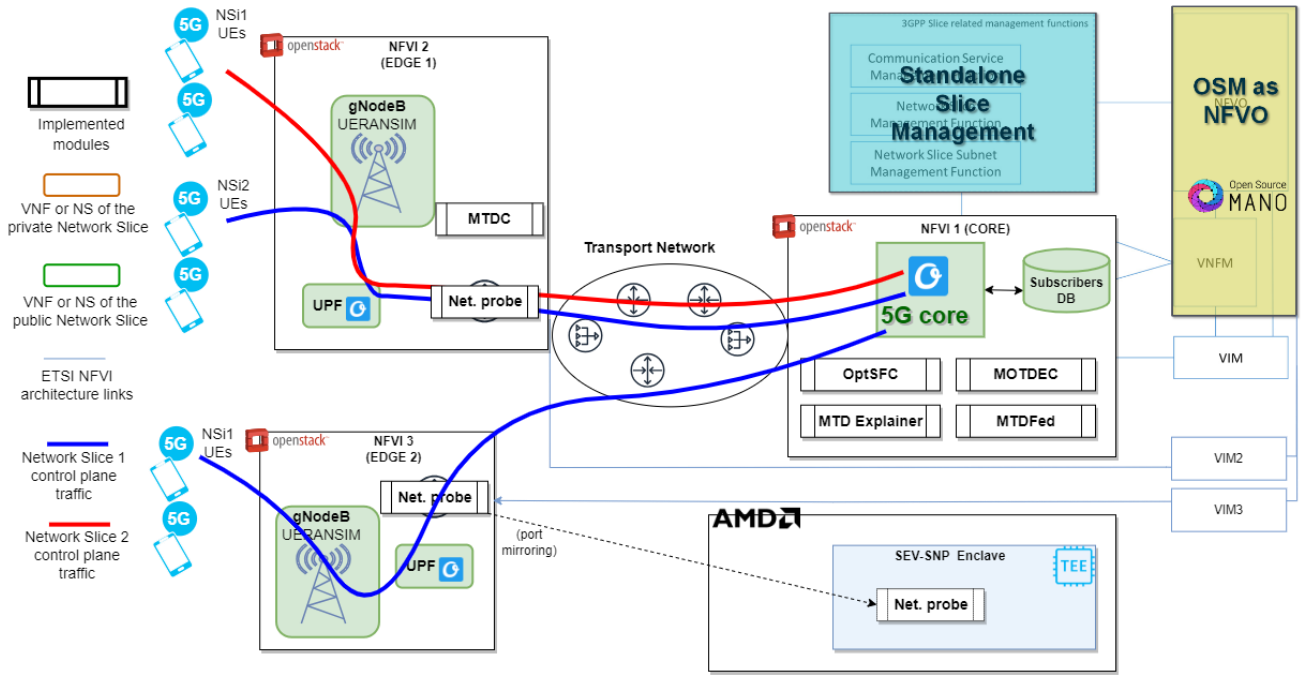

Figure 23: Testbed of sub-use case 4.5

3.4.5.3. *Involved Services and Components*

UC4.5 implements the AI-based MTD service, which is composed of the following components:

1) MTD Controller

The MTD Controller dynamically executes MTD actions (e.g., live migration of VNFs/CNFs, service reinstantiation, IP/port shuffling) to disrupt attack surfaces while maintaining service continuity. It focuses on creating minimal disruption to the protected service during such actions, but optimizing live migration techniques and using lightweight CNFs over VNFs.

2) MTD Strategy Optimizer

This is the cognitive component of the AI-based MTD service developed for this use case and employs deep-RL based optimization to dynamically adjust MTD actions, vis-à-vis of a multi-objective problem spanning 3 objectives: *Security* (reducing attack surfaces and mitigating threats like data exfiltration and malware infection), *QoS/QoE* (maintaining low latency, high throughput, and service reliability), and *Resource Efficiency* (minimizing computational and network overhead from MTD operations). The MTD strategy optimizer operates across edge nodes and core cloud nodes for a centralized view and global MTD optimized strategy aware of the state of the full network.

3) MTD Explainer

The MTD Explainer uses XAI for deep-RL models and post-hoc explanation techniques to clarify why specific MTD actions (e.g., migration vs. shuffling) for specific CNFs were chosen. It provides insights into trade-offs between security, QoS, and resource usage. Finally, it enables human operators to validate and refine automated MTD strategies.

4) MTDFed

MTDFed is a FL-based cooperative optimisation for virtual network operators (VNOs) wanting to improve their MTD strategy without sharing sensitive data on their own network resources and traffic data. It is a FL extension to the MTD Strategy Optimizer component, providing a multi-tenant optimisation of MTD strategies, where VNOs run their own decision system while also cooperating in learning a model optimizing MTD strategies.

3.4.5.4. Validation Scenarios

Various scenarios will be used to validate the AI-based MTD service on different fronts, specifically:

1) Proactive Security: threats and attack scenarios will be used to validate the MTD service's reduction of the likelihood of successful exploit (LSE).
    a. Related KPIs: A-KPI 4.15
    b. Related UC Requirements: REQ-4.5-2
2) Efficient Resource Utilization: scenarios will present various workloads to measure the scalability of the MTD service and its optimization performance on resource overhead.
    a. Related KPIs: A-KPI 4.13
    b. Related UC Requirements: REQ-4.5-1
3) Transparent Automation: XAI outputs will be analyzed by humans in a qualitative evaluation to make sure that AI-driven MTD decisions are interpretable and auditable.
    a. Related KPIs: A-KPI 4.17
    b. Related UC Requirements: REQ-4.5-4
4) Seamless Edge-to-Cloud Integration: MTD operations across distributed network domains (e.g., from edge to core domains) will be evaluated for minimal disruption of the protected services.
    a. Related KPIs: A-KPI 4.12
    b. Related UC Requirements: REQ-4.5-1

3.4.6. Sub-Use Case 4.6: DoS attack detection by payload self-monitoring.

3.4.6.1. Description

Use Case 4.6 is an additional use case, created in the preparation work of D2.2. This addition was meant at separating it from use case 4.1 into which it was described as a secondary and ancillary research activity. This separation and novel use case creation was decided for clarity.

This use case features a low readiness and explores the relevance of exploiting the extraction of software payload performance ratio as a metric for DoS detection, without incurring unsustainable penalty. The performance ratio derives from probes duly and precisely inserted inside the payload control flow graph and able to capture the global program speed of execution. As this use case success aligns with the identified technical risk as stated in the proposal and defined as "*Control time and frequency metadata extraction or exploitation cannot be done",* it will be initiated with a proof of concept and feasibility study, with the objective to better grasp the problem to solve (ie, DoS attack on the payload), and means (ie, the payload performance ratio). This study will also consider the benefits of collecting other metrics (e.g., cache misses' ratio, Processor Monitoring Counter (PMC)-derived Instruction per cycle (IPC), payload 's CPU usage rate versus the other running processes). Last, the study will consider if and how machine learning can be useful, defining notably possible training data if practicable. According to this initial research stage, an implementation of a proof of concept will be defined and showcased. To proceed efficiently with the initial study phase, we have engaged technical exchanges with MONT (i.e., which so called MMT probe delivers traffic anomaly detection) and exploits machine learning. MONT's expertise is therefore akin to the use case.

3.4.6.2. Architecture, Testbed and Setup

The feasibility study work will be done by setting up the testbed as defined below:

- Leverage of MMT as the victim code. Alternative victim code may be considered (e.g., L2Fwd network function, x86-compiled P4 smartNIC network function) is deemed more appropriate.
- Get is instrumented with TSS's self-monitoring to collect time series of the payload performance ratio (i.e., its speed of execution)
- Proceed to other metrics collection by TSS and MONT, initially defined as:
    - Cache misses
    - IPC (ie, Instruction Per Cycle)
    - CPU use rate (versus other process)
    - Other metrics potentially collected in ring-O (through a kernel module to be considered)
- Define representative scenarios for DoS attacks:
    - Flood attack with replicated sockets

- Flood attack with random sockets
- Qualify the merits and challenges of using machine learning for detection, notably considering the aspects of:
  - Attack detection logic (deterministic or probabilistic (i.e., ML-based) detection)
  - Qualification and existence (or generation) of a training data set, potentially used for the proof of concept.

This work will be carried out in TSS's own premises and shared with MONT.

The implementation of a proof of concept will be defined according to the feasibility study.

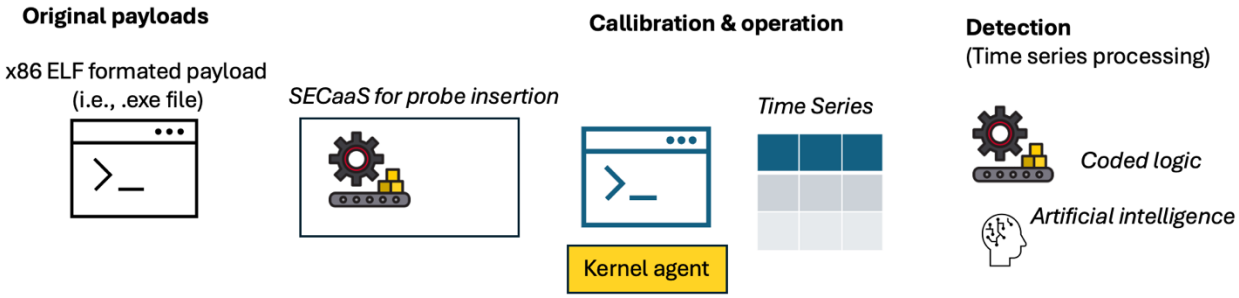

Figure 24. U.C 4.6 Architecture overview

3.4.6.3. Involved Services and Components

The use case implements the service TSS's self-performance monitoring. The use case also implements the SECaaS service for the binary rewriting needed for the self-monitoring probe insertion.

3.4.6.4. Validation Scenarios

The course of action regarding UC4.6 is the following:

Feasibility study validation:

The validation of this initial research includes:

- Assessment of the **modelization of a payload performance and associated metrics:**
  - Enumeration of performance variation causes (e.g., variation of data distribution resulting in an elevated/decreased cache miss rate)
  - Enumeration and characterization of other potential metrics collected at ring-0 (ie, kernel level) or ring-3 (ie, user application level)
- Analysis of the **detection logics** and relevance of machine learning:
  - Definition of the prediction logic, consuming performance rate and additional metrics.
  - Study of the usability of machine learning for the detection

  - Analysis of potential synthetic or real training data
- **Experiments**:
  - Synthesized DoS attacks causing performance variation causes and other metrics variation
  - Good match between theoretical predictions and experimental measurements
  - Analysis of false positives and negatives
- **Conclusion work**
  - Provide conclusions on the usability of self-performance monitoring for the detection of DoS attacks
  - Provides conclusions on the detection logics and relevance of machine learning therein
  - Define other potential usages for performance self-monitoring

PoC validation Implementation

The PoC will be carried out according to the feasibility study. Eventually, it will integrate the experiments conducted there and additional technical integration work showcasing the complete workflow. The PoC will integrate the following elements:

- Component design including
  - Synthetic attack generation
  - Deterministic or machine learning attack detector
- Showcasing the solution setup workflow
- Training or data set able to demonstrate:
  - Attack detection
  - False positives and negatives rate

Expected outcomes:

- Validate the relevance and usability of performance self-monitoring for DoS attack detection, in consideration of additional collectable metrics. This work will validate the relevance of a novel, today unused runtime metrics. Usability will be defined according to the criteria of accuracy (i.e., false positive and negative rates) and penalty (i.e., performance loss)
- Devise the detection logic for DoS detection as part of teamwork and validate/invalidate the relevance of machine learning therein.
- Find other and alternative potential use cases for performance self-monitoring.
- If practical and relevant, devise a PoC with its distinct components and showcase it.

4. KPI Evaluation

This section provides a detailed description of the Key Performance Indicators (KPIs) used to evaluate the performance and effectiveness of the different use cases. Each KPI is documented in a standardized template (see Table 8), ensuring consistency in measurement and reporting across testbeds and sub-use cases.

The KPI template includes the following fields: KPI-ID (a unique identifier in the format KPI <subUC>-<KPI-number according to D2.2>), Name and Description of the KPI, and Leading Partner. A-KPI stands for additional KPIs, i.e., those devised after the project started. Each KPI is linked to a specific sub-use case and testbed where it is validated, ensuring traceability. The template also maps KPIs to relevant NATWORK services, provides a Baseline (existing measurements before implementation, if it exists), and sets a Goal (target measurements, either qualitative or quantitative). Finally, the Means of Verification describes the methodology and tools used to measure and validate the KPI.

This structured approach ensures that performance targets are clearly defined, measurable, and aligned with project objectives. It also facilitates comparison between Baseline (if it exists) and post-implementation results, enabling objective assessment of progress and success.

Table 8: KPI template

KPI-ID	KPI <subUC>-<number>
Name	
Description	
Leading Partner	
Validation sub-UC & testbed	Indicate sub use case identifier and testbed where the KPI is validated
Mapping to services	Indicate the services concerned (refer to D2.3)
Baseline	Indicate any existing measurements before
Goal	Indicate target measurements (qualitative or quantitative)
Means of verification, methodology, tools	How the measurement was calculated or what tools were used to perform the measures

4.1. Use Case 1

4.1.1. Sub-Use Case 1.1

KPI-ID	KPI 1.1-1.1
Name	End-to-end compliance with latency tolerance
Description	Measures the ability of the orchestrated 6G slices to maintain latency thresholds across the edge-to-cloud continuum, especially under stress scenarios like DoSt attacks.
Leading Partner	UEssex
Validation sub-UC & testbed	NCL testbed
Mapping to services	S3-S-C1 Energy Efficient orchestration Secure-by-design orchestration Service
Baseline	There is no standard orchestration baseline for latency performance under DoSt attack scenarios. A baseline will be established during initial measurements using default Kubernetes orchestration without optimizations, serving as the comparative reference point.
Goal	Achieve ≤10% deviation from latency tolerance thresholds under dynamic conditions.
Means of verification, methodology, tools	Continuous monitoring via Prometheus and/or Kubemetrics during normal and attack conditions.

KPI-ID	KPI 1.1-1.2
Name	Energy waste: CPU utilization under normal/attack conditions to measure energy consumption (used to estimate Energy waste percentage)
Description	Measures the increase in CPU usage during DoSt attacks compared to normal operation, used to estimate the percentage of energy waste caused by inefficient scaling or attack-triggered load.
Leading Partner	UEssex
Validation sub-UC & testbed	NCL testbed
Mapping to services	S3-S-C1 Energy Efficient orchestration Secure-by-design orchestration Service
Baseline	Average CPU utilization of relevant services during stable operation (no attack, no dynamic scaling). Since no fixed baseline exists across use cases, it will be measured per scenario during early tests, referencing methodologies similar to those in FORK [1].

KPI-ID	KPI 1.1-1.2
Goal	Limit energy waste to within 10% deviation from baseline CPU consumption.
Means of verification, methodology, tools	CPU metrics collection using Prometheus; comparative analysis of CPU load during normal vs DoSt attack scenarios; analysis of orchestration decisions' impact on resource use.

KPI-ID	A-KPI 1.1-1.5
Name	Cluster Hygiene Scores (Number of vulnerabilities shared with score 8+/Total number of vulnerabilities)
Description	Indicates the proportion of high-severity vulnerabilities (CVSS score 8+) among the total reported vulnerabilities. A lower ratio implies a more secure and "clean" cluster.
Leading Partner	UEssex
Validation sub-UC & testbed	NCL testbed
Mapping to services	S3-S-C1 Secure-by-design orchestration Service
Baseline	Initial vulnerability reports without CTI optimization or orchestration-driven function placement.
Goal	Establish a cluster-specific hygiene score (range 0–1). A ratio <0.3 is considered indicative of a secure cluster; final thresholds will be cluster- and scenario-specific and refined during evaluation.
Means of verification, methodology, tools	Vulnerability scans using integrated security tools; scoring through CVSS; tracking shared vs high-severity vulnerabilities via CTI dashboard

KPI-ID	A-KPI 1.1-1.6
Name	Cluster CTI Exposed information Ratio (Number of vulnerability data parts revealed/Total information per CTI data)
Description	Measures the proportion of vulnerability information that is shared from a cluster's CTI data, relative to the total available data. This reflects the openness and effectiveness of the CTI exchange while balancing privacy and sensitivity.
Leading Partner	UEssex
Validation sub-UC & testbed	NCL testbed
Mapping to services	S3-S-C1

KPI-ID	A-KPI 1.1-1.6
	Secure-by-design orchestration Service Security-compliant Slice Management
Baseline	Initial vulnerability reports without CTI optimization or orchestration-driven function placement. CTI exposure settings without adaptive sharing based on security policies or risk level.
Goal	Achieve an exposed information ratio of 0.4–0.6, balancing information usage with privacy. This estimate reflects expected improvement through contextual orchestration and policy-aware CTI exchange.
Means of verification, methodology, tools	CTI messages, metadata tracking in STIX/TAXII exchange; audit of shared vs total data volume using cluster-local CTI agents and dashboards.

KPI-ID	A-KPI 1.1-1.7
Name	Cluster CTI Hidden information Ratio (Number of vulnerability data parts hidden/Total information per CTI data)
Description	Represents the fraction of vulnerability-related data intentionally withheld during CTI exchange, relative to the total CTI dataset. This reflects the level of confidentiality applied to shared intelligence and helps assess the trade-off between security transparency and data protection.
Leading Partner	UEssex
Validation sub-UC & testbed	NCL testbed
Mapping to services	S3-S-C1 Secure-by-design orchestration Service Security-compliant Slice Management
Baseline	Non-adaptive CTI exchange where either most data is exposed or overly restricted.
Goal	Maintain a hidden information ratio between 0.4 and 0.7, adapting based on cluster risk profile and policy constraints. This range reflects a balance between data protection and effective intelligence sharing; actual ratio will be dynamically tuned per cluster during runtime.
Means of verification, methodology, tools	CTI messages, metadata tracking in STIX/TAXII exchange; audit of shared vs total data volume using cluster-local CTI agents and dashboards.

4.1.2. Sub-Use Case 1.2

KPI-ID	KPI 1.2-1.3.1
Name	Time for remote attestation cycle for x86 payload
Description	Remote attestation induces the generation of the hashing of a memory footprint allotted to the payload by the operating system and its verification by a remote verifier.
Leading Partner	TSS
Validation sub-UC & testbed	Sub-UC 1.2 TSS's own testbed
Mapping to services	S14-F aka SECaaS
Baseline	No baseline as for an unprotected payload, the latency before its start is unexistant.
Goal	1 second.
Means of verification, methodology, tools	Several timestamps are inserted inside the payload, in the source code, bytecode (for WASM) or at executable level. They are placed at the original code entry point and once the operations linked to the proving (i.e., hashing of the memory footprint) are worked out, hence **excluding for the verifcation and blockchain block print operations, processed asynchronously).** As this the timing is workload dependent (i.e., the larger the code, the higher it takes to make its hashing), our measurement will be made using a **code of sufficient size.** This KPI may only makes sense if the code can only execute if the remote attestation test is positive. Alternative scheme may be considered, enabling the code to start before triggering its remote attestation. This model is viewed as far more relevant for 6G instant service start. Noticeably, with this "Attest after Starting" method **cancels this KPI** as dropping the timing to nil.

