Smart, Automated, and Reliable Security Service Platform for 6G

Deliverable D6.2
Intermediate Validation Results

ROBUST-6G project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101139068.

Date of delivery:	30/09/2025	Version:	1.0
Project reference:	101139068	Call:	HORIZON-JU-SNS-2023
Start date of project:	01/01/2024	Duration:	30 months

Document properties:

Document Number:	D6.2
Document Title:	Intermediate Validation Results
Editor(s):	Lucía Cabanillas, Ignacio Domínguez, Riccardo Nicolicchia (TID)
Authors:	Listed below
Contractual Date of Delivery:	30/09/2025
Dissemination level:	PU
Status:	Final
Version:	1.0
File Name:	ROBUST-6G D6.2_v1.0

Revision History

Revision	Date	Issued by	Description
v0.1	06.05.2025	ROBUST-6G WP6	Initial draft of ToC
v0.1	23.07.2025	ROBUST-6G WP6	ToC structure changed
v0.2	8.07.2025	ROBUST-6G WP6	Internal review finalized
v0.3	19.09.2025	ROBUST-6G WP6	External review finalized
v1.0	26.09.2025	ROBUST-6G WP6	Final version finalized

Abstract

This deliverable outlines the operational execution of the validation strategy set out in D6.1 for the ROBUST-6G project. It details the intermediate results obtained through a structured, flow-based methodology that enables the progressive integration and testing of project components, flows, and scenarios. The validation activities cover three key use cases: the trustworthiness of AI models in distributed 6G environments, automatic threat detection and mitigation in IoT systems, and the secure exposure of network capabilities via Network Security as a Service (NetSecaaS). Each use case has been broken down into functional flows designed to be validated using dedicated Partner Testbed Assets (PTAs) against predefined Key Performance Indicators (KPIs). The document summarises the technical protocols, testbed configurations, and validation criteria employed to evaluate the robustness, scalability, and security of the ROBUST-6G platform. These intermediate results confirm the feasibility of the proposed architecture and inform the roadmap for the final validation activities set out in future deliverables.

Keywords

ROBUST-6G Validation Plan, ROBUST-6G Components, Use Case Validation, Unified Testbed Design, Component Validation, Performance Metrics

Disclaimer

Funded by the European Union. The views and opinions expressed are however those of the author(s) only and do not necessarily reflect the views of ROBUST-6G Consortium nor those of the European Union or Horizon Europe SNS JU. Neither the European Union nor the granting authority can be held responsible for them.

List of Contributors

Participant	Short Name	Contributors
Ericsson Araştırma Geliştirme ve Bilişim Hizmetleri A.Ş	EBY	Leyli Karaçay
Universidad de Murcia	UMU	Alberto García Pérez, Fernando Torres Vega, Enrique Tomás Martínez Beltrán, José María Jorquera Valero, Manuel Gil Pérez
University College Dublin	UCD	Bartlomiej Siniarski
University of Padova	UNIPD	Stefano Tomasin, Giovanni Perin
Nextworks	NXW	Enrico Alberti, Marco Ruta, Pietro G. Giardina
ENSEA	ENSEA	Arsenia Chorti, Luan Chen, Sotiris Skaperas, Mamady Delamou
GOHM Elektronik ve Bilişim San. Tic. Ltd. Şti.	GOHM	Cem Ayyildiz, Fatih Emre Yıldız
Axon Logic	AXON	Chih Yang Pee, Wei Chuen Yau, Su Fong Chien, Charilaos Zarakovitis

List of Reviewers

Participant	Short Name	Reviewers	Phase
University of Murcia	UMU	Alberto García Pérez, José María Jorquera Valero	Internal
Ericsson Araştırma Geliştirme ve Bilişim Hizmetleri A.Ş	EBY	Mustafa Akdeniz	External
Ericsson Araştırma Geliştirme ve Bilişim Hizmetleri A.Ş	EBY	Ramin Fuladi	Final

Executive Summary

This deliverable presents the intermediate validation results of the ROBUST 6G project, based on the framework and methodology defined in Deliverable D6.1. While D6.1 established the validation plan and testbed design, D6.2 focuses on executing that plan operationally, reporting on progress in validating individual components, functional flows and initial scenario integrations across the Unified ROBUST 6G Testbed.