KPI-ID	KPI 1.2-1.3.2
Name	Time for payload decryption for x86 payloads
Description	AES decryption of the encrypted text section of the executable is processed before the resulting content is stored and execution starts.
Leading Partner	TSS
Validation sub-UC & testbed	Sub-UC 1.2 TSS's own testbed
Mapping to services	S14-F aka SECaaS
Baseline	No baseline. For un-encrypted payload, the latency is null.
Goal	The latency at start induces by workload decryption shall be **below 3 seconds.**

KPI-ID	KPI 1.2-1.3.2
Means of verification, methodology, tools	Several timestamps are inserted inside the payload, in the source code, bytecode (for WASM) or at executable level. They are placed at the original code entry point and once the AES decryption has been carried out. As this the timing is workload dependent (i.e., the larger the code, the higher it takes to make its hashing), our measurement will be made using a **code of sufficient size.**

KPI-ID	KPI 1.2-1.3.3
Name	Performance degradation during runtime caused by runtime verification and performance monitoring for x86 payloads
Description	Runtime integrity verifications and performance ratio metrics generation and collection impact the performance it measures.
Leading Partner	TSS
Validation sub-UC & testbed	Sub-UC 1.2 TSS's own testbed
Mapping to services	S14-F aka SECaaS
Baseline	No baseline. For un-verified and un-monitored payloads, the penalty impact is inexistent.
Goal	The performance degradation induced by the integrity verification and self-monitoring is kept below 10%.
Means of verification, methodology, tools	Several timestamps are inserted inside the payload at various locations to measure the impact caused by the aggregation of runtime integrity verification and performance self-monitoring. **For runtime integrity verification**: The performance penalty is workload-dependent (e.g., the smaller the code, the higher is the penalty induced), our measurement will **produce our tests on a set of representative workloads.** **For self-monitoring:** The contours of that function will be defined in D3.5. The type of timing measurements **will be defined accordingly**, possibly on a set of representatives workloads.

KPI-ID	KPI 1.2-1.3.4
Name	Overall energy waste for the aggregation of CIA hardening (i.e., confidentiality of the payload, remote attestation, runtime integrity verification and self-monitoring) for x86 payloads
Description	The CIA-hardening functions and the remote attestation all induce energy consumption, which shall be kept at sustainable level.
Leading Partner	TSS

KPI-ID	KPI 1.2-1.3.4
Validation sub-UC & testbed	Sub-UC 1.2 TSS's own testbed
Mapping to services	S14-F aka SECaaS
Baseline	No baseline. For unprotected payloads, no energy is consumed by security-related routines.
Goal	Overall energy budget for all security-related functions is limited to 10%
Means of verification, methodology, tools	Method: Instrument the workload with software-based Energy Estimation (e.g., RAPL, perf, PowerAI). Alternative method based on virtualization/cloud APIs (e.g., Cloud Carbon). Conditions: As energy waste induced by security is payload-dependent, we will select a **representative set of payloads.**

KPI-ID	KPI 1.2-1.4.1
Name	Feasibility study covering our security challenge
Description	WASM security will be enhanced into several directions of runtime integrity, confidentiality preservation and performance monitoring. These security enhancements will be made possible with add-ons on the WASM runtime, which feasibility study must be first carried, including the specifications of the required developments.
Leading Partner	TSS
Validation sub-UC & testbed	Sub-UC 1.2 TSS's own testbed
Mapping to services	S14-F aka SECaaS
Baseline	No baseline
Goal	Completedness of the feasibility study, covering Confidentiality, Integrity and Availability preservation. This will be included in D3.5
Means of verification, methodology, tools	Internal review of the sub section in D3.5

KPI-ID	KPI 1.2-1.4.2
Name	Development of novel WASM security functions (covering the CIA triad)
Description	WASM CIA-related security enablers will be developed according to the specifications as produced in the feasibility study as stated above.
Leading Partner	TSS
Validation sub-UC & testbed	Sub-UC 1.2 TSS's own testbed

KPI-ID	KPI 1.2-1.4.2
Mapping to services	S14-F aka SECaaS
Baseline	No baseline
Goal	Completeness of the development of WASM runtime integrity, confidentiality preservation and self-monitoring.
Means of verification, methodology, tools	Internal review. Check the conformity of the offered security hardening with the content of the feasibility study relevant with KPI 1.4.1

KPI-ID	KPI 1.2-1.4.3
Name	Alignment of WASM security enhancers with the KPIs 1.3.1/2/3/4, limiting the latency, performance penalty and energy waste
Description	The KPI aggregates all KPI 1.3.x to WASM security enhancers developed in NATWORK. This KPI depends on the KPI 1.4.1 stated above (i.e., feasibility study).
Leading Partner	TSS
Validation sub-UC & testbed	Sub-UC 1.2 TSS's own testbed
Mapping to services	S14-F aka SECaaS
Baseline	No baseline
Goals	Check the adequacy of KPIs of the security measures as defined below: 1. Latency at start induced by remote attestation < 1 second 2. Latency at start induced by confidentiality preservation < 3 seconds 3. Performance penalty caused by runtime verification and self-monitoring <10% 4. Energy consumption induced by WASM security enforcers < 10%
Means of verification, methodology, tools	Pending the results of the feasibility study as stated in KPI 1.4.1 Use representative WASM payload Similar techniques for verification as stated in KPIs 1.3

4.1.3. Sub-Use Case 1.3

KPI-ID	A-KPI 1.3-1.8
Name	Denial of credentials of devices running non-trusted software.
Description	Deny authentication and block service communication from devices with no or untrusted attestation.
Leading Partner	IMEC

KPI-ID	A-KPI 1.3-1.8
Validation sub-UC & testbed	Sub-UC 1.3 IMEC testbeds
Mapping to services	Attack Resilient/green orchestration Attack Resilient/green orchestration (Flocky/Trust-Edge)
Baseline	No baseline
Goal	Blocking of credentials from software on untrusted devices: 100%
Means of verification, methodology, tools	Status overview in Kubernetes (kubectl, kube API) - Nodes not allowed to join cluster - No deployments possible on untrusted nodes

KPI-ID	A-KPI 1.3-1.9
Name	Additional latency of attestation below target value.
Description	Startup/communication latency of devices must fall below defined values to avoid performance issues.
Leading Partner	IMEC
Validation sub-UC & testbed	Sub-UC 1.3 IMEC testbeds
Mapping to services	Attack Resilient/green orchestration (Flocky/Trust-Edge)
Baseline	Startup and communication latencies as measured in a default Kubernetes cluster (regular kubelet or Feather).
Goal	Additional latency: <2% Additional device deployment time: <1min
Means of verification, methodology, tools	Startup time as measured from initial contact with cluster (kubectl join) to operational status (node “READY” status). Communication latency to be determined by service logs and traffic monitoring.

4.2. Use Case 2

4.2.1. Sub-Use Case 2.1

In this part, the KPIs of sub-Use case 2.1 are presented. The missing ones are not relevant to that sub-Use case.

KPI-ID	KPI 2.1-2.1
Name	Jamming Attacks Detection & Mitigation
Description	The system must be able to accurately detect a jamming attack and distinguish it by a degradation of the signal due to another reason such as

KPI-ID	KPI 2.1-2.1
	the blockage of Line-of-Sight. Once a jamming attack has been detected, the mitigation of it should be feasible. The mitigation of the signal requires an extra antenna in the receiver that acts as shield in the jamming attack. The efficient mitigation of the jamming attack is strongly connected with the estimation of Angle-of-Arrival and phase difference between jamming attacker and legitimate user.
Leading Partner	CERTH
Validation sub-UC & testbed	Sub UC2.1, CERTH-testbed
Mapping to services	AI-based anti-jamming, ML-based MIMO.
Baseline	For the detection of jamming attack 99%. For mitigation, the baseline is the enhancement about 12 dB in the SNR. As concerns the identification, there is no baseline for error. However, it can be evaluated commonly with the SNR enhancement since a great error could limit the mitigation capability.
Goal	For the detection 99.9 % across all modulation schemes supported by IEEE 802.11 p. For jamming identification an error of 5%. For jamming mitigation, the goal an enhancement in SNR more than 20dB. Furthermore, the signal should be in format that can be demodulated efficiently across all modulation schemes and jamming types (constant, periodic, reactive).
Means of verification, methodology, tools	The jamming detection accuracy will be evaluated in CERTH SDR-based setup. The jamming identification and mitigation require string synchronization of the signals in legitimate receivers and its shield. It would be investigated if it is possible to be evaluated in SDR-setup. Otherwise, these components will be evaluated based on realistic simulation data.

KPI-ID	KPI 2.1-2.2
Name	Time needed to detect and prevent a jamming attack
Description	Jamming detection and mitigation accuracy is strongly connected with the required computational time for in-time notification.
Leading Partner	CERTH
Validation sub-UC & testbed	Sub UC2.1, CERTH-testbed
Mapping to services	AI-based anti-jamming, ML-based MIMO
Baseline	None
Goal	<4s

KPI-ID	KPI 2.1-2.2
Means of verification, methodology, tools	The jamming detection required computational time will be evaluated in CERTH SDR-based setup.

KPI-ID	KPI 2.1-2.5
Name	Throughput enhancement during jamming attack.
Description	The evaluation of the jamming mitigation mechanism performance needs a well-defined, widespread used metric that can express the QoS in the receiver before and after the jamming mitigation. A such is throughput. However, different metrics can be utilized such as BER or SNR.
Leading Partner	CERTH
Validation sub-UC & testbed	Sub UC2.1, CERTH-testbed
Mapping to services	AI-based anti-jamming, ML-based MIMO, AI-based RIS configuration
Baseline	None
Goal	SNR enhancement at least 20dB across all modulation schemes during all the types of jamming.
Means of verification, methodology, tools	The method will be developed in a simulation environment. The transfer in SDR-based setup will be investigated in terms of feasibility as an extra task.

4.2.2. Sub-Use Case 2.2

KPI-ID	KPI 2.2-2.1
Name	Detection and mitigation of jamming attacks
Description	Spectrum monitoring: Spectrum must be monitored to inspect the signals present at a given frequency and extract the key features (e.g. SINR) to perform the detection of jamming signals.
Leading Partner	GRADIANT
Validation sub-UC & testbed	Sub UC2.2, Gradiant 5G Lab
Mapping to services	AI-based anti-jamming
Baseline	None
Goal	Minimum accuracy of jamming detection: 90%
Means of verification,	A validation set of signals would be used to assess the accuracy of the AI algorithm in jamming classification

KPI-ID	KPI 2.2-2.1
methodology, tools	

KPI-ID	KPI 2.2-2.2
Name	Time needed to detect and prevent a jamming attack
Description	This will include the time required from the previous KPI 2.1 jamming detection.
Leading Partner	GRADIANT
Validation sub-UC & testbed	Sub UC2.2, Gradiant 5G Lab
Mapping to services	AI-based anti-jamming
Baseline	None
Goal	Maximum time needed to detect and prevent: 5s
Means of verification, methodology, tools	Time will be measured from the start of the receiving signal until a classification (either positive or negative) of it is done

KPI-ID	KPI 2.2-2.4
Name	Downtime prevented
Description	Time with the connection down due to jamming reduced due to the detection and reaction algorithm. It depends on the KPI 2.1, due to the detection phase being previous to the reaction phase
Leading Partner	GRADIANT
Validation sub-UC & testbed	Sub UC2.2, Gradiant 5G Lab
Mapping to services	AI-based anti-jamming
Baseline	No use of antijamming reaction
Goal	>20% of downtime prevented
Means of verification, methodology, tools	A sufficiently large amount of time is needed with the algorithm running, in order to statistically compare the amount of downtime reduced due to the service

KPI-ID	KPI 2.2-2.5
Name	Throughput enhancement during jamming attack
Description	Enhancement of the throughput while a jamming attack is happening
Leading Partner	GRADIANT

KPI-ID	KPI 2.2-2.5
Validation sub-UC & testbed	Sub UC2.2, Gradiant 5G Lab
Mapping to services	AI-based anti-jamming
Baseline	No use of antijamming reaction
Goal	>40% of the throughput
Means of verification, methodology, tools	Iperf or similar measurement tools will be used to compare the throughput of the communication during a jamming attack

KPI-ID	A-KPI 2.2-2.6
Name	Successful establishment of connectivity to avoid jammed channels/paths
Description	The reaction phase needs to reconnect using channels where the jamming attack is not present, so it also depends on the detection phase to locate those channels
Leading Partner	GRADIANT
Validation sub-UC & testbed	Sub UC2.2, Gradiant 5G Lab
Mapping to services	AI-based anti-jamming
Baseline	None
Goal	Reconnection on a frequency band with no jamming
Means of verification, methodology, tools	As the jammed frequencies will be selected, the service must reconnect to a frequency band not being attacked

4.2.3. Sub-Use Case 2.3

KPI-ID	KPI 2.3-2.1
Name	Jamming detection and mitigation
Description	Jamming is detected by monitoring signal parameters such as RSSI and mitigated using adaptive MCS algorithm
Leading Partner	ISRD
Validation sub-UC & testbed	Sub UC2.3, ISRD Testbed
Mapping to services	AI-based anti-jamming
Baseline	Link Adaptation Algorithm based on CSI and HARQ

KPI-ID	KPI 2.3-2.1
Goal	90% jamming detection
Means of verification, methodology, tools	The jamming detection accuracy will be evaluated in ISRD lab setup using a validation signal.

KPI-ID	KPI 2.3-2.2
Name	Jamming detection time
Description	Time needed to detect a jamming attack
Leading Partner	ISRD
Validation sub-UC & testbed	Sub UC2.3, ISRD Testbed
Mapping to services	AI-based anti-jamming
Baseline	No baseline
Goal	<5s
Means of verification, methodology, tools	Time will be measured from the start of the receiving signal until jamming attack is detected

KPI-ID	KPI 2.3-2.3
Name	Jamming recovery time
Description	Time needed to recover from a jamming attack
Leading Partner	ISRD
Validation sub-UC & testbed	Sub UC2.3, ISRD Testbed
Mapping to services	AI-based anti-jamming
Baseline	No baseline
Goal	<5s
Means of verification, methodology, tools	Time will be measured from the start of the receiving signal until jamming attack is detected

KPI-ID	KPI 2.3-2.4
Name	Downtime prevented
Description	Time with the connection down due to jamming before the connection is re-established.

KPI-ID	KPI 2.3-2.4
Leading Partner	ISRD
Validation sub-UC & testbed	Sub UC2.3, ISRD Testbed
Mapping to services	AI-based anti-jamming
Baseline	Link Adaptation Algorithm based on CSI and HARQ
Goal	<20% downtime
Means of verification, methodology, tools	Mean downtime calculated at ISRD lab setup during long experiment time

KPI-ID	KPI 2.3-2.5
Name	Throughput enhancement during jamming attack
Description	Enhancement of the throughput while a jamming attack is happening
Leading Partner	ISRD
Validation sub-UC & testbed	Sub UC2.3, ISRD Testbed
Mapping to services	AI-based anti-jamming
Baseline	No use of antijamming reaction
Goal	>40% of the throughput
Means of verification, methodology, tools	Iperf or similar measurement tools will be used to compare the throughput of the communication during a jamming attack

4.2.4. Sub-Use Case 2.4

KPI-ID	A-KPI 2.4-2.7
Name	Key Generation Length
Description	Generation of 128-bit keys to ensure strong encryption for secure communications, providing the necessary cryptographic strength for applying AES128.
Leading Partner	GRAD
Validation sub-UC & testbed	Sub UC2.4 & Gradiant 5G Lab
Mapping to services	Key Generation Service and Security Validation Service
Baseline	128 bits

KPI-ID	A-KPI 2.4-2.7
Goal	128 bits
Means of verification, methodology, tools	Direct measurement of key output for different conditions. Compare the Key Generation Length with the expected value.

KPI-ID	A-KPI 2.4-2.8
Name	NIST Random Test Compliance
Description	The generated keys will comply with the NIST random test suite, achieving a P-value greater than 0.01 to ensure optimal randomness and security in the key generation process.
Leading Partner	GRAD
Validation sub-UC & testbed	Sub UC2.4 & Gradiant 5G Lab
Mapping to services	Security Validation Service
Baseline	p-value > 0.01
Goal	p-value > 0.01
Means of verification, methodology, tools	Through randomness evaluations on the generated keys and confirm they meet the p-value criterion with the NIST Test suite.

KPI-ID	A-KPI 2.4-2.9
Name	Key Generation Rate
Description	The rate of key generation will increase in proportion to the quality of the physical channel, ensuring efficient key production up to an optimal threshold, adapting dynamically to the channel conditions.
Leading Partner	GRAD
Validation sub-UC & testbed	Sub UC2.4 & Gradiant 5G Lab
Mapping to services	Key Generation Service and Characteristics Extraction Service
Baseline	KGR between 70% and 80% for FDD in 2.4-2.5GHz
Goal	KGR > 90% for TDD & FDD
Means of verification, methodology, tools	Measurement of channel metrics and measurement with the output from the AI module. Compare key generation rates under FDD and TDD scenarios and assess the impact of AI optimizations.

KPI-ID	KPI 2.4-2.4
Name	Downtime Prevention
Description	Minimize traditional downtime and delays during the key generation process by ensuring that session re-authentication performed within less time.
Leading Partner	GRAD
Validation sub-UC & testbed	Sub UC2.4 & Gradiant 5G Lab
Mapping to services	Key Generation Service and Characteristics Extraction Service
Baseline	Latency of 10–20 ms in traditional systems
Goal	Achieve a significant reduction in reauthentication latency.
Means of verification, methodology, tools	Measure the total authentication latency from the initiation of the resumption channel to the establishment of the secure session.

4.3. Use Case 3

4.3.1. Sub-Use Case 3.1

KPI-ID	KPI 3.1-3.1
Name	Mean Time to Detect (MTTD)
Description	Average time taken to identify a security threat
Leading Partner	MONT
Validation sub-UC & testbed	UC#3.1, MONT 5G testbed with MMT monitoring framework and IoT wireless sniffer
Mapping to services	Security monitoring
Baseline	None
Goal	Quantitative
Means of verification, methodology, tools	Performing several use case scenarios so that the KPIs can be measured using the testbed and emulation techniques. Mean detection time less than 5 minutes for ML-based predictions and 10 ms for Montimage Monitoring Tool (MMT) framework rules.

KPI-ID	KPI 3.1-3.2
Name	Number of False Positives (FP)
Description	Incorrect threat alerts

KPI-ID	KPI 3.1-3.2
Leading Partner	MONT
Validation sub-UC & testbed	UC#3.1, MONT 5G testbed with MMT monitoring framework and IoT wireless sniffer
Mapping to services	Security monitoring
Baseline	None
Goal	Quantitative
Means of verification, methodology, tools	Performing several use case scenarios so that the KPIs can be measured using the testbed and emulation techniques. FP rates are less than 1%.

KPI-ID	KPI 3.1-3.3
Name	Number of False Negatives (FN)
Description	Missed threats
Leading Partner	MONT
Validation sub-UC & testbed	UC#3.1, MONT 5G testbed with MMT monitoring framework and IoT wireless sniffer
Mapping to services	Security monitoring
Baseline	None
Goal	Quantitative
Means of verification, methodology, tools	Performing several use case scenarios so that the KPIs can be measured using the testbed and emulation techniques. FN rates are less than 1%.

KPI-ID	KPI 3.1-3.4
Name	Packet Loss Ratio (PLR)
Description	Packet loss by security monitoring probes
Leading Partner	MONT
Validation sub-UC & testbed	UC#3.1, MONT 5G testbed with MMT monitoring framework and IoT wireless sniffer
Mapping to services	Security monitoring
Baseline	None
Goal	Quantitative

KPI-ID	KPI 3.1-3.4
Means of verification, methodology, tools	Performing several use case scenarios so that the KPIs can be measured using the testbed and emulation techniques. Packet Loss Ratio (PLR) less than 0.001%

KPI-ID	KPI 3.1-3.5
Name	Mean Time to Resolve (MTTR)
Description	Average time to neutralize a detected threat
Leading Partner	MONT
Validation sub-UC & testbed	UC#3.1, MONT 5G testbed with MMT monitoring framework and IoT wireless sniffer
Mapping to services	Security monitoring
Baseline	None
Goal	Quantitative
Means of verification, methodology, tools	Performing several use case scenarios so that the KPIs can be measured using the testbed and emulation techniques. Mean resolution time less than 10 minutes.

4.3.2. Sub-Use Case 3.2

KPI-ID	A-KPI 3.2-3.6
Name	Impact on QoS by AI-DoS evaluation tool
Description	This KPI measures the effects of the AI-DOS created attack against the system.
Leading Partner	CERTH
Validation sub-UC & testbed	UC#3.2 CERTH testbed
Mapping to services	AI driven penetration Testing
Baseline	Results from KPI-3.2-1
Goal	The AI-DoS will be considered successful if it can effectively reduce the QoS by more than 70% in the evaluated 5G/6G service provided by CERTH.
Means of verification, methodology, tools	We will measure two KPIs to see how QoS is affected by the AI-DoS attack: UE Throughput (measured in MBPS) and E2E round-trip time latency between UE and the Core (measured in ms). These will be measured in normal network conditions and during an attack.

KPI-ID	A-KPI 3.2-3.7
Name	Comparison of results between AI-DoS and other QoS assessment tools to determine the most effective tool.
Description	This KPI will report the comparison of results between AI-DoS and other QoS assessment tools to determine the most effective tool.
Leading Partner	CERTH
Validation sub-UC & testbed	UC#3.2 CERTH testbed
Mapping to services	AI driven penetration Testing
Baseline	None
Goal	This is measured to create a comparison baseline for KPI-3.2-2.
Means of verification, methodology, tools	We will measure two KPIs to see how QoS is affected by tools that perform DoS attack: UE Throughput (measured in MBPS) and E2E round-trip time latency between UE and the Core (measured in ms). These will be measured in normal network conditions and during an attack. We will compare these measurements with KPI-3.2.2 results.

KPI-ID	A-KPI 3.2-3.8
Name	Perform a vulnerability report regarding DoS resilience on 5G/6G components.
Description	AI-DoS will have to provide detailed information about which strategy it implemented for the penetration Testing
Leading Partner	CERTH
Validation sub-UC & testbed	UC#3.2 CERTH testbed
Mapping to services	AI driven penetration Testing
Baseline	Results from KPI-3.2-1
Goal	Yes/No (Binary): AI driven penetration Testing also produces a detailed report that describes in detail the strategy it implemented.
Means of verification, methodology, tools	Report produced

4.3.3. Sub-Use Case 3.3

KPI-ID	A-KPI 3.3-3.9
Name	Mean Time to Detect (MTTD)
Description	The time required to detect an anomaly attack against the security and trust management system
Leading Partner	ELTE
Validation sub-UC & testbed	Sub-UC#3.3 ELTE testbed
Mapping to services	S3-Security by Design Orchestration S3-S-C2 - E2E Security Management
Baseline	None
Goal	<10ms for not ML-based rules
Means of verification, methodology, tools	The verification approach involves simulating attacks including impersonation attacks using self-generated Python code alongside the Foundry Blockchain. In the impersonation scenario, an attacker attempts to access the IoT service provider using either outdated or randomly generated credentials. The service provider cross-verifies trust token through the blockchain ledger, and if no valid trust anchor is found, access is immediately denied.

KPI-ID	A-KPI 3.3-3.10
Name	Number of False Positives (FP)
Description	The percentage of legitimate entities incorrectly flagged as threats during security and trust management
Leading Partner	ELTE
Validation sub-UC & testbed	Sub-UC#3.3 ELTE testbed
Mapping to services	S3-Security by Design Orchestration S3-S-C2 - E2E Security Management
Baseline	None
Goal	< 1%
Means of verification, methodology, tools	To assess FP rate, Open5GS, UERANSIM, and custom Solidity smart contracts are employed as core tools to simulate and monitor authentication of legitimate UEs. It involves running controlled experiments where authenticated devices attempt to access services. The means of verification focus on analyzing blockchain logs and smart contract decisions to identify cases where legitimate devices are mistakenly flagged as threats.

KPI-ID	A-KPI 3.3-3.11
Name	Number of False Negatives (FN)
Description	The percentage of malicious entities incorrectly flagged as benign during security and trust management
Leading Partner	ELTE
Validation sub-UC & testbed	Sub-UC#3.3 ELTE testbed
Mapping to services	S3-Security by Design Orchestration S3-S-C2 - E2E Security Management
Baseline	None
Goal	<1%
Means of verification, methodology, tools	The verification of this KPI involves monitoring authentication attempts from both legitimate and malicious UEs simulated using UERANSIM, while tracking whether any unauthorized access is incorrectly flagged as benign. The means of verification include analyzing records and authentication outcomes to measure the FN rate, ensuring that the trust management system accurately identifies threats

KPI-ID	A-KPI 3.3-3.12
Name	Trust Establishment Time (TET)
Description	Measures the average time required to establish trust between devices in a decentralized manner
Leading Partner	ELTE
Validation sub-UC & testbed	Sub-UC#3.3 ELTE testbed
Mapping to services	S3-Security by Design Orchestration S3-S-C2 - E2E Security Management
Baseline	5G standard authentication and authorization
Goal	<1 s
Means of verification, methodology, tools	To verify the trust establishment time, Open5GS is utilized to simulate the 5G core and UERANSIM for IoT node, combined with custom Solidity smart contracts deployed on a blockchain using Foundry. A token-based access system is introduced to reduce blockchain interaction overhead during repeated authentications. The means of verification focus on validating decentralized trust establishment

4.4. Use Case 4

4.4.1. Sub-Use Case 4.1

KPI-ID	KPI 4.1-4.1.1
Name	DFE processing latency
Description	Delay experienced by flows under DFE program offloaded in programmable switch/NIC.
Leading Partner	CNIT
Validation sub-UC & testbed	Sub UC4.1 & CNIT ARNO Labs
Mapping to services	AI-based behavioral analysis
Baseline	Traditional systems – few ms
Goal	Achieve <50us with up to 10k different flow rules
Means of verification, methodology, tools	Measure the transit time of a flow packet. Tools: traffic generators and analyzers

KPI-ID	KPI 4.1-4.1.2
Name	DFE computational efficiency
Description	Processing for retrieving specific traffic features
Leading Partner	CNIT
Validation sub-UC & testbed	Sub UC4.1 & CNIT ARNO Labs
Mapping to services	AI-based behavioral analysis, P4 based Analytics
Baseline	Traditional systems – raw in-band/postcard telemetry with packet and header mirroring – 10-16 seconds [7]
Goal	Achieve 50% improved efficiency (reduce computation time of around 33%)
Means of verification, methodology, tools	Measure the time needed to perform the feature extraction. Tools: traffic generators and analyzers

KPI-ID	KPI 4.1-4.1.3
Name	DFE power consumption
Description	Evaluate the power needed for offloaded solutions with respect to software-based feature selection and extraction

KPI-ID	KPI 4.1-4.1.3
Leading Partner	CNIT
Validation sub-UC & testbed	Sub UC4.1 & CNIT ARNO Labs
Mapping to services	AI-based behavioral analysis, Energy efficient orchestration
Baseline	Traditional systems – software switches and compute node programs – Compute Node Processing power 80-100W
Goal	Achieve 20% power consumption reduction
Means of verification, methodology, tools	Estimation with device sensors. Tools: internal sensors /power meters

KPI-ID	KPI 4.1-4.1.4
Name	WAI latency
Description	Evaluate the latency introduced by the WAI component (data plane)
Leading Partner	CNIT
Validation sub-UC & testbed	Sub UC4.1 & CNIT ARNO Labs
Mapping to services	AI-based behavioral analysis
Baseline	-
Goal	Hardware backends<10us, software-based backends <100 us
Means of verification, methodology, tools	Traffic generators and analyzers

KPI-ID	KPI 4.1-4.1.5
Name	Global DFE+WAI solution
Description	Global Power Consumption
Leading Partner	CNIT
Validation sub-UC & testbed	Sub UC4.1 & CNIT ARNO Labs
Mapping to services	AI-based behavioral analysis, Energy efficient orchestration
Baseline	Outsourced classical AI systems running in the cloud, switch between 80 and 200W, compute node with GPU between 500 and 800W
Goal	50% reduction using hardware accellerations and network devices avoiding GPUs unless absolutely necessary

KPI-ID	KPI 4.1-4.1.5
Means of verification, methodology, tools	Estimation with device sensors. Tools: internal sensors /power meters

4.4.2. Sub-Use Case 4.2

KPI-ID	KPI 4.2-4.2.1
Name	Energy Efficiency Improvement
Description	The AI slicing framework should reduce the overall energy consumption compared to centralized AI inference systems.
Leading Partner	ELTE
Validation sub-UC & testbed	Sub-UC4.2, ELTE testbed
Mapping to services	AI-based behavioral analysis
Baseline	Centralized inference using full AI models on high-power compute nodes. Assuming the typical power draw per GPU about 400 Watts a single-node setup with 4 GPUs sets the baseline at 1.6 kW for the GPUs only. If we include the additional overhead (CPUs, cooling, etc) we can estimate 2kW for the centralized inference.
Goal	< 80% power consumption compared to centralized solution
Means of verification, methodology, tools	Device energy usage comparison with centralized baseline under similar loads

KPI-ID	KPI 4.2-4.2.2
Name	Latency Reduction
Description	The deployment of AI slices near the data source should reduce end-to-end latency significantly.
Leading Partner	ELTE
Validation sub-UC & testbed	Sub-UC4.2, ELTE testbed
Mapping to services	AI-based behavioral analysis

KPI-ID	KPI 4.2-4.2.2
Baseline	No baseline
Goal	< 1 ms end-to-end latency
Means of verification, methodology, tools	Packet timestamping, in-band telemetry, comparison of inference delay between centralized and sliced deployments

KPI-ID	KPI 4.2-4.2.3
Name	Resource Utilization
Description	The system should offload at least 50% of AI model components to underutilized network resources.
Leading Partner	ELTE
Validation sub-UC & testbed	Sub-UC4.2, ELTE testbed
Mapping to services	Distributed Federated Learning across the Continuum
Baseline	No baseline
Goal	> 50% of AI model executed on programmable network hardware
Means of verification, methodology, tools	Hardware utilization logs, deployment reports, resource monitoring

KPI-ID	KPI 4.2-4.2.4
Name	AI Model Accuracy Maintenance
Description	Despite slicing and deployment across the network, the AI model must retain at least 90% of its original accuracy.
Leading Partner	ELTE
Validation sub-UC & testbed	Sub-UC4.2, ELTE testbed
Mapping to services	Distributed Federated Learning accross the Continuum
Baseline	Centralized AI model full-accuracy benchmark with a typical F1 score 0.95–0.99
Goal	> 0.9 × centralized model F1-score

KPI-ID	KPI 4.2-4.2.4
Means of verification, methodology, tools	Standard AI evaluation (precision/recall/F1), comparison of outputs between centralized and sliced models

KPI-ID	KPI 4.2-4.2.5
Name	Dynamic Reconfiguration Time
Description	The system must detect degraded performance and dynamically reconfigure AI slices within a few seconds.
Leading Partner	ELTE
Validation sub-UC & testbed	Sub-UC4.2, ELTE testbed
Mapping to services	AI-based behavioral analysis
Baseline	No baseline
Goal	< 5 seconds reconfiguration time
Means of verification, methodology, tools	Controller logs

4.4.3. Sub-Use Case 4.3

KPI-ID	A-KPI 4.3-4.6
Name	Jamming/adversary attacks mitigation
Description	This measures the percentage of unjammed signal recovered after mitigating the jamming attack.
Leading Partner	CERTH
Validation sub-UC & testbed	UC#4.3 CERTH testbed
Mapping to services	AI-based anti-jamming
Baseline	-
Goal	At least 80% accuracy in unjammed signal recovery
Means of verification,	The jamming identification and mitigation require string synchronization of the signals in legitimate receivers and its shield. It would be investigated

KPI-ID	A-KPI 4.3-4.6
methodology, tools	if it is possible to be evaluated in SDR-setup. Otherwise, these components will be evaluated based on realistic simulation data.

KPI-ID	A-KPI 4.3-4.7
Name	Time needed to prevent or mitigate a jamming/adversary attack via AI/ML frequency and protocol switching
Description	This KPI measures the time needed to prevent or mitigate a jamming/adversary attack via applying AI/ML frequency and protocol switching based approaches.
Leading Partner	CERTH
Validation sub-UC & testbed	UC#4.3 CERTH testbed
Mapping to services	AI-based anti-jamming
Baseline	-
Goal	< 5s
Means of verification, methodology, tools	The jamming detection required computational time will be evaluated in CERTH SDR-based setup.

KPI-ID	A-KPI 4.3-4.8
Name	Time needed to recover from a jamming attack
Description	The time needed by the system to recover from a jamming attack
Leading Partner	CERTH
Validation sub-UC & testbed	UC#4.3 CERTH testbed
Mapping to services	AI-based anti-jamming
Baseline	-
Goal	< 10s
Means of verification, methodology, tools	The jamming detection required computational time will be evaluated in CERTH SDR-based setup

KPI-ID	A-KPI 4.3-4.9
Name	Downtime reduction
Description	Reduced downtime values of the system caused by an attack, after the application of the proposed cybersecurity methods application.

KPI-ID	A-KPI 4.3-4.9
Leading Partner	CERTH
Validation sub-UC & testbed	UC#4.3 CERTH testbed
Mapping to services	AI-based anti-jamming Attack detection and mitigation
Baseline	Downtime of the system caused by an attack
Goal	At least 20% improvement
Means of verification, methodology, tools	The jamming detection required computational time will be evaluated in CERTH SDR-based setup

KPI-ID	A-KPI 4.3-4.10
Name	Throughput increase
Description	A jamming/adversary attack cannot be immediately mitigated. However, applying the proposed countermeasures is expected to lead to improved throughput values during the attack.
Leading Partner	CERTH
Validation sub-UC & testbed	UC#4.3 CERTH testbed
Mapping to services	AI-based anti-jamming
Baseline	-
Goal	At least 40 %, expected throughput improvement during jamming/adversary attack.
Means of verification, methodology, tools	The jamming detection required computational time will be evaluated in CERTH SDR-based setup

4.4.4. Sub-Use Case 4.4

KPI-ID	KPI 4.4-4.4
Name	Probability of DoS Attack Detection
Description	The KPI measures the likelihood that the system correctly identifies a Denial of Service (DoS) attack. It reflects the effectiveness of detection mechanisms in flagging malicious traffic patterns.
Leading Partner	CERTH
Validation sub-UC & testbed	UC#4.3 CERTH testbed

KPI-ID	KPI 4.4-4.4
Mapping to services	AI-based Intrusion Detection
Baseline	>70%
Goal	>80%
Means of verification, methodology, tools	Extensive experimentation on the CERTH testbed will be performed under different attack scenarios in order to measure how often an attack is identified.

KPI-ID	KPI 4.4-4.5
Name	Probability of false detection
Description	The KPI measures how often legitimate traffic is incorrectly flagged as a DoS attack. It indicates the rate of false positives generated by the detection system.
Leading Partner	CERTH
Validation sub-UC & testbed	UC#4.3 CERTH testbed
Mapping to services	Attack detection AI-based Intrusion Detection
Baseline	<15%
Goal	<10%
Means of verification, methodology, tools	AI-based attack execution and extensive experimentation on the CERTH testbed including both attack and stress conditions to validate that the IDS system identifies attacks and to minimize false positives.