The approach's foundation is the three-step, flow-based integration pipeline outlined in D6.1, which guarantees traceability from component-level validation to full scenario execution. This deliverable details how this methodology has been applied in practice, highlighting the following key aspects:

- **Formalisation of validation procedures**: D6.2 establishes and implements the procedures for validating components, flows and scenarios via standardised documentation and interface specifications. It provides clear guidelines on how components are tested and how they interact within flows. It also explains how results are recorded and reviewed, ensuring consistency, traceability and reproducibility across all validation activities.
- **Unified Testbed Federation:** This deliverable reports on the federation of partner testbed assets (PTAs) into a cohesive, distributed environment capable of supporting multi-domain integration. Connectivity requirements, deployment strategies and containerisation practices have been consolidated to enable seamless cross-partner testing.
- **Intermediate Validation Outcomes:**
  - Component-level: Many project components have undergone standalone validation against predefined KPIs to confirm their functional correctness and readiness for integration.
  - Flow-Level: Some functional flows have been deployed and tested to demonstrate interoperability and compliance with interface specifications.
  - Scenario-Level: The initial end-to-end validations for the selected scenarios within the three project use cases have begun, providing early evidence of architectural coherence and functional robustness.
- **KPI Monitoring and Gaps:** While several KPIs, such as detection accuracy, mitigation time and API latency, have been partially validated, others remain under assessment due to ongoing integration or dependency on external components. These gaps, along with the interoperability challenges that have been identified, are being documented in order to guide the next phase.

In summary, D6.2 confirms the viability of the ROBUST 6G validation strategy and demonstrates tangible progress towards the project's objectives. The results reported here validate the soundness of the flow-based methodology and the effectiveness of the unified testbed federation. They also demonstrate that the core components and flows are ready for final integration. D6.3 will build on these achievements by carrying out scenario-level validations, refining KPI measurements and resolving the issues raised in this report. This will ensure the delivery of a secure, scalable and ROBUST 6G system.

Table of Contents

1 Introduction 11
1.1 Objective of the document 11
1.2 Structure of the document 11

2 Methodological Framework for Validation 12
2.1 Scope of Updates 12
2.2 Unified Testbed Federation 13
2.3 Operationalisation of KPIs 14
2.4 Gap Analysis and Alignment with the Architecture 15

3 UC1: AI model trustworthiness evaluation for 6G distributed scenarios 16
3.1 Scenario 1 - Decentralized federated learning for joint privacy-preserving ML/DL model training 17
3.1.1 Functional Flows description and mapping 18
3.1.2 Testbed Requirements and Deployment 30
3.1.3 KPIs and Validation Criteria 31
3.1.4 Flow Progress Tracking 32
3.2 Scenario 2 - Physical and sensing layer trustworthiness and resilience 33
3.2.1 Functional Flows description and mapping 33
3.2.2 Testbed Requirements and Deployment 39
3.2.3 KPIs and Validation Criteria 40
3.2.4 Flow Progress Tracking 40

4 UC2: Automatic threat detection and mitigation in 6G-enabled IoT environments 42
4.1 Scenario 1 - Device violation to cause an economic harm (a) 42
4.1.1 Functional Flows description and mapping 43
4.1.2 Testbed Requirements and Deployment 50
4.1.3 KPIs and Validation Criteria 52
4.2 Scenario 2 - Fraudulent usage of device resources 53
4.2.1 Functional Flows description and mapping 53
4.3 Scenario 3 - Device violation to cause an economic harm (b) 56
4.3.1 Functional Flows description and mapping 56
4.3.2 Testbed Requirements and Deployment 61
4.3.3 KPIs and Validation Criteria 62
4.4 Flow Progress Tracking 62

5 UC3: Security Capabilities Exposure with Network-Security-as-a-Service (NetSecaaS) 63

5.1 Scenario Summary 63
5.1.1 Functional Flows description and mapping 64
5.1.2 Testbed Requirements and Deployment 71
5.1.3 KPIs and Validation Criteria 72
5.1.4 Flow Progress Tracking 73

6 Conclusions and Next Steps 74

List of Tables

Table 3-1 ROBUST-6G Components implementing UC1 - Scenario 1 19
Table