4.4.5. Sub-Use Case 4.5

KPI-ID	A-KPI 4.5-4.11
Name	Mean Time to implement the MTD action (MTID)
Description	Every MTD action has a time it requires to complete its enforcement on NFs (whether CNFs or VNFs). This time does not correspond to an NF service disruption/downtime (which is instead defined in the next KPI) but it defines the time in which no other operation can be performed on the NF.
Leading Partner	ZHAW
Validation sub-UC & testbed	Sub-UC 4.5 PATRAS 5G testbed
Mapping to services	S10-S: AI-based MTD

KPI-ID	A-KPI 4.5-4.11
Baseline	No baseline, because this KPI is only about our MTD solution itself and cannot exist without it
Goal	Max MTID < 2 minutes
Means of verification, methodology, tools	Simulation of MTD actions on the testbed measuring the relay of the MTD action to the enforcer: MTID = [(The time when MTD Controller receives an MTD action) - (The time when an MTD action is determined by the Strategy Optimizer)] in seconds

KPI-ID	A-KPI 4.5-4.12
Name	Worst-case MTD service disruption (WMSD)
Description	WMSD defines the maximum service downtime of a NF (CNF/VNF) due to an ongoing MTD action.
Leading Partner	ZHAW
Validation sub-UC & testbed	Sub-UC 4.5 PATRAS 5G testbed
Mapping to services	S10-S: AI-based MTD
Baseline	No baseline, because this KPI is only about our MTD solution itself and cannot exist without it
Goal	WMSD < 20 seconds
Means of verification, methodology, tools	Simulation of MTD actions on the testbed. Measuring the times when the service gets cut off and when the service goes back up: WMSD = [(End of MTD action downtime) - (Start of MTD action downtime)] in seconds

KPI-ID	A-KPI 4.5-4.13
Name	MTD action cost overhead (MACO)
Description	MACO defines the cost of MTD actions based on the cloud prices of used CPU, RAM and disk resources.
Leading Partner	ZHAW
Validation sub-UC & testbed	Sub-UC 4.5 PATRAS 5G testbed
Mapping to services	S10-S: AI-based MTD
Baseline	No baseline, because this KPI is only about our MTD solution itself and cannot exist without it
Goal	MACO < 100% increase

KPI-ID	A-KPI 4.5-4.13
Means of verification, methodology, tools	Simulation of MTD actions on the testbed Measuring the resource usage continuously during an MTD action: MACO (CPU/RAM/Storage) = 100 * { [Max (CPU/RAM/Storage) during the MTD Action] - [Mean(CPU/RAM/Storage) before the MTD Action] } / [Mean(CPU/RAM/Storage) before the MTD Action] (%)

KPI-ID	A-KPI 4.5-4.14
Name	MTD-induced green energy consumption [MGEC]
Description	Green energy consumption based on the 0-net index of destination's hosting infrastructure.
Leading Partner	ZHAW
Validation sub-UC & testbed	Sub-UC 4.5 PATRAS 5G testbed
Mapping to services	S10-S: AI-based MTD
Baseline	The baseline shall be defined as the MGEC before applying our MTD solution
Goal	MGEC > 5% (highly depending on the conditions)
Means of verification, methodology, tools	Simulation of HW with different carbon intensities / fossil-green ratio, measuring the energy consumption using both fossil and green energy: MGEC = [(Energy Consumption using Green sources after MTD action / Total Energy Consumption after MTD action) - (Energy Consumption using Green sources before MTD action / Total Energy Consumption before MTD action)] (%)

KPI-ID	A-KPI 4.5-4.15
Name	Protection gain of an MTD policy
Description	This evaluates the proactive MTD security measured based on risk and threat analysis done on the network estimating exploitability and attack impacts on VNFs/CNFs and how this is reduced with MTD.
Leading Partner	ZHAW
Validation sub-UC & testbed	Sub-UC 4.5 PATRAS 5G testbed
Mapping to services	S10-S: AI-based MTD
Baseline	The baseline shall be defined as the LSE without the MTD solution

KPI-ID	A-KPI 4.5-4.15
Goal	Worst case: Up to 5% reduction in the Likelihood of Successful Exploitation (LSE) Mean case: Up to 10% reduction in LSE
Means of verification, methodology, tools	Periodical threat and risk assessments on VNFs/CNFs with vulnerability scans, and CVSS standards scores. D-LSE = (LSE before MTD action - LSE after MTD action) % LSE = Likelihood of Successful Exploitation D-LSE = Decrease in Likelihood of Successful Exploitation

KPI-ID	A-KPI 4.54.16
Name	Mean decision time for MTD action [MDTA]
Description	This measures the time required by the ML model of the MTD Optimizer to decide on an MTD action to perform given the observation of the network state.
Leading Partner	ZHAW
Validation sub-UC & testbed	Sub-UC 4.5 PATRAS 5G testbed
Mapping to services	S10-S: AI-based MTD
Baseline	No baseline, because this KPI is only about our MTD solution itself and cannot exist without it
Goal	**Proactive** case: MDTA < 500 ms **Reactive** case (including the time elapsed by network probes): MDTA < 5s
Means of verification, methodology, tools	**Proactive** case: Time elapsed from receiving the MOMDP observation to the ML model providing an MTD action. **Reactive** case: Time elapsed from an attack being detected to the MTD action being selected and enforced. MDTA = [(The time when an MTD action is determined by the Strategy Optimizer) - (The time of the event: E_P or E_R)] in seconds E_P = Observation of MOMDP state E_R = Detection of an attack

KPI-ID	KPI - 4.5 - A-KPI 4.17
Name	Decision Explainability for MTD [DEFM]
Description	This measures the time required by the ML model of the MTD Optimizer to decide on an MTD action to perform given the observation of the network state.
Leading Partner	ZHAW

KPI-ID	KPI - 4.5 - A-KPI 4.17
Validation sub-UC & testbed	Sub-UC 4.5 PATRAS 5G testbed
Mapping to services	S10-S: AI-based MTD
Baseline	No baseline, because this KPI is only about our MTD solution itself and cannot exist without it
Goal	Obtaining human-readable explanation indicating the objective/reasoning of the MTD decision
Means of verification, methodology, tools	E_R = Detection of an attackReactive case: Time elapsed from an attack being detected to the MTD action being selected and enforced. For each decision made by the MTD Strategy Optimizer, a humanly interpretable explanation should also be provided, and these responses will be evaluated based on correctness and rationality.

4.4.6. Sub-Use Case 4.6

KPI-ID	KPI 4.6-4.3
Name	Software Control Flow monitoring specification (feasibility study)
Description	As a result of the initial feasibility study of a software control flow monitoring, a specification document will detail its technical definition. The use case is of low TRL and starts with a feasibility study supported by TSS and other partners (i.e., MONT, CNIT) having expertise in both AI/ML and DoS attacks mitigation techniques. TSS's area relates to design and develop a **self-contained** workload performance monitoring method, which extracts time series improving DoS detection decreasing the false positives and negatives. The feasibility study will be worked out to validate the relevance of producing these novel self-contained performance metrics for DoS detection, to identify the general method setting workflow to produce these metrics and finally to assess the performance penalty induced by the metrics collection on-the-fly. This feasibility study will be based on preliminary design elements delivered in D3.5.
Leading Partner	TSS
Validation sub-UC & testbed	Sub-UC 4.6 TSS's own testbed
Mapping to services	S14-F aka SECaaS, branch for self-contained monitoring
Baseline	There is no baseline for an unprotected payload; the latency before its start is non-existent.

KPI-ID	KPI 4.6-4.3
Goal	1. Validate or invalidate the use of self-contained performance monitoring. 2. Define the different types of control flow extracted metrics (e.g., call blocks call frequency, code block execution time, other) and their usefulness, accuracy and costs. 3. Validate if these metrics can be used to discriminate against various causes of performance variation. 4. Assess the relevance of the solution with respect to other existing solutions. 5. Assess how these metrics can be supportive for CNIT's GNN ensemble-based DoS detection and MONT's ML or FL based DDoS
Means of verification, methodology, tools	Internal review

KPI-ID	KPI 4.6-4.4
Name	Probability of detection of DoS attack
Description	This KPI is conditioned by the outcome of KPI 4.3 above If DoS attack detection can be worked out, the KPI relates to the probablity of detection. At the current stage, we believe that the method shall be considered as a detection booster, improving pre-existing method and not as a new DoS detection method per-se.
Leading Partner	TSS
Validation sub-UC & testbed	Sub-UC 1.2 TSS's own testbed
Mapping to services	S14-F aka SECaaS
Baseline	The baseline shall be defined as one of the two considered AI-based methods for DoS detection w/o using the novel metrics.
Goal	Shall be defined as "any substantial gain" brought by the novel metrics on one of the two AI- based DoS detection.
Means of verification, methodology, tools	This KPI will be refined during the execution of the work. The methodology shall be considered with consideration of the type of DoS attack detection considered (ie, MONT, CNIT) as the method will be supportive to the detection method used there.

KPI-ID	KPI 4.6-4.5
Name	Probability of false detection of DoS attack
Description	This KPI is conditioned by the outcome of KPI 4.3 above

KPI-ID	KPI 4.6-4.5
	If DoS attack detection can be worked out, the KPI relates to the probability of false alarm (or DoS false positive rate). At the current stage, we believe that the method shall be considered as a detection booster, improving pre-existing method and not as a new DoS detection method per-se.
Leading Partner	TSS
Validation sub-UC & testbed	Sub-UC 1.2 TSS's own testbed
Mapping to services	S14-F aka SECaaS
Baseline	The baseline value shall be defined as one of the two considered AI-based methods for DoS detection w/o using the novel metrics
Goal	The target value shall be defined as "any substantial gain" brought by the novel metrics on one of the two AI- based DoS detection.
Means of verification, methodology, tools	The relevant testbed and means of verification will result from the feasibility study as defined in KPI 4.3

5. Requirements Evaluation

This section describes the structured approach for evaluating requirements across sub-use cases, ensuring alignment with system, functional, and non-functional needs. Each requirement is documented in a standardized template (see Table 9), which captures key attributes such as priority, validation context, and verification methodology. Use Case requirements were first presented in deliverable D2.2, and service-related requirements from deliverable D2.3. The same naming convention is adopted here for the ones derived from D2.3.

The requirement template includes the following fields: Requirement ID (a unique identifier in the format `REQ <subUC>-<number>`), Name and Description of the requirement, Leading Partner, and Type (classified as SYSTEM, FUNCTIONAL, NON-FUNCTIONAL, etc.). Each requirement is assigned a priority (MUST, SHOULD, or MAY) and mapped to the relevant sub-use case and testbed for validation. Additionally, the template specifies the NATWORK services involved and the means of verification (e.g., simulation tools, lab tests, or benchmarks).

The measurable requirements follow the acceptance criteria defined, ensuring traceability and testability. This structured approach enables systematic validation across different testbeds and use cases while maintaining consistency with project deliverables.

Table 9: Requirements template

Req-ID	REQ <subUC>-<number>
Name	
Description	
Leading Partner	
Type	<SYSTEM, FUNCTIONAL, NON-FUNCTIONAL>
Priority	*<MUST, SHOULD or MAY>*
Validation sub-UC & testbed	*Indicate sub use case identifier and testbed where the KPI is validated*
Mapping to services	*Indicate the services concerned (refer to D2.3)*
Means of verification, methodology, tools	*How the requirement was assessed or what tools were used to verify it*

5.1. Use Case 1

5.1.1. Sub-Use Case 1.1

Req-ID	REQ 1.1-1
Name	DoSt Attack Detection and Demonstration
Description	Simulate and demonstrate Denial of Sustainability (DoSt) attacks on 6G slices using HTTP-based oscillating demand to trigger continuous scaling of Kubernetes containers.
Leading Partner	UEssex
Type	SYSTEM, NON-FUNCTIONAL
Priority	SHOULD
Validation sub-UC & testbed	UC1.1 – NCL testbed (UEssex)
Mapping to services	Secure-by-design orchestration service
Means of verification, methodology, tools	HTTP load generation tools (custom scripts or traffic generators), monitoring through Prometheus and ONOS, and log analysis from FORK orchestration layer to verify detection and response.

Req-ID	REQ 1.1-2
Name	Real-time CTI Exchange
Description	Enable decentralised and adaptive CTI sharing between clusters. The solution must collect vulnerability data from local scanners and selectively share it using standardised formats (e.g., STIX/TAXII), based on security policies.
Leading Partner	UEssex
Type	SYSTEM, NON-FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	UC1.1 – NCL testbed (UEssex)
Mapping to services	Security-compliant Slice Management
Means of verification, methodology, tools	Integration of CTI agents in each cluster; verification through examining STIX/TAXII exchanges, adaptive filtering logic tests, and runtime validation of shared CTI data.

Req-ID	REQ 1.1-3
Name	Adaptive Information Sharing in CTI
Description	Control the amount of CTI data shared with decision making mechanisms, dynamically adjusting based on vulnerability context and security requirements, avoiding sensitive/confidential info exposure.
Leading Partner	UEssex
Type	FUNCTIONAL
Priority	SHOULD
Validation sub-UC & testbed	UC1.1 – NCL testbed (UEssex)
Mapping to services	Security-compliant Slice Management
Means of verification, methodology, tools	Policy-based dynamic filtering validation using CTI logs and control parameters; monitoring STIX/TAXII message structure and comparing exposed vs. total data parts in each message.

Req-ID	REQ 1.1-4
Name	Secure-by-design Orchestration Decisions based on Cluster Hygiene Assessment
Description	The orchestration system must take into account CTI-based vulnerability assessments and hygiene scores to guide deployment decisions. High-security applications should be placed in clusters with higher trustworthiness and lower risk exposure.
Leading Partner	UEssex
Type	FUNCTIONAL
Priority	SHOULD
Validation sub-UC & testbed	UC1.1 – NCL testbed (UEssex)
Mapping to services	Secure-by-design orchestration service
Means of verification, methodology, tools	Analysis of orchestration logs and hygiene score reports; test deployments compared against cluster risk levels; verification through placement audit trails and CTI integration validation.

Req-ID	REQ 1.1-5
Name	Energy Efficiency Optimisation
Description	The system should optimize energy usage across orchestration and slice management functions. This includes minimizing unnecessary scaling,

Req-ID	REQ 1.1-5
	intelligently placing workloads, and adapting resource allocation to reduce energy consumption while maintaining performance.
Leading Partner	UEssex
Type	NON-FUNCTIONAL
Priority	SHOULD
Validation sub-UC & testbed	UC1.1 – NCL testbed (UEssex)
Mapping to services	Secure-by-design orchestration service
Means of verification, methodology, tools	Resource usage monitored via Prometheus; CPU utilization benchmarks under varying load; comparison of orchestration behaviors with and without energy-aware policies.

Req-ID	REQ 1.1-6
Name	AI-Driven Security Enhancements
Description	Integrate AI-based techniques into the orchestration and security layers to support real-time monitoring, anomaly detection, and vulnerability analysis. These AI models should enhance threat visibility and decision-making in the 6G core and edge environments.
Leading Partner	Uessex
Type	FUNCTIONAL
Priority	*SHOULD*
Validation sub-UC & testbed	UC1.1 – NCL testbed (UEssex)
Mapping to services	Secure-by-design orchestration service
Means of verification, methodology, tools	Evaluation through model outputs for detection accuracy; telemetry correlation using Prometheus, AI model training logs, and real-time orchestration feedback loops.

Req-ID	REQ S1-F-C1
Name	Orchestration over edge-cloud for energy sustainable security-by-design
Description	This component focuses on AI-driven scheduling using federated learning to ensure secure, energy-efficient, and delay-aware orchestration of 6G network slices. It leverages federated learning to train AI models locally at edge nodes, preserving privacy and reducing bandwidth usage while optimizing resource allocation to balance energy consumption, delay, and security.

Req-ID	REQ S1-F-C1
	The system operates within a closed-loop framework, utilizing near-real-time telemetry and Cyber Threat Intelligence (CTI) data to adaptively re-optimize slice resources. It integrates tools like Kubernetes and lightweight distributions like K3s [8] for managing cloud and edge clusters. It enables seamless coordination for microservice chaining, initially using solutions such as the Multi-Cluster Service API (MCS API) [9](and its wrapper Submariner [10]) and potentially exploring service mesh solutions. The MCS API enables service peering across a fleet of Kubernetes clusters through DNS exports, while service mesh enable service-level communication within or across a cluster.
Leading Partner	UEssex
Type	FUNCTIONAL
Priority	*SHOULD*
Validation sub-UC & testbed	UC1.1 – NCL testbed (UEssex)
Mapping to services	Secure-by-design orchestration service
Means of verification, methodology, tools	Validation through simulation and testbed deployment at UEssex. Metrics include energy usage, CTI-driven orchestration decisions, and slice performance under varying conditions. Telemetry monitoring tools and CTI feedback will be used to evaluate dynamic adaptation and optimization efficiency.

Req-ID	REQ S3-S-C1
Name	Secure-by-design orchestration service
Description	This middleware service monitors the status of a cluster or domain and the security requirements of requested deployments, making configuration decisions to meet security and sustainability goals. These decisions may include actions like placement/scheduling and scaling within Kubernetes. The service has two main components: the orchestrator and the CNF manager. The orchestrator coordinates resources across clusters or domains, managing workload placement, scaling, and migration to ensure continuity and meet security and sustainability targets. The CNF manager oversees the lifecycle of cloud-native functions, supporting the orchestrator by maintaining performance and security standards through efficient scaling and updates.
Leading Partner	UEssex
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	UC1.1 – NCL testbed (UEssex)

Req-ID	REQ S3-S-C1
Mapping to services	Secure-by-design orchestration service
Means of verification, methodology, tools	Verification via deployment of services in NCL testbed under various trust/hygiene conditions. Sustainability performance will be measured using resource utilization and energy metrics under dynamic orchestration scenarios.

5.1.2. Sub-Use Case 1.2

Req-ID	REQ 1.2-1
Name	SECaaS validation over x86 workloads
Description	The sub use case 1.2 relates to the SECaaS hardening of x86 and covers both x86 and WASM workloads hardening. This requirement relates to the validation of x86 workloads only The x86 can be deployed natively or inside a container
Leading Partner	TSS
Type	FUNCTIONAL
Priority	Must-Have
Validation sub-UC & testbed	All KPI 1.3.1-1.3.4 relates to this sub use case 1.2 TSS 's testbed will be used to collect the KPIs. Possible integration on UESSEX or ISRD tested UESSEX or ISRD testbeds can eventually be used to test the solution on containerized UESSEX' micro service or ISRD' xAPP security (as discussed in T 3.4), leveraging the SECaaS hardening. If this occurs, TSS's SECaaS will remain on premises. Reversely, TSS will supply blockchain nodes used for integrity verification to be possibly installed inside UESSEX or ISRD premises if deemed appropriate by both hosting entities.
Mapping to services	S14 aka Workload hardening SECaaS
Means of verification, methodology, tools	KPIs 1.3.1-1.3.4 will be verified by means by: 1. Identifying a relevant set of the executables for a good coverage of the measurements 2. Producing timestamp-based measurements for latency and performance degradation 3. Tooling the execution environment with software-based energy monitoring 4. Driving tests for both native and containerized deployments For simplicity, these KPIs are restated here: **KPI 1.3** Respective x86 native payloads latency at start, performance degradation during runtime and overall energy waste for the aggregation

Req-ID	REQ 1.2-1
	of confidentiality, integrity runtime and correct execution monitoring (UC#1 .2, <1sec, <10%, <10%). This proposal stated KPI can be splitted as follows for simplicity: KPI 1.3.1, time for remote attestation cycle for x86 payloads < 1 sec KPI 1.3.2, time for payload decryption for x86 payloads < 3 sec KPI 1.3.3 performance degradation during runtime caused by runtime verification and performance monitoring for x86 payloads < 10 %. KPI 1.3.4, overall energy waste for the aggregation of confidentiality, integrity runtime verification and correct execution monitoring for x86 payloads < 10%.

Req-ID	REQ 1.2-2
Name	SECaaS validation for WASM workloads
Description	The sub use case 1.2 relates to the SECaaS hardening of x86 and covers both x86 and WASM workloads hardening. This requirement relates to the validation of WASM modules.
Leading Partner	TSS
Type	FUNCTIONAL
Priority	M: Must-Have
Validation sub-UC & testbed	All KPI 1.4.1-1.3.4 relates to this sub-use case. TSS 's testbed will be used to collect the KPIs. Considered integration with UESSEX or IMEC (as part of UC1) UESSEX and/or IMEC testbed are used for demonstrating the solution, notably with the support of D-MUTRA blockchain based remote attestation and runtime verification of WASM workloads. If this occurs, a technical requirement is to permit the deployment of WASMTIME modified runtime. The blockchain reversely does not require installation on the targeted execution environment (i.e., the testbed) and will be delivered by TSS
Mapping to services	S14-F aka Workload hardening SECaaS
Means of verification, methodology, tools	WASM security will be first attained with WASM payload runtime integrity verification, a significant step taken over the state of the art. WASM payload encryption will be then tested. These two security enablers will be attained through the modification of the WASM interpreter. On that sake, the open source WASMTIME interpreter will be considered.

Req-ID	REQ 1.2-2
	KPIs 1.4.1-1.4.3 will be verified by means by: 1. Technical feasibility study of applying integrity, confidentiality and availability preservation techniques for WASM workloads, delivered in D3.5 (M21). 2. Modification of WASMTIME runtime according to feasibility study result (ie, point 1. Above) 3. Identifying a relevant set of WASM modules for good coverage of the measurements 4. Producing timestamp-based measurements for latency and performance degradation 5. Tooling the execution environment with software-based energy monitoring For simplicity, these KPIs are restated below: **KPI 1.4** WASM security enforcement (according to our security challenge results), equivalent to x86 native implementation. We would split this KPI as below: KPI 1.4.1, Feasibility study covering the four novel security functions of confidentiality preservation, authenticity, runtime integrity and monitoring: 1 KPI 1.4.2, Development of novel WASM security functions as the resulting of the feasibility study: 1 KPI 1.4.3, alignment with KPI 1.3 latency, performance degradation and energy waste: 1 These KPIs will be defined with the outcomes of the feasibility study.

5.1.3. Sub-Use Case 1.3

Req-ID	REQ S2-S-C1
Name	Feather
Description	Feather is a Kubernetes-compatible service orchestration agent designed explicitly for low-resource edge devices. It only implements the subset of Kubernetes features useful for edge computing and removes heavy cloud dependencies to reduce agent footprint and dependencies. Advanced features include microVM support for mixed workload pods.
Leading Partner	IMEC
Type	SYSTEM
Priority	*MUST*
Validation sub-UC & testbed	UC1.3 - IMEC & UEssex testbeds

Req-ID	REQ S2-S-C1
Mapping to services	Attack Resilient payload engine
Means of verification, methodology, tools	Create a generic platform for the integration of new runtimes w.r.t. storage, (pod) networking and payload execution. Evaluate functional and non-functional properties of runtimes as implemented (security, mapping to expected container functionality, resource use).

Req-ID	REQ S2-S-C2
Name	Trust-Edge
Description	Trust-edge is a platform for securely enrolling edge devices as trusted and remotely attested Kubernetes worker nodes. It integrates with Feather to create remotely attested Feather workers.
Leading Partner	IMEC
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	UC1.3 - IMEC & UEssex testbeds
Mapping to services	Attack Resilient payload engine
Means of verification, methodology, tools	Orchestrator/operator (Trust-Edge) statistics on attested devices and denied connections/credentials compared to ground truth for scenarios.

Req-ID	REQ 1.3-1
Name	Green-energy-awareness
Description	The orchestrator must be able to detect sources of green energy and take them into account during scheduling and rescheduling of workloads, preferably in near real-time and with low migration overhead.
Leading Partner	IMEC
Type	FUNCTIONAL
Priority	SHOULD
Validation sub-UC & testbed	UC1.3 - IMEC & UEssex testbeds
Mapping to services	Attack Resilient/green orchestration

Req-ID	REQ 1.3-1
Means of verification, methodology, tools	Feedback from orchestrator/scheduling algorithm, power consumption statistics of devices combined with site-dependent power statistics.

Req-ID	REQ 1.3-2
Name	Intent-based
Description	There must be a trustworthy source of green energy information, as well as node security properties, in the form of node and payload metadata. This enables the orchestrator to take various requirements into account based on trusted information.
Leading Partner	IMEC
Type	NON-FUNCTIONAL
Priority	*MUST*
Validation sub-UC & testbed	UC1.3 - IMEC & UEssex testbeds
Mapping to services	Attack Resilient payload engine Attack Resilient/green orchestration
Means of verification, methodology, tools	Flocky is intent-based by design; develop intents to match other use case requirements, verify by (automated) comparison of live cluster to intended (theoretical) cluster with all its available properties.

Req-ID	REQ 1.3-3
Name	Hardware & infrastructure support
Description	Underlying infrastructure must enable appropriate trust to ensure workload can safely move between datacenters without compromising privacy and Intellectual Property of workloads.
Leading Partner	IMEC
Type	SYSTEM
Priority	*MUST*
Validation sub-UC & testbed	UC1.3 - IMEC & UEssex testbeds
Mapping to services	Attack Resilient payload engine Attack Resilient/green orchestration
Means of verification,	- Manual verification for and during use case setups

methodology, tools	- Automated TPM & other feature detection and ongoing verification at runtime by software; comparison with use case setup for accuracy/efficiency

Req-ID	REQ 1.3-4
Name	Cross-site orchestrator compatibility
Description	Involves setting up a multi-location compute mesh with trusted computing-enabled hosts and verified sources of green energy information. Specifically, this will include a Kubernetes cluster spanning multiple geographic locations and using Remotely Attested Kubernetes workers to ensure the trustworthiness of the compute node.
Leading Partner	IMEC
Type	SYSTEM, NON-FUNCTIONAL
Priority	*MUST*
Validation sub-UC & testbed	UC1.3 - IMEC & UEssex testbeds
Mapping to services	Attack Resilient/green orchestration
Means of verification, methodology, tools	In cooperation with UESSEX. Verification includes: - Checking node presence and correct properties after joining the Kubernetes cluster (kubectl, kube API) - Validating the correctness of attestation mechanism (kube API, TrustEdge)

5.2. Use Case 2

5.2.1. Sub-Use Case 2.1

Firstly, we define the functional requirements of sub-use case 2.1. They are presented in detail in the following tables.

Req-ID	REQ 2.1-1
Name	Jamming Detection
Description	The physical layer security demands accurate and the on-time detection of any type of jamming attack (constant, periodic, reactive). The detection should also identify if the deterioration of the signal quality from the transmitter to the receiver is due to an attack or bad channel situation (e.g., blockage of Line-of-Sight)
Leading Partner	CERTH
Type	SYSTEM

Req-ID	REQ 2.1-1
Priority	MUST
Validation sub-UC & testbed	Sub-UC2.1, Sub-UC4.3, CERTH lab
Mapping to services	AI-based anti-jamming
Means of verification, methodology, tools	The verification will be done within the SDR-setup of CERTH lab.

Req-ID	REQ 2.1-2
Name	Jamming Mitigation
Description	After the detection, the jamming attack should be mitigated properly leading to a signal enhancement that will be sufficient for the accurate demodulation. The jamming mitigation should be done in near-real-time periods in order the communication link to be restored on the fly.
Leading Partner	CERTH
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	Sub-UC2.1, Sub-UC4.3, Simulation setup
Mapping to services	AI-based anti-jamming, ML-based MIMO
Means of verification, methodology, tools	The verification will be done using realistic simulation models.

Req-ID	REQ 2.1-3
Name	Jamming Identification
Description	The properties identification of the jammer(s) can make feasible, more efficient and better the mitigation of their impact. This procedure should cover all the jamming types (constant, periodic, reactive) and scenarios with multiple attackers within the network.
Leading Partner	CERTH
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	Sub-UC2.1, Sub-UC4.3, Simulation setup

Req-ID	REQ 2.1-3
Mapping to services	AI-based anti-jamming, ML-based MIMO
Means of verification, methodology, tools	The verification will be done using realistic simulation models.

Req-ID	REQ 2.1-4
Name	Acceleration of codebook compilation
Description	The codebook compilation of RIS configurations in mainly the exhaustive optimization procedure that links specific RIS functionality with the optimal state of its active elements. The acceleration of this procedure is of high importance. Different optimization tools will be evaluated. The investigation and pattern recognition based on the physics aspects for acceleration of the optimization procedure will also equip the optimization procedure for more direct movement close to the global, optimal value.
Leading Partner	CERTH
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	Sub-UC2.1, Simulation setup
Mapping to services	AI-based RIS configuration
Means of verification, methodology, tools	The verification will be done in CERTH SDR-based setup using a physical RIS unit.

Req-ID	REQ 2.1-5
Name	Signal Suppression
Description	One of the RIS properties is the suppression of the signal in specific areas. The usage of this aspect can support both proactive covertness communication and jamming mitigation. The investigation of this direction demands the development of accurate and realistic physics-based simulation setups. The time-efficient computation of RIS configuration for directive suppression without diminishing signal delivery in other regions is the final goal.
Leading Partner	CERTH
Type	SYSTEM
Priority	MUST

Req-ID	REQ 2.1-5
Validation sub-UC & testbed	Sub-UC2.1, Simulation setup
Mapping to services	AI-based antijamming, ML-based MIMO, AI-based RIS configuration
Means of verification, methodology, tools	The verification will be done in CERTH SDR-based setup using a physical RIS unit and with realistic simulation frameworks.

Req-ID	REQ S4-S
Name	AI-Based RIS configuration
Description	An evaluation of the RIS units and their capabilities will be conducted. The analysis will focus on key functionalities, particularly the ability to steer communication signals toward desired directions while creating "quiet zones" to minimize interference in other areas of the network. Additionally, the system's sensing and localization features will enhance its physical layer security.
Leading Partner	CERTH
Type	SYSTEM
Priority	*MUST*
Validation sub-UC & testbed	*Sub-UC2.1, Sub-UC4.3, CERTH lab*
Mapping to services	AI-Based RIS configuration
Means of verification, methodology, tools	The verification will be done in CERTH SDR-based setup using the physical RIS unit.

Req-ID	REQ S5-S
Name	ML-based MIMO
Description	MIMO technology, employed in receiver and transmitter antennas, is a promising advancement for modern networks. However, its implementation presents challenges, particularly in signal processing techniques, which demand efficient parallel computation and low-latency solutions.
Leading Partner	CERTH
Type	SYSTEM
Priority	MUST

Req-ID	REQ S5-S
Validation sub-UC & testbed	Sub-UC2.1, CERTH lab
Mapping to services	ML-based MIMO
Means of verification, methodology, tools	The verification will be done using realistic simulation tools.

Req-ID	REQ S6-S-C1
Name	JASMIN & Filter Mitigation
Description	An AI-driven jamming detection module will accurately identify the presence of a jammer within the communication network. Upon detection, a signal-processing-based technique will be activated to design an appropriate filter, enabling the separation of legitimate signals from interference. Advanced methods, such as Physical Layer Key Generation (PKG) and RIS-assisted communication paths will further enhance the system's security capabilities.
Leading Partner	CERTH
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	Sub-UC2.1, CERTH lab
Mapping to services	AI-based anti-jamming
Means of verification, methodology, tools	The verification will be done in a dual manner. The jamming mitigation via the SDR-setup of CERTH lab and the mitigation using realistic simulation models.

5.2.2. Sub-Use Case 2.2

Req-ID	REQ 2.2-1
Name	DetAction spectrum monitoring
Description	Spectrum must be monitored to inspect the signals present at a given frequency and extract the key features (e.g. SINR) to perform the detection of jamming signals.
Leading Partner	GRADIANT
Type	SYSTEM, NON-FUNCTIONAL
Priority	SHOULD

Req-ID	REQ 2.2-1
Validation sub-UC & testbed	Sub-UC2.2, Gradiant 5G Lab
Mapping to services	AI-based anti-jamming
Means of verification, methodology, tools	Spectrum can be monitored providing captures and representations of the frequencies observed and used in communication.

Req-ID	REQ 2.2-2
Name	Jamming detection
Description	Identification of jamming signals, as they will usually be masked by legitimate signals.
Leading Partner	GRADIANT
Type	SYSTEM, FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC2.2, Gradiant 5G Lab
Mapping to services	AI-based anti-jamming
Means of verification, methodology, tools	The detection of jamming would be verified by showing the output of the AI algorithm, and applying metrics with it as accuracy, recall, f1-score and others.

Req-ID	REQ 2.2-3
Name	Jamming mitigation
Description	Action/countermeasures to mitigate jamming attacks such as frequency hopping or adaptive beamforming.
Leading Partner	GRADIANT
Type	SYSTEM, FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC2.2, Gradiant 5G Lab
Mapping to services	AI-based anti-jamming

Req-ID	REQ 2.2-3
Means of verification, methodology, tools	The mitigation of the jamming can be verified from two perspectives: - The first one is to simply measure the avoidance of the frequencies being attacked in a statistical approach - The other is to compute metrics as throughput, SINR and others and see how they change when the countermeasures are activated This is dependent on the jamming detection phase, which will provide input for the action*/countermeasures done by the reaction phase*

Req-ID	REQ 2.2-4
Name	Multi-path routing
Description	Selection of alternative path to avoid jamming signals.
Leading Partner	GRADIANT
Type	SYSTEM, FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC2.2, Gradiant 5G Lab
Mapping to services	AI-based anti-jamming
Means of verification, methodology, tools	As the previous one, the selection of the alternative path depends on the detection and localization of the jamming signal on the spectrum. This routing can be verified by monitoring the spectrum and following where the jamming attack is and which frequencies are selected to avoid it.

Req-ID	REQ S6-S-C2
Name	DetAction: Detection and reAction against jamming attacks
Description	AI-based framework for the detection of jamming attacks and the adoption of countermeasures against them. Jamming detection over UEs (downlink) relies on available RAN performance metrics. Jamming detection over gNB (uplink) relies on wideband I/Q preprocessing.
Leading Partner	GRADIANT
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	Sub-UC2.2, Gradiant 5G Lab
Mapping to services	AI-based anti-jamming
Means of verification, methodology, tools	Validate the detection phase by measuring accuracy, F1-score and confusion matrix to check the performance of the AI detection algorithm against jamming attacks of different powers. Validate the reaction phase by checking the effective change of frequency to avoid the jamming attack

5.2.3. Sub-Use Case 2.3

Req-ID	REQ 2.3-1
Name	Jamming Detection
Description	Jamming attack must be detected
Leading Partner	ISRD
Type	SYSTEM, FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub UC2.3, ISRD Testbed
Mapping to services	AI-based anti-jamming
Means of verification, methodology, tools	The detection of jamming would be verified by showing the output of the JDM-xApp algorithm, and applying appropriate metrics.

Req-ID	REQ 2.3-2
Name	Jamming Mitigation
Description	Jamming Attack must be mitigated
Leading Partner	ISRD
Type	SYSTEM, FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub UC2.3, ISRD Testbed
Mapping to services	AI-based anti-jamming
Means of verification, methodology, tools	The mitigation of the jamming can be verified by computing metrics such as throughput, SINR and others and see how they change when the countermeasures are activated.

Req-ID	REQ A-S6-S-C3
Name	AMC-based Jamming Detection and Mitigation
Description	Continuous adaptation of the traditional MCS algorithm to maintain best signal metrics under jamming scenario
Leading Partner	ISRD
Type	SYSTEM, NON-FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC2.4, ISRD Testbed

Req-ID	REQ A-S6-S-C3
Mapping to services	AI-based anti-jamming
Means of verification, methodology, tools	The detection of jamming would be verified by showing the output of the JDM-xApp algorithm, and applying appropriate metrics. The mitigation of the jamming can be verified by computing metrics such as throughput, SINR and others and see how they change when the countermeasures are activated

5.2.4. Sub-Use Case 2.4

Req-ID	REQ 2.4-1
Name	High quality metrics extraction
Description	The system must accurately extract relevant metrics from the communication channel, as channel state information (CSI), to serve as the basis for key generation.
Leading Partner	GRADIANT
Type	SYSTEM, FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC2.4, Gradiant 5G Lab
Mapping to services	Characteristics Extraction Service
Means of verification, methodology, tools	Comparing its output against controlled reference measurements under varied channel conditions to ensure reliability and randomness.

Req-ID	REQ 2.4-2
Name	AI model Optimization
Description	The AI model must process the collected metrics to optimize the key generation process, ensuring that the keys are generated without discrepancies (Key Generation Rate - KGR).
Leading Partner	GRADIANT
Type	SYSTEM, FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC2.4, Gradiant 5G Lab
Mapping to services	Key Generation Service

Req-ID	REQ 2.4-2
Means of verification, methodology, tools	Measurement of channel metrics and measurement with the output from the AI module. Compare key generation rates under FDD and TDD scenarios and assess the impact of AI optimizations.

Req-ID	REQ 2.4-3
Name	Security Evaluation
Description	The generated keys must pass security checks, such as the NIST random test, to ensure randomness and resistance to attacks.
Leading Partner	GRADIANT
Type	SYSTEM, NON-FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC2.4, Gradiant 5G Lab
Mapping to services	Security Validation Service
Means of verification, methodology, tools	Evaluation by NIST Test suite of the generated keys.

Req-ID	REQ A-S6-S-C4
Name	PKGen: Generation of secure key for sub-TH bands
Description	The PKG system, enhanced with AI, processes channel metrics to simultaneously generate a symmetric key for Alice and Bob.
Leading Partner	GRADIANT
Type	SYSTEM, NON-FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC2.4, Gradiant 5G Lab
Mapping to services	Characteristics Extraction Service, Key Generation Service and Security Validation Service
Means of verification, methodology, tools	Validate secure links under simulated eavesdropping, verify key randomness, and demonstrate improved KGR with reduced KDR in both FDD and TDD.

5.3. Use Case 3

5.3.1. Sub-Use Case 3.1

Corresponds to service requirement S11-S in [NATWORK-D2.3]: AI-driven security monitoring for anomaly detection and root cause analysis in IoT networks and 3 Use Case requirements.

Req-ID	S11-S
Name	AI-driven security monitoring for anomaly detection and root cause analysis in IoT networks
Description	AI-based intrusion detection system (IDS) uses advanced machine learning techniques, such as Convolutional Neural Networks (CNNs) and reinforcement learning. Its primary goal is to detect anomalies in 5G/IoT network traffic, often indicating early signs of potential DDoS attacks or other malicious activities such as data breaches, and unauthorized access, by analyzing network traffic patterns and device behaviors.
Leading Partner	MONT
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	UC3.1, UC4.5; MONT 5G/IoT testbed
Mapping to services	Security monitoring
Means of verification, methodology, tools	Generation of 5G and IoT network traffic. Injection of attacks and use of open source datasets. Training of ML models. Replay of network traffic. Detection of anomalies using MMT monitoring Framework. Validation using the following KPIs: Mean Time to Detect (MTTD), Number of False Positives (FP), Number of False Negatives (FN), Packet Loss Ratio (PLR), and Mean Time to Resolve (MTTR).

Req-ID	REQ 3.1-1
Name	Continuous Monitoring & Data Collection
Description	Real-time network monitoring. Need to consider for instance: encrypted network traffic; packets and flows metadata from SDN, MEC, NTN, IoT and core networks; system and application logs; events from servers, VMs, containers, microservices and endpoints; IoT & OT sensor data; telemetry from industrial control systems (ICS), SCADA and smart devices.
Leading Partner	MONT
Type	FUNCTIONAL
Priority	MUST

Req-ID	REQ 3.1-1
Validation sub-UC & testbed	UC#3.1, MONT 5G testbed with MMT monitoring framework and IoT wireless sniffer
Mapping to services	Security monitoring
Means of verification, methodology, tools	Performing several use case scenarios to demonstrate obtaining the following results: -Extracted statistics and features. These need to be evaluated by end-users to obtain user-reported experiences, feedback on functionality, subjective usability ratings, and qualitative insights.

Req-ID	REQ 3.1-2
Name	AI-Driven Threat Detection, Anomaly Analysis & Root Cause Analysis
Description	Need to analyse extracted statistics and features and perform real-time analysis and detection of suspicious activity with severity classification; produce security event logs/alarms/reports; store forensics data and timestamped logs for investigation and compliance; and interact with security orchestrator to mitigate/respond/prevent security breaches. The detection rules and algorithms should reflect the specified security policies, and be dynamically updated to adapt to changing threats (e.g., continuous learning and consideration of CTI).
Leading Partner	MONT
Type	FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	UC#3.1, MONT 5G testbed with MMT monitoring framework and IoT wireless sniffer
Mapping to services	Security monitoring and anomaly detection and response
Means of verification, methodology, tools	Performing several use case scenarios to demonstrate obtaining the following results: -Real-time alerts and anomaly reports; actionable insights for SecOps teams, compliance officers and decision-makers; root cause analysis (RCA) insights for pinpointing attack origin, affected assets; and providing possible mitigation steps. These need to be evaluated by end-users to obtain user-reported experiences, feedback on functionality, subjective usability ratings, and qualitative insights. The functions need to be scalable (i.e., tested in real or simulated environments with large number of devices and high bandwidths, accurate (i.e., with reduced false positive and true negative rates) evaluated using penetration and fuzz testing.

Req-ID	REQ 3.1-3
Name	Advanced Visualization & Reporting
Description	Need to present extracted statistics, features and analysis results (e.g., attack alerts and reports) in intuitive dashboards, and provide pertinent information to the security orchestrators for zero-touch security service management (ZSSM) loops.
Leading Partner	MONT
Type	FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	UC#3.1, MONT 5G testbed with MMT monitoring framework and IoT wireless sniffer
Mapping to services	Security monitoring and ZSSM
Means of verification, methodology, tools	Performing several use case scenarios to demonstrate obtaining the following results: -Providing real-time security insights for security analysts, network operators, and automated mitigation and prevention of security breaches. These need to be evaluated by end-users to obtain user-reported experiences, feedback on functionality, subjective usability ratings, and qualitative insights. Test scenarios need to show the effectiveness and efficiency of integration with other network services (e.g., decision engines, security orchestrators).

5.3.2. Sub-Use Case 3.2

Req-ID	REQ S8-S-C3 (3.2-1)
Name	AI-enabled DoS attack
Description	An AI-powered penetration testing tool that surpasses traditional solutions. Unlike other tools, it models intricate DoS attack scenarios and provides deeper insights into network vulnerabilities. Combining DoS attacks with protocol-level fuzzing generates custom network packets tailored to target 5G services. This approach reveals vulnerabilities that other tools may miss, offering a comprehensive evaluation of the network's capabilities and communication protocols, ultimately bolstering the security of 5G and 6G networks.
Leading Partner	CERTH
Type	S: System
Priority	M: Must-Have
Validation sub-UC & testbed	UC#3.2, CERTH testbed

Req-ID	REQ S8-S-C3 (3.2-1)
Mapping to services	AI-based Intrusion Detection: AI driven penetration Testing
Means of verification, methodology, tools	Measurement of QoS impact of the proposed approach and other attack tools that affect QoS. The aim is to assess the extent to which service quality is compromised during the attack, such as reduced performance or outages. It measures the overall impact on end users, including any degradation in their experience. This evaluation helps identify weaknesses and gauge the network's stability under adverse conditions. We consider the requirement to be successful if the following criteria are fulfilled: – The AI-DoS will be considered successful if it can effectively reduce QoS by more than 70% in the evaluated 5G/6G service provided by CERTH. – The impact of AI-DoS on QoS must be over 70% compared to other DoS evaluation tools. – AI-DoS will have to provide detailed information about which strategy it implemented and its impact on reducing the QoS.

Req-ID	REQ 3.2-2
Name	Hardware Requirements
Description	The hardware requirements for the PC required to run the Use Case.
Leading Partner	CERTH
Type	FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	UC#3.2, CERTH testbed
Mapping to services	AI-based Intrusion Detection: AI driven penetration Testing
Means of verification, methodology, tools	Requirement is met (YES/NO): Multi-core processor (e.g., AMD Ryzen 7 or Intel Core i7), at least 16 GB of RAM, at least 1TB RAM, Linux with KVM, or Windows with Hyper-V.

Req-ID	REQ 3.2-3
Name	Minimum protocols utilized in UC
Description	The communication protocols that are available to be targeted by DoS attacks in the testbed.
Leading Partner	CERTH
Type	FUNCTIONAL
Priority	MUST

Req-ID	REQ 3.2-3
Validation sub-UC & testbed	UC#3.2, CERTH testbed
Mapping to services	AI-based Intrusion Detection: AI driven penetration Testing
Means of verification, methodology, tools	Requirement is met (YES/NO): To achieve the assessment of resilience to DoS attacks, the target services should include protocols such as TCP, UDP and SCTP.

Req-ID	REQ 3.2-4
Name	Effectiveness of AI-DOS
Description	The effectiveness of AI-DoS should be compared with other DoS attack tools and the results presented.
Leading Partner	CERTH
Type	FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	UC#3.2, CERTH testbed
Mapping to services	AI-based Intrusion Detection: AI driven penetration Testing
Means of verification, methodology, tools	Requirement is met (YES/NO): Other DoS attack tools that perform the same attacks are available in the testbed. A report is produced for comparison of their performance against the proposed solution

Req-ID	REQ 3.2-5
Name	Phishing mail evaluation
Description	For the effectiveness of LLM, it would be good to have various people, that could be targets of such a mail, to evaluate the persuasiveness of the emails it produces.
Leading Partner	CERTH
Type	FUNCTIONAL
Priority	SHOULD
Validation sub-UC & testbed	UC#3.2, CERTH testbed
Mapping to services	AI-based Intrusion Detection: AI driven penetration Testing
Means of verification,	Requirement is met (YES/NO): Survey that examines if personalize phishing mails produced by LLM is persuasive.

Req-ID	REQ 3.2-5
methodology, tools	

5.3.3. Sub-Use Case 3.3

Req-ID	REQ 3.3-1
Name	Decentralized Trust Management
Description	Trust relationships and security decisions lead to trust and access control must be managed without centralized entities. The trust management requirement ensures that the IoT user equipment (UE) and the IoT service provider can securely verify each other's identity and actions before exchanging data. It enables secure and end-to-end communication.
Leading Partner	ELTE
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	Sub-UC#3.3 ELTE testbed
Mapping to services	S3-Security by Design Orchestration S3-S-C2 - E2E Security Management
Means of verification, methodology, tools	The means of verification involve testing whether trust is correctly established between the IoT UE and the IoT service provider using the testbed. The methodology includes deploying virtualized 5G core, RAN, IoT UE, and IoT service provider (DN), and using Foundry blockchain with smart contracts to manage trust. Tools include network logs, smart contract execution records, and blockchain transaction data to confirm secure interactions and identity verification. We consider the requirement to be successful if the IoT node established a secure connection with the IoT service provider.

Req-ID	REQ 3.3-2
Name	Real-time Trust and Access Establishment
Description	Trust and access must be controlled, established, monitored, and updated dynamically as devices and users join or leave the network. The system also must detect and block any malicious IoT UE that tries to establish trust and access the IoT service provider. This helps protect the system from unauthorized access and potential attacks.
Leading Partner	ELTE
Type	FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC#3.3 ELTE testbed

Req-ID	REQ 3.3-2
Mapping to services	S3-Security by Design Orchestration S3-S-C2 - E2E Security Management
Means of verification, methodology, tools	Verification is done by simulating a malicious IoT UE with invalid or tampered identity data. The methodology includes attempting to connect this UE to the IoT service provider through the 5G-enabled IoT network and checking if the system correctly denies access. Tools used include smart contract logs, blockchain transaction history, and security monitoring logs on the IoT service provider machine to confirm that the malicious device was identified and blocked. We consider the requirement to be successful if the malicious UE denied access to the IoT service provider.

Req-ID	REQ 3.3-3
Name	Security Data Aggregation
Description	Aggregate the security and trust data in a secure and privacy preserving approach. The system must allow the IoT service provider to verify the identity of the IoT UE by checking its credentials stored on the blockchain. This ensures that only trusted devices can access the service.
Leading Partner	ELTE
Type	FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC#3.3 ELTE testbed
Mapping to services	S3-Security by Design Orchestration S3-S-C2 - E2E Security Management
Means of verification, methodology, tools	Verification is done by testing whether the IoT service provider can successfully read and validate the IoT UE's identity data from the blockchain using smart contracts. The methodology includes the IoT service provider reading attempts and observing the blockchain interactions. Tools used include blockchain explorers, smart contract logs, and system logs on the IoT service provider machine to confirm that identity checks are correctly triggered and validated. We consider the requirement is successful if the IoT service provider could successfully verify the identity of an already registered UE.

Req-ID	REQ S3-S-C2-1
Name	Trust Establishment
Description	The service provides establishing trust between IoT devices and service providers based on decentralized trust records.
Leading Partner	ELTE
Type	FUNCTIONAL

Req-ID	REQ S3-S-C2-1
Priority	MUST
Validation sub-UC & testbed	Sub-UC#3.3 ELTE testbed
Mapping to services	S3-Security by Design Orchestration
Means of verification, methodology, tools	Trust establishment between the IoT device and the IoT service provider is validated using a controlled testbed. The setup comprises a virtualized 5G Core, RAN, IoT UE, and data network (DN) representing the service provider. Trust management is facilitated through Foundry blockchain and smart contracts. Verification relies on analyzing network logs, smart contract executions, and blockchain transactions. The requirement is considered met if the IoT node successfully initiates a secure and authenticated session with the service provider.

Req-ID	REQ S3-S-C2-2
Name	Privacy
Description	The service must ensure that generated service-related tokens do not expose sensitive data, using anonymization or hashing to protect privacy.
Leading Partner	ELTE
Type	NONFUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC#3.3 ELTE testbed
Mapping to services	S3-Security by Design Orchestration
Means of verification, methodology, tools	Verification focuses on ensuring that the blockchain-stored token used for trust does not reveal sensitive or identifiable information. The methodology involves reviewing blockchain entries and smart contract logic to confirm that only anonymized or hashed data is recorded. Tools such as formal verification tools, blockchain explorers and hash validators are used to inspect transaction payloads and token content. The requirement is met if the token preserves privacy and no raw identity data is exposed on-chain.

5.4. Use Case 4

5.4.1. Sub-Use Case 4.1

Use case requirements:

Req-ID	REQ 4.1-1
Name	Availability of programmable data plane devices
Description	The DFE and WAI are embedded as offloading data plane functions running inside programmable devices such as P4 switches or SmartNICs with acceleration capabilities (i.e., DPU). Backends with the following features are strictly required to implement the offloading of network functions inside them with API.
Leading Partner	CNIT
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	Sub-UC#4.1 ARNO testbed
Means of verification, methodology, tools	Evaluation in the ARNO testbed using DPUs and programmable switches, OFA interfaces with Security Orchestrator and DFE Telemetry to feed Intrusion Detection Systems

Req-ID	REQ 4.1-2
Name	Inter- and intra- edge data center scenarios with Ethernet connectivity from 25Gb/s up to 100Gb/s
Description	Connectivity between programmable edge nodes needed to test the attack detection location and check the capability of the orchestrator to decide where to enforce the offloaded functions in both static and dynamic cases.
Leading Partner	CNIT
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	Sub-UC#4.1 ARNO testbed
Means of verification, methodology, tools	Evaluation in the ARNO testbed using DPUs and programmable switches, OFA interfaces with Security Orchestrator and DFE Telemetry to feed Intrusion Detection Systems.

Req-ID	REQ 4.1-3
Name	Availability of telemetry collectors
Description	WAI and DFE can work in strict relationship with telemetry of features to external collectors. This is needed to understand if the offloaded function is always the opticaml one to block current attacks or should be replaced/updated. External anomaly detectors not running at wire speed are needed to close the loop.
Leading Partner	CNIT, MONT
Type	SYSTEM
Priority	SHOULD
Validation sub-UC & testbed	Sub-UC#4.1 ARNO testbed
Means of verification, methodology, tools	Evaluation in the ARNO testbed using DPUs and programmable switches, OFA interfaces with Security Orchestrator and DFE Telemetry to feed Intrusion Detection Systems.

Req-ID	REQ 4.1-4
Name	Control and management API
Description	WAI and DFE, including DFET Telemetry, require SDN-oriented and /or NFV-oriented dynamic configuration of security functions at the data plane (e.g., deployment of function, dynamic configuration at runtime, telemetry configuration and activation). This is needed to provide dynamicity of the offloaded functions configuration at the data plane.
Leading Partner	CNIT
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	Sub-UC#4.1 ARNO testbed
Means of verification, methodology, tools	Evaluation in the ARNO testbed using DPUs and programmable switches, OFA interfaces with Security Orchestrator and DFE Telemetry to feed Intrusion Detection Systems.

Component/service requirements:

Req-ID	REQ S9-F-C4
Name	WAI and DFE efficiency on blocking attacks at the data plane.
Description	The DFE and WAI are embedded as offloading data plane functions running inside programmable devices such as P4 switches or SmartNICs with acceleration capabilities (i.e., DPU). The DFE part is responsible for

Req-ID	REQ S9-F-C4
	extracting and storing stateless and stateful features from traffic packets and additional metadata to keep available for telemetry or local processing. WAI functions implement ML/AI algorithms directly inside the backend, producing real-time inference detection and mitigation at the data plane.
Leading Partner	CNIT
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	Sub-UC#4.1 ARNO testbed
Mapping to services	S9-F-C4: Wirespeed AI (WAI) and Decentralized Feature Extraction (DFE) (Service: P4 Behavioral Analysis)
Means of verification, methodology, tools	Evaluation in the ARNO testbed using DPUs and programmable switches, OFA interfaces with Security Orchestrator and DFE Telemetry to feed Intrusion Detection Systems. The target KPI are the following: - DFE processing latency <50 us with data plane device scalability up to 10k different flow rules. - DFE computational efficiency is 50% higher than existing methods (raw in-band telemetry). - DFE reduces power consumption by 20% compared to standard software-based feature selection and extraction at the compute engines. - WAI-based latency purely on hardware < 10 us, latency on software-based WAI < 100 us.

Req-ID	REQ S13-F-C2
Name	DFE Telemetry Efficiency
Description	The component relies on an offloaded data plane program configuring a telemetry stream, reporting a list of real-time per-packet or aggregated features for network analytics. The component may also feed distributed and federated learning collectors with telemetry of selected features.
Leading Partner	CNIT
Type	SYSTEM
Priority	SHOULD
Validation sub-UC & testbed	Sub-UC#4.1 ARNO testbed
Mapping to services	S13-F-C2: DFE Telemetry (Service: P4-based Analytics)

Req-ID	REQ S13-F-C2
Means of verification, methodology, tools	- Verify that the traffic flows under analysis are managed and pass through the switch/DPU. - Configure the desired DFE streams. - Verify that the streams are correctly generated and sent to the desired collectors. - DFE telemetry network and computational efficiency should be 50% higher than existing methods (raw in-band or out-of-band telemetry with cloning operation).

5.4.2. Sub-Use Case 4.2

Req-ID	REQ 4.2-1
Name	AI model disaggregation
Description	AI/ML models are disaggregated into slices for deployment across data plane components (switches, DPUs, NICs)
Leading Partner	ELTE
Type	FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC4.2, ELTE testbed
Mapping to services	Distributed Federated Learning across the Continuum
Means of verification, methodology, tools	Use the testbeds to deploy and validate slices with test traffic, check deployment and controller logs.

Req-ID	REQ 4.2-2
Name	Dynamic slice reconfiguration
Description	Real-time management of AI slices in response to network traffic patterns and workload changes
Leading Partner	ELTE
Type	SYSTEM
Priority	SHOULD
Validation sub-UC & testbed	Sub-UC4.2, ELTE testbed
Mapping to services	AI-based behavioral analysis
Means of verification,	Confirm that AI slices are reconfigured in real time based on changing traffic or workload without service disruption. Verify by sending test traffic

Req-ID	REQ 4.2-2
methodology, tools	with different characteristics and check the logs if reconfiguration occurred.

Req-ID	REQ 4.2-3
Name	Localized AI computation
Description	AI workloads are processed at the network's edge, closer to the data source
Leading Partner	ELTE
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	Sub-UC4.2, ELTE testbed
Mapping to services	AI-based behavioral analysis
Means of verification, methodology, tools	Deploy the AI workloads on edge devices and verify the logs and check if test traffic is processed correctly.

Req-ID	REQ 4.2-4
Name	Monitoring and verification of AI slices
Description	Continuous real-time monitoring of AI slices to ensure correct execution and performance
Leading Partner	ELTE
Type	SYSTEM
Priority	SHOULD
Validation sub-UC & testbed	Sub-UC4.2, ELTE testbed
Mapping to services	AI-based behavioral analysis
Means of verification, methodology, tools	After deployment, the controller is able to monitor the AI slices. Verify by sending test queries to the slices and checking the responses.

Req-ID	REQ S9-S-C3-1
Name	Deployment
Description	The ML model can be successfully trained by the coordinator, transmitted to the slice controllers, and deployed to the target programmable network devices (HW or SW) without errors.

Req-ID	REQ S9-S-C3-1
Leading Partner	ELTE
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	Sub-UC4.2, ELTE testbed
Mapping to services	AI-based behavioral analysis
Means of verification, methodology, tools	By performing detailed traffic inspection using tools such as tcpdump or Wireshark, we will be able to verify that AI slices have been correctly deployed and are actively processing data at the intended network locations.

Req-ID	REQ S9-S-C3-2
Name	Performance
Description	The deployed ML models achieve similar F1 scores to the centralized solution, with lower latency.
Leading Partner	ELTE
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	Sub-UC4.2, ELTE testbed
Mapping to services	AI-based behavioral analysis
Means of verification, methodology, tools	Side-by-side evaluation of output predictions from centralized and disaggregated deployments using standardized datasets (e.g., CIC-IDS), calculation of accuracy, precision, recall, and F1-score.

Req-ID	REQ S9-S-C3-3
Name	Adaptability
Description	The coordinator can dynamically update and redeploy models based on changing traffic patterns or operational requirements.
Leading Partner	ELTE
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	Sub-UC4.2, ELTE testbed
Mapping to services	AI-based behavioral analysis

Req-ID	REQ S9-S-C3-3
Means of verification, methodology, tools	Test dynamic updates and redeployment of AI models in response to simulated changes in traffic or operational conditions. Validate using controller logs and deployment timelines.

Req-ID	REQ S15-F-C1-1
Name	Privacy Preservation
Description	The cryptographic building blocks should ensure the security and privacy of aggregated data.
Leading Partner	ELTE
Type	SYSTEM
Priority	SHOULD
Validation sub-UC & testbed	Sub-UC4.2, ELTE testbed
Mapping to services	Distributed Federated Learning across the Continuum
Means of verification, methodology, tools	Demonstrate that no sensitive input can be inferred with a success rate significantly above random guessing.

Req-ID	REQ S15-F-C1-2
Name	Efficient Integration
Description	The component should be compatible with the infrastructure and integrate and function appropriately in the integrated service.
Leading Partner	ELTE
Type	SYSTEM
Priority	MUST
Validation sub-UC & testbed	Sub-UC4.2, ELTE testbed
Mapping to services	Distributed Federated Learning across the Continuum
Means of verification, methodology, tools	Verification that existing services remain stable and functional after integration. Validate using logs.

5.4.3. Sub-Use Case 4.3

This use case shares requirements with UC2.1 (REQ 2.1-1), shown in section 5.2.1.

5.4.4. Sub-Use Case 4.4

Req-ID	REQ 4.4-1
Name	Dynamic Resource Management
Description	The orchestration system must dynamically allocate resources (CPU, memory, bandwidth) across microservices, optimizing for performance, energy efficiency, and cost.
Leading Partner	CERTH
Type	S: System
Priority	M: Must-Have
Validation sub-UC & testbed	UC4.4, CERTH *testbed*
Mapping to services	Security-performance balancer
Means of verification, methodology, tools	Extensive testing will be performed to ensure that indicators such as latency, packet loss and throughput are within acceptable limits under stress scenarios.

Req-ID	REQ 4.4-2
Name	Real-time Adaptation
Description	The system must adapt to changing network conditions, user demands, and service requirements in real time, ensuring QoS consistency.
Leading Partner	CERTH
Type	S: System
Priority	M: Must-Have
Validation sub-UC & testbed	UC4.4, CERTH testbed
Mapping to services	Security by Design Orchestration, AI-based behavioural analysis
Means of verification, methodology, tools	Testing under stress scenarios (e.g. varying workload) will be performed to ensure that resource allocation adapts dynamically to microservices performance so that optimal microservice CPU and memory usage are achieved.

Req-ID	REQ 4.4-3
Name	Scalability
Description	The framework should handle the deployment of microservices across a distributed 6G network infrastructure, scaling up or down as needed.
Leading Partner	CERTH
Type	S: System
Priority	M: Must-Have
Validation sub-UC & testbed	UC4.4, CERTH testbed
Mapping to services	Security by Design Orchestration, AI-based behavioural analysis
Means of verification, methodology, tools	Extensive testing will be performed to validate that scaling actions are performed dynamically to meet microservices requirements.

Req-ID	REQ 4.4-4
Name	Resilience and Fault Tolerance
Description	The system must detect and mitigate failures in microservices or underlying infrastructure, ensuring service continuity.
Leading Partner	CERTH
Type	S: System
Priority	M: Must-Have
Validation sub-UC & testbed	UC4.4, CERTH testbed
Mapping to services	AI-based Intrusion Detection, AI-based behavioural analysis
Means of verification, methodology, tools	Testing under different attack scenarios to ensure that attacks are detected in a timely manner and corresponding mitigation actions are taken immediately.

Req-ID	REQ 4.4-5
Name	Security
Description	The orchestration process must incorporate security measures to protect microservices from attacks and ensure data integrity and confidentiality
Leading Partner	CERTH
Type	S: System
Priority	M: Must-Have
Validation sub-UC & testbed	UC4.4, CERTH testbed

Req-ID	REQ 4.4-5
Mapping to services	Security by Design Orchestration
Means of verification, methodology, tools	Testing under different attack scenarios to ensure that deployment decisions are made after attack detection in a timely manner to ensure uninterrupted service operation.

Req-ID	REQ S8-S-C1
Name	Multimodal Fusion Approach for Intrusion Detection System for DoS attacks
Description	The multimodal fusion IDS employs multiple AI models to analyze different types of data extracted from network traffic (e.g., statistical/temporal features, embeddings, and images). This approach aims to detect DoS attacks across diverse protocols in real time with high accuracy and minimal false positives.
Leading Partner	CERTH
Type	S: System
Priority	M: Must-Have
Validation sub-UC & testbed	UC4.4, CERTH testbed
Mapping to services	AI-based Intrusion Detection
Means of verification, methodology, tools	The requirement is considered to be met if the following criteria are successfully met: - Mean Time to Detect (MTTD) for DoS attacks under 5 minutes. - False Positive Rate (FPR) < 5%. - False Negative Rate (FNR) < 5%. - Supports diverse protocols in 5G/Beyond 5G environments. - Integration with security policies for real-time threat mitigation. Specifics attacks and scenarios to be defined.

Req-ID	REQ S8-S-C2
Name	Lightweight SDN-based AI-enabled Intrusion Detection System for cloud-based services
Description	This component orchestrates microservices dynamically in a 6G environment, combining AI and SDN-based Intrusion Detection to monitor resource consumption and respond to anomalies continuously in real time. The system leverages AI-driven profiling, anomaly detection, and automated mitigation for managing microservices' security and performance. By analyzing resource consumption patterns, the IDS detects and classifies suspicious behavior, including DoS attacks and zero-

Req-ID	REQ S8-S-C2
	day threats, while also orchestrating network policies (e.g., load balancing) to prevent microservice overload and ensure resilient service continuity.
Leading Partner	CERTH
Type	S: System
Priority	S: Should-Have
Validation sub-UC & testbed	UC4.4, CERTH testbed
Mapping to services	AI-based Intrusion Detection
Means of verification, methodology, tools	The requirement is considered to be met if the following criteria are successfully met: - Me Detection Accuracy: ≥ 80% accuracy in detecting anomalies related to microservice resource usage. - Mitigation Time: Response within 2 seconds to isolate malicious traffic or reallocate resources under stress. - Scalability: System supports real-time scaling to prevent overloads during traffic surges. - Mean Time to Detect (MTTD) attacks within 5 minutes. - False Positive Rate (FPR) < 5% - Efficient resource monitoring across CPU, memory, and network usage. - Effective integration with SDN controllers for real-time data flow adjustments in response to detected threats. Specifics attacks and scenarios to be defined.

Req-ID	REQ S9-F-C5
Name	Microservice behavioral analysis for detecting malicious actions
Description	This component orchestrates microservices dynamically in a 6G environment, combining AI and SDN-based Intrusion Detection to monitor resource consumption and respond to anomalies continuously in real time. The system leverages AI-driven profiling, anomaly detection, and automated mitigation for managing microservices' security and performance. By analyzing resource consumption patterns, the IDS detects and classifies suspicious behavior, including DoS attacks and zero-day threats, while also orchestrating network policies (e.g., load balancing) to prevent microservice overload and ensure resilient service continuity.
Leading Partner	CERTH
Type	F: Functional
Priority	C: Could-Have

Req-ID	REQ S9-F-C5
Validation sub-UC & testbed	UC4.4, CERTH testbed
Mapping to services	AI-based behavioural analysis
Means of verification, methodology, tools	The requirement is considered to be met if the following criteria are successfully met: - The system detects and flags resource anomalies linked to potential attacks. - Mitigation actions prevent service degradation. - QoS parameters remain stable during attack scenarios. - False Positive Rate (FPR) <5%. - False Negative Rate (FNR) <5%. Specifics attacks and scenarios to be defined.

Req-ID	REQ S12-F
Name	Security-performance balancer
Description	The service aims to balance the performance of radio elements, and the security added to the radio for the constant availability of radio resources. The balancer will consider, on the one hand, the risks that appeared in the radio interface and, on the other hand, the performance requirements posed with the radio software/hardware due to increased traffic. The main task of the balancer is to understand when the increased performance required is due to an attack in progress or regular peak traffic. The balancer will inform the agents when they should apply for a deeper packet inspection or when the security controls can be reduced.
Leading Partner	CERTH
Type	F: Functional
Priority	M: Must-Have
Validation sub-UC & testbed	UC4.4, CERTH testbed
Mapping to services	Security-performance balancer
Means of verification, methodology, tools	The requirement is considered to be met if the following criteria is successfully met: - Verify that the service receives data from RAN components and the security xApp. - Verify that the balancer returns a decision. - Verify that the optimization policies are applied correctly in the RAN network Specifics optimization scenarios to be defined.

5.4.5. Sub-Use Case 4.5

Req-ID	REQ 4.5-1
Name	MTD Framework Scalability
Description	Scaling the usage of MTD operations on a large set of network functions spanning network slices operated both on edge and core infrastructures.
Leading Partner	ZHAW
Type	SYSTEM, NON-FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC4.5, PNET 5G Lab
Mapping to services	S10-S: AI-based MTD
Means of verification, methodology, tools	MTD Framework can be tested under varying load conditions. When the workload is increased by NFV MANO / Kubernetes via more VNFs/CNFs, the MTD Framework should continue to operate within reasonable response time.

Req-ID	REQ 4.5-2
Name	Network State Assessment
Description	Monitoring the network and present the network state in a near real-time manner with a formal model for application of MTD strategy optimization with deep-RL.
Leading Partner	ZHAW
Type	SYSTEM, FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC4.5, PNET 5G Lab
Mapping to services	S10-S: AI-based MTD
Means of verification, methodology, tools	Under different network conditions (e.g., attack frequency, load, etc.), the evaluated network state should be different so that the MTD framework can determine the best action based on varying network state.

Req-ID	REQ 4.5-3
Name	Multi-Tenant Support
Description	MTD mechanism must properly operate in an environment where various CSPs are running their own MTD frameworks on a shared network infrastructure, without having any access to other tenants' data.
Leading Partner	ZHAW
Type	SYSTEM, FUNCTIONAL

Req-ID	REQ 4.5-3
Priority	MUST
Validation sub-UC & testbed	Sub-UC4.5, Patras 5G testbed
Mapping to services	S10-S: AI-based MTD
Means of verification, methodology, tools	Multiple CSPs can be simulated on the shared infrastructure, each having its own deployment of the MTD Framework. We should ensure that the environment of each tenant should be isolated from others such that the MTD framework will not have any access to the data of other tenants.

Req-ID	REQ 4.5-4
Name	Explainable MTD Strategies
Description	MTD Framework must provide humanly interpretable, high-level explanations on the determined actions by the framework, using Explainable AI (XAI) techniques.
Leading Partner	ZHAW
Type	SYSTEM, FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC4.5, Patras 5G testbed
Mapping to services	S10-S: AI-based MTD
Means of verification, methodology, tools	Either separate XAI models, targeting deep-RL algorithms, need to be developed for providing explanations on MTD decisions, or the MTD decision process should be adjusted to include inherent explainability. Certain metrics for measuring the quality of XAI can be adapted to evaluate the explainability part, with the help of expert feedback.

Req-ID	REQ 4.5-5
Name	MTD with Federated Learning (MTDFed)
Description	Multiple CSPs, with varying setups/environments, should be able to collaboratively train a global model for achieving a more accurate MTD Framework, without revealing their private data.
Leading Partner	ZHAW
Type	SYSTEM, FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC4.5, Patras 5G testbed
Mapping to services	S10-S: AI-based MTD

Req-ID	REQ 4.5-5
Means of verification, methodology, tools	Multiple CSPs can be simulated on the shared infrastructure, each having its own deployment of the MTD Framework. Then, using a Federated Learning framework (e.g., Flower), they can train a global MTD model. The performance of the global model can be compared against individually trained models.

Req-ID	REQ S10-F-C1
Name	MTD Controller
Description	The component handles the MTD actions determined by the Strategy Optimizer, enabling migrating one service (e.g., VNF/CNF) from a source node/slice to a destination node/slice.
Leading Partner	ZHAW
Type	SYSTEM, FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC4.5, Patras 5G testbed
Mapping to services	S10-S: AI-based MTD
Means of verification, methodology, tools	**Security Impact**: Demonstrable reduction in LSE through dynamic MTD actions, measured against a baseline of static configurations. **Performance Metrics**: MTD operations should maintain acceptable service latency levels (within a defined threshold) and optimize energy consumption within the edge-to-cloud continuum.

Req-ID	REQ S10-F-C2
Name	MTD Strategy Optimizer
Description	This component will integrate AI-driven policy optimization, focusing on Deep Reinforcement Learning (DRL), to optimize MTD strategies and dynamically orchestrate payload migrations. This orchestration will consider multiple domains and seek to enhance security without compromising network functionality.
Leading Partner	ZHAW
Type	SYSTEM, FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC4.5, Patras 5G testbed
Mapping to services	S10-S: AI-based MTD
Means of verification,	**Security Impact:** Demonstrable reduction in LSE through dynamic MTD actions, measured against a baseline of static configurations.

Req-ID	REQ S10-F-C2
methodology, tools	**Adaptability**: The system should successfully incorporate data from infrastructure performance, vulnerability assessments, and threat intelligence to inform real-time MTD policies and be adaptable across different network domains. This criterion reflects the bio-inspired principle of adaptive immunity. **AI Optimization**: The DRL-driven policy optimization must improve the balance of security benefits and operational overhead, outperforming static or rule-based policies in efficiency tests.

Req-ID	REQ S10-F-C3
Name	MTD Explainer
Description	This component provides human-interpretable explanations for the MTD actions decided by the MTD Strategy Optimizer. MTD Explainer will take the environmental data, along with the MTD actions, as inputs and provide a reasonable explanation for each action and why MTD Strategy Optimizer decided on this action.
Leading Partner	ZHAW
Type	SYSTEM, FUNCTIONAL
Priority	MUST
Validation sub-UC & testbed	Sub-UC4.5, Patras 5G testbed
Mapping to services	S10-S: AI-based MTD
Means of verification, methodology, tools	**Clear Explanations:** The explanations generated by the MTD Explainer should be understandable by humans. **Robust Explanations**: Explanations should be consistent across the same actions under the near-same conditions and, thus, not significantly affected by minor changes.

5.4.6. Sub-Use Case 4.6

Req-ID	REQ 4.6-1
Name	Feasibility study and subsequent development of self-contained performance monitoring used by AI/ML DoS attack detection
Description	The sub use case 4.6 which is a teamwork between on the one hand AI/ML and DoS mitigation experts (i.e., CNIT, MONT) and TSS on the other hand. The objective is to first assess if the novel metrics derived from a control

Req-ID	REQ 4.6-1
	flow-based performance monitoring can be used to augment the F1 score or other model performance for DoS attacks detection. The first requirement is to assess objectivity, flair and pragmatism if the novel metrics can be of any use. For that, MONT and CNIT experts will be involved. A second requirement is to assess how these metrics can be easily collected, and the CPU associated costs of such collection. Last, a final requirement is to integrate an existing testbed implementing such AI-based DoS detection and implement the solution.
Leading Partner	TSS
Type	Functional
Priority	SHOULD
Validation sub-UC & testbed	The testbed will be defined according to the feasibility study and deeper discussion with partners
Mapping to services	S-14 aka SECaaS
Means of verification, methodology, tools	Simulated DoS attacks, F1 score monitoring with and without the method.

5.5. NATWORK Non-Functional Requirements

This section defines the non-functional requirements expected from the NATWORK system to operate maintaining certain qualities. It focuses on maintainability and interoperability, data management, and legal and ethical requirements.

5.5.1. Maintainability and Interoperability Requirements

Table 10: Requirements of Maintainability and Interoperability

Category	Description
ID:	NF-MI
Name	Modular system architecture
Description	All system components should be developed in a modular approach in order to allow easy modifications and enhancements of the offered functionalities.
UC#	All UCs
Leading partner	All UCs Partners
Requirement Type	NF: Non-Functional

Category	Description
Priority	M: Must-Have
Mapping to services	Services that have multiple individual components
Means of verification, methodology, tools	The modular system architecture will be considered achieved by demonstrating the ability to connect and integrate independent (separate) components inside a service or use independent (separate) components across multiple services

5.5.2. Data Management Requirements

Table 11: Requirements of FAIR Data

Category	Description
ID:	NF-DM
Name	Findable, accessible, interoperable, and re-usable (FAIR) data
Description	Datasets generated in NATWORK will be evaluated to determine whether they can be published with open access. Where such an evaluation is positive, datasets will be made available online via research data-sharing platforms like Zenodo. The work-in-progress source code will be openly shared on GitHub under Creative Common CCO license where internal and external contributions to any aspects are welcomed.
UC#	All UCs
Leading partner	All UCs Partners
Requirement Type	NF: Non-Functional
Priority	M: Must-Have
Means of verification, methodology, tools	The requirement is considered to be met if all the following criteria are successfully met. - Assign persistent identifiers (e.g. DOI, URN). - Tag project's results with Metadata. - Upload datasets to Zenodo. - Upload source code to GitHub.

5.5.3. Legal and Ethical Requirements

Table 12: Legal and Ethical Requirements

Category	Description
ID:	NF-LE
Name	Compliance with the project's ethics manual
Description	Compliance of developed components and AI mechanisms with the ethical principles and legal frameworks in Europe as well as laws and regulations in the partners countries.

Category	Description
UC#	All UCs
Leading partner	All UCs Partners
Requirement Type	NF: Non-Functional
Priority	M: Must-Have
Acceptance Criteria	The requirement is considered to be met if the following criteria are successfully met. - Compliance with the ethics manual described within D1.2 “Quality Assurance, Risk Management, Data Management Plan, Ethics & Regulatory issues”. - Collaboration with Ethical Committees to ensure that all research activities comply with Horizon Europe ethics rules and standards.

6. KVIs Evaluation

This section describes the KVIs defined to measure the strategic impact of the proposed use cases across multiple domains, including security, trust, sustainability, and innovation. The KVIs are presented in the following Table 13. There exist additional KVIs (A-KVIs), devised after the project has started. Each KVI is linked to specific Key Performance Indicators (KPIs) or Additional KPIs (A-KPIs), which offer quantitative metrics to evaluate technological progress and societal relevance. For example, KVIs related to environmental sustainability (e.g., KVI-1, KVI-3, KVI-15, KVI-19, and KVI-20) focus on reducing carbon footprint and digital waste through efficient CPU utilization and energy-aware mechanisms. KVIs such as KVI-2, KVI-4, KVI-5, and KVI-11 address aspects of trustworthy data processing, public protection, anti-jamming effectiveness, and secure IoT environments, using indicators like vulnerability exposure ratios, key generation compliance, and false positive/negative rates. Furthermore, advanced KVIs explore adaptive cybersecurity strategies (e.g., Moving Target Defence), offloading efficiency, and AI integration (KVI-13 to KVI-21), ensuring the 6G ecosystem is not only technically robust but also aligned with European values of privacy, reliability, and sustainability. These KVIs serve as a foundational framework for assessing the success and impact of each use case, from early experimentation (TRL 2–3) to lab-validated technologies (TRL 4–5).

Table 13: NATWORKS's KVIs and associated UCs and KPIs

KVI-ID	KVI name	UC	Type	Associated KPIs
1	Reduced Carbon footprint, reduced digital waste	1	Environmental Sustainability	**KPI 1.1** End-to-end compliance with latency tolerance **KPI 1.2** Energy waste: CPU utilization under normal/attack conditions to measure energy consumption (used to estimate Energy waste percentage)
2	Trustworthy and secured user data processing	1	Trust	**KPI 1.1** End-to-end compliance with latency tolerance **A-KPI 1.5** Cluster Hygiene Scores (Number of vulnerabilities shared with score 8+/Total number of vulnerabilities) **A-KPI 1.6** Cluster CTI Exposed information Ratio (Number of vulnerability data parts revealed/Total information per CTI data) **A-KPI 1.7** Cluster CTI Hidden information Ratio (Number of vulnerability data parts hidden/Total

KVI-ID	KVI name	UC	Type	Associated KPIs
				information per CTI data) **A-KPI 1.8** Denial of credentials of devices running non-trusted software. **A-KPI 1.9**: Additional latency of attestation below target value.
3	Efficient energy use, transitioning to green energy	1	Environmental Sustainability	**KPI 1.2:** Energy waste: CPU utilization under normal/attack conditions to measure energy consumption **KPI 1.3.4:** overall energy waste for the aggregation of confidentiality, integrity runtime verification and correct execution monitoring for x86 payloads
4	Public protection and disaster recovery	2	Trust	**KPI 2.4**: Downtime prevented **A-KPI 2.7**: Key Generation Length: Generation of 128-bit keys. **A-KPI 2.8** NIST Random Test Compliance: The generated keys will comply with the NIST random test suite.
5	Anti-jamming effectiveness	2	Trust	**KPI 2.2**: Time needed to detect and prevent a jamming attack **KPI2.3** Time needed to recover from a jamming attack **KPI 2.4**: Downtime prevented **KPI 2.5**: Throughput enhancement during jamming attack
6	Adaptability to new jamming attacks	2	Economical Sustainability & Innovation	**KPI 2.5**: Throughput enhancement during jamming attack
7	Jamming detection rate	2	Economical Sustainability & Innovation	**KPI 2.1**: Detection and mitigation of jamming attacks
8	Spectral efficiency improvement	2	Knowledge	**A-KPI 2.6**: Successful establishment of connectivity to avoid jammed channels/paths
9	Increased Level-of-Trust (LoT) for AVs	2	Trust	**KPI 2.1**: Detection and mitigation of jamming attacks **KPI 2.2**: Time needed to detect and prevent a jamming attack **KPI2.3** Time needed to recover from a jamming attack **KPI 2.5**: Throughput enhancement during jamming attack

KVI-ID	KVI name	UC	Type	Associated KPIs
10	Security and privacy issues related to the use of AI	2	Privacy & Confidentiality	**A-KPI 2.7**: Key Generation Length: Generation of 128-bit keys. **A-KPI 2.8** NIST Random Test Compliance: The generated keys will comply with the NIST random test suite. **A-KPI 2.9** Key Generation Rate (KGR): The rate of key generation will increase in proportion to the quality of the physical channel.
11	Trustworthy IoT network	3	Trust	**KPI 3.2** - Number of False Positives (FP) **KPI 3.3** - Number of False Negatives (FN) **KPI 3.4** - Packet Loss Ratio (PLR) **A-KPI 3.10** - Number of False Positives (FP) **A-KPI 3.11** - Number of False Negatives (FN) **A-KPI 3.12** - Trust Establishment Time (TET)
12	IoT service continuity	3	Privacy & Confidentiality	**KPI 3.1** - Mean Time to Detect (MTTD) **KPI 3.5** - Mean Time to Resolve (MTTR) **A-KPI 3.9** - Mean Time to Detect (MTTD)
A-13	Savings due to lower Attack Success Probability (ASP)	4.5	Privacy & Confidentiality	**A-KPI 4.13:** MTD action cost overhead [MACO] (worst-case) **A-KPI 4.15:** Protection gain of an MTD policy **A-KPI 4.17:** Decision Explainability for MTD [DEFM]
A-14	Business continuity assurance	4.5	Economical Sustainability & Innovation	**A-KPI 4.11:** Mean Time to implement the MTD action (MTID) **A-KPI 4.12:** Worst-case MTD service disruption [WMSD] **A-KPI 4.16:** Mean decision time for MTD action (MDTA)
A-15	Improved Carbon footprint	4.5	Environmental Sustainability	**A-KPI 4.14:** MTD green energy consumption [MGEC]
A-16	b5G/6G system Resilience	3.2	Privacy & Confidentiality	**A-KPI 3.2.1** - Impact on QoS by AI-DoS evaluation tool **A-KPI 3.2.2** - Comparison of results between AI-DoS and other tools used

KVI-ID	KVI name	UC	Type	Associated KPIs
				for QoS assessment, to determine which is the most effective tool. **A-KPI 3.2.3** - Perform a vulnerability report regarding DoS resilience on 5G/6G components.
A-17	b5G/6G system security	4.3 & 4.4	Privacy & Confidentiality	**A-KPI 4.6:** Jamming/adversary attacks mitigation (at least 80% accuracy in unjammed signal recovery) **A-KPI 4.7:** Time needed to mitigate a jamming/adversary attack via AI/ML frequency and protocol switching **A-KPI 4.8:** Time needed to recover from a jamming attack **A-KPI 4.9:** Downtime reduction **A-KPI 4.10:** Throughput improvement during jamming/adversary attack. **KPI 4.4:** Probability of DoS Attack Detection **KPI 4.5:** Probability of False Detection
A-18	Offloading system security and effectiveness	4.1	Privacy & Confidentiality	**KPI 4.1.1** DFE processing latency <50us with data plane device scalability up to 10k different flow rules **KPI 4.1.2** DFE computational efficiency 50% higher than existing methods (raw in-band telemetry) **KPI 4.1.4** WAI-based latency purely on hardware < 10 microseconds, latency on software-based WAI < 100 microseconds.
A-19	Improved offloading carbon footprint	4.1	Environmental Sustainability	**KPI 4.1.3:** DFE reduces power consumption by 20% compared to standard software-based feature selection and extraction at the computational engines **KPI 4.1.5:** 50% less power consumption compared to outsourced AI systems that run on cloud or edge nodes
A-20	Reduced Carbon footprint, reduced digital waste	4.2	Environmental Sustainability	**KPI 4.2.1:** Energy Efficiency Improvement: The AI-aware network slicing approach should reduce energy consumption significantly compared

KVI-ID	KVI name	UC	Type	Associated KPIs
				to traditional centralized AI model deployment. **KPI 4.2.3:** Resource Utilization: The network resource utilization should be optimized, with at least 50% of the AI model components running on underutilized network resources.
A-21	Offloading system security and effectiveness	4.2	Privacy & Confidentiality	**KPI 4.2.2:** Latency Reduction: The deployment of AI slices closer to the data plane should reduce end-to-end latency. **KPI 4.2.4:** AI Model Accuracy Maintenance: Despite the disaggregation, the AI model's accuracy should be maintained within 90% of the performance of the centralized model. **KPI 4.2.5:** Dynamic Reconfiguration Time: The time required to dynamically reconfigure AI slices to accommodate changes in network traffic should be under a few seconds.

7. Obstacles and Barriers

To assess the risk and potential deviation from targeted Use Case evaluation, we need to identify the possible obstacles and barriers. Table 14 below presents the template:

Table 14: UC Barrier template

ID	UC-<subUC or UC>-<number>
Name	
Leading Partner + contact person	
Severity	Does it affect the project? How? No deviation? Small deviation? Big?
Probability of the risk to happen	In a scale 0-5 (indicating from very unlikely to very likely to happen)
Description	
Actions to be taken to avoid the risk (if possible)	
Actions to be taken once the risk has happened	
Allocation of time, financial resources	

Some obstacles and barriers identified as of now and reported by the use case owners are presented in the following tables:

ID	UC-2.2-1
Name	Configuration of 5G scheduler
Leading Partner + contact person	GRAD
Severity	If the configuration of the scheduler is not feasible with the proposed architecture (BubbleRAN) would lead to a change in architecture or even in the testing of the algorithm with less realistic conditions, reducing the pertinence of the results
Probability of the risk to happen	In a scale 0-5 (indicating from very unlikely to very likely to happen): 3

ID	UC-2.2-1
Description	Changing the configuration of the 5G scheduler, even in open-source tools such as OAI, may be challenging or even impossible
Actions to be taken to avoid the risk (if possible)	Review different 5G architectures, like OAI, BubbleRAN, Amarisoft… to know if any of them let us achieve this.
Actions to be taken once the risk has happened	If the configuration is not possible, we can use a scenario by changing manually the frequency instead of using the scheduler. Perform the Action functionality at lower layers
Allocation of time, financial resources	Extra work to develop the manual change in frequency

ID	UC-2.2-2
Name	Utilization of dedicated SDRs
Leading Partner + contact person	GRAD
Severity	Would cause delay in multitude of tasks, like capturing signals to train and validate the algorithms
Probability of the risk to happen	In a scale 0-5 (indicating from very unlikely to very likely to happen): 2
Description	The lab environment has limited hardware resources, some of them might not be available all the time.
Actions to be taken to avoid the risk (if possible)	Save signals preventively to train and validate the detection phase. Planification of SDRs resources
Actions to be taken once the risk has happened	N/A
Allocation of time, financial resources	Delay in tasks while the hardware is not available

ID	UC-2.2-3
Name	Problems with the Signals database
Leading Partner + contact person	GRAD

ID	UC-2.2-3
Severity	Can reduce the quality of the algorithms
Probability of the risk to happen	In a scale 0-5 (indicating from very unlikely to very likely to happen): 2
Description	Training the AI model requires enough quality signals, both legitimate UE and jamming.
Actions to be taken to avoid the risk (if possible)	Planification of signals database creation and searching synergies with other NATWORK's partners. Simulate signals to obtain preliminary results
Actions to be taken once the risk has happened	Obtain more appropriate signals from the lab.
Allocation of time, financial resources	Time needed to save more signals and retrain the algorithms.

ID	UC-2.4-1
Name	Challenges in Channel Estimation Beyond Simulation
Leading Partner + contact person	GRAD
Severity	Important barrier that could reduce the KGR and raises the KDR
Probability of the risk to happen	In a scale 0-5 (indicating from very unlikely to very likely to happen): 4: likely to happen
Description	The AI CSI optimization network is initially trained on synthetic channel data generated by QuaDRiGa. Real sub-THz captures exhibit harsher path-loss, RF front-end impairments and time-varying reciprocity errors, causing a distribution shift.
Actions to be taken to avoid the risk (if possible)	Develop the AI model accordingly to fast adaptation: keep early layers frozen and fine-tune only the last layers.
Actions to be taken once the risk has happened	Fine-tune the existing model with laboratory traces to adapt it to the new conditions. Or in the worst scenario fully retrain the model from scratch if fine-tuning alone does not restore target performance.
Allocation of time, financial resources	Some extra work might be needed to retrain the model in the worst scenario.

ID	UC-3.2-1
Name	Lack of datasets to evaluate the effectiveness of the detection of anomalies
Leading Partner + contact person	MONT
Severity	Important barrier that could reduce the pertinence of the results
Probability of the risk to happen	In a scale 0-5 (indicating from very unlikely to very likely to happen): 5: likely to happen
Description	To train the AI-based detection models and test their effectiveness, a large quantity of quality datasets is needed. This is a recurring problem for all ML-based solutions.
Actions to be taken to avoid the risk (if possible)	Use of a set of open-source datasets and generation of new datasets using MONT's: - **TAS** (Test and Simulation) platform that allows creating many virtual IoT devices, i.e., Digital Twins that send and receive artificially generated IoT communications where anomalies and attacks can be injected; - **5greplay** tool that allows replaying 5G network traffic (i.e., sending traffic to the gNodeB or the 5G core, but not wireless traffic) where anomalies and attacks can be injected.
Actions to be taken once the risk has happened	TAS and 5greplay will be configured to produce normal and modified network traffic.
Allocation of time, financial resources	Some extra work might be needed to generate network traffic.

ID	UC-4.1-1
Name	New incoming attacks classification delay
Leading Partner + contact person	CNIT
Severity	Important barrier that could reduce the pertinence of the results
Probability of the risk to happen	In a scale 0-4 (indicating from very unlikely to very likely to happen): 3: slightly likely to happen
Description	The performance of blocking attacks at the data plane using DFE and WAI may be inhibited by incoming new attacks for which there is not a model to apply. Training would need too much time to address the issue
Actions to be taken to avoid	Use different recognition methods: 1) apply DFE/WAI for the incoming attack 1 (model is available). 2) Continue to monitor the traffic using DFET.

ID	UC-4.1-1
the risk (if possible)	3) Send DFET reports to an external Attack Detector. 4) When attack 2 is identified, ask the security orchestrator to take defensive action at the data plane. 5) Configure the same or another backend to block attack 2.
Actions to be taken once the risk has happened	N/A
Allocation of time, financial resources	Some extra work might be needed to include two attack models.

ID	UC-4.2-1
Name	AI slice reconfiguration lag under highly dynamic traffic
Leading Partner + contact person	ELTE
Severity	Important risk that could degrade real-time responsiveness and violate service requirements
Probability of the risk to happen	In a scale 0-4 (indicating from very unlikely to very likely to happen): 2: unlikely but possible under heavy or unpredictable traffic loads
Description	AI/ML model slices may not be reconfigured quickly enough in response to rapid traffic changes. This can lead to degraded inference accuracy.
Actions to be taken to avoid the risk (if possible)	Test the reconfiguration thresholds with multiple different datasets.
Actions to be taken once the risk has happened	Log the incident, analyse bottlenecks, and refine reconfiguration thresholds.
Allocation of time, financial resources	Some development effort and test cycles are needed to fine-tune the reconfiguration logic.

ID	UC-4.5-1
Name	Obstacles in the solution integration with Telco Cloud CNFs
Leading Partner + contact person	ZHAW
Severity	Important barrier that could reduce the pertinence of the results
Probability of the risk to happen	In a scale 0-5 (indicating from very unlikely to very likely to happen): 2: unlikely to happen but possible

ID	UC-4.5-1
Description	The MTD solution is being developed and tested against generic cloud native services with the objective of being agnostic from the protected NFs, but some 5G and 6G CNFs could have elements (e.g. specific communication protocols) that MTD actions do not handle yet.
Actions to be taken to avoid the risk (if possible)	Telco Cloud CNFs are numerous and with different properties. Results could be shown on CNFs that do permit MTD actions.
Actions to be taken once the risk has happened	Identify which elements in a CNF render MTD actions unusable and whether a solution could be found.
Allocation of time, financial resources	Some extra work might be needed to find or implement a CNF that enables all features of the MTD framework.

ID	UC-4.5-2
Name	Performance evaluation limits due to 5G testbed size
Leading Partner + contact person	ZHAW
Severity	Would not allow to assess the usability of the MTD framework in a large-scale telecommunication network.
Probability of the risk to happen	In a scale 0-5 (indicating from very unlikely to very likely to happen): 5: very likely to happen
Description	The local 5G testbed used for evaluation has limited hardware resources and is limited in size. This would make for very limited scalability tests of the solution.
Actions to be taken to avoid the risk (if possible)	This barrier is hard to avoid as it requires heavy investment in hardware and/or rental of a very high number of resources from a telecommunication network, which hardly agrees to enable the setup required for the scalability test to take place
Actions to be taken once the risk has happened	The action to take is to test the MTD framework on scenarios where the network is realistically small, e.g., a private 5G network limited to the sites of an institution or industrial factories.
Allocation of time, financial resources	Increase in the financial resources used to increase the size of a testbed to a minimum realistic size and/or to include special hardware (e.g. TEE capable servers).

After reviewing the obstacles and barriers some of them refer to:

- Reconfiguration of the 5G scheduler could be resolved by sharing efforts. Some of the partners working on the jamming device may have considered the same solution, xApp to control the scheduler, to resolve jamming attacks. IS-Wireless, CERTH, HES-SO and GRAD. That also may apply to USRPs or SDRs. Individuals may have limited resources, so gathering for testing purposes should also be promoted.
- Not enough data. This could be solved with better cooperation between partners. For instance, CERTH, GRAD and IS-WIRELESS work on jamming detection and mitigation. Indeed, each one has a particular objective but, in the process, they may collect data useful for another partner.
- Regarding the size of the testbed, the possibility of renting a bigger testbed eventually could also be considered or request the use of a bigger installation when the higher solution wants to be tested.

These are the obstacles and barriers collected by a preliminary analysis of the pilot requirements and target KPI and KVI. Further obstacles can be identified during the components and services development stage and during the integration phase. Further analysis and mitigation will be undertaken in conjunction with the global risks collected in NATWORK Risk Registry in WP1.

8. Conclusions

This deliverable marks a significant step in preparing NATWORK for its upcoming evaluation phase. By translating high-level design and requirements into detailed validation plans and pilot configurations, it provides a solid operational foundation for real-world testing.

The defined evaluation framework ensures that the project's key performance and validation indicators are not only measurable but also closely aligned with the technical objectives of each use case. The inclusion of testbed-specific scenarios and mappings enhances traceability, while the mapping of KPIs to KVIs ensures that NATWORK's goals will be fulfilled.

Importantly, the deliverable anticipates potential evaluation challenges by establishing a methodology for identifying and addressing obstacles, promoting resilience and flexibility across pilot deployments.

The core work of T6.1, as reported in this deliverable, will pave the way for and support the activities performed in WP6. The evaluation strategy devised will be serve as guidelines for the processes of T6.4 "System Validation and Evaluation" and its respective deliverable 'D6.4 - System validation, End-user evaluation & Lessons Learnt Alliances'.

References

[1] Ejaz, S., & Al-Naday, M. (2024, March). FORK: A Kubernetes-compatible federated orchestrator of fog-native applications over multi-domain edge-to-cloud ecosystems. In *2024 27th Conference on Innovation in Clouds, Internet and Networks (ICIN)* (pp. 57–64). IEEE.

[2] Ettus Research. (n.d.). *UB210 Kit*. https://www.ettus.com/all-products/ub210-kit/

[3] NVIDIA. (n.d.). *Jetson Orin*. https://www.nvidia.com/en-eu/autonomous-machines/embedded-systems/jetson-orin/

[4] TMYTEK. (n.d.). *XRifle dynamic RIS*. https://tmytek.com/products/components/xrifle-dynamic-ris

[5] BubbleRAN. (n.d.). *Transforming the future of connectivity* [Corporate website]. Retrieved May 20, 2025, from https://bubbleran.com/

[6] Bassham, L. E., Rukhin, A. L., Soto, J., Nechvatal, J. R., Smid, M. E., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, N., Dray, J., & Vo, S. (2010). *A statistical test suite for random and pseudorandom number generators for cryptographic applications* (NIST Special Publication 800-22 Rev. 1a). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-22r1a

[7] Musumeci, F., Tornatore, M., & Pattavina, A. (2022). Machine-learning-enabled DDoS attacks detection in P4 programmable networks. *Journal of Network and Systems Management, 30*(1), 1–27. https://doi.org/10.1007/s10922-021-09633-5

[8] K3s Contributors. (n.d.). *K3s: Lightweight Kubernetes*. https://k3s.io/

[9] SIG Multicluster. (n.d.). *Multi-cluster service APIs*. Retrieved December 24, 2024, from https://github.com/kubernetes-sigs/mcs-api

[10] Submariner Contributors. (n.d.). *Submariner K8s project documentation website*. Retrieved December 24, 2024, from https://submariner.io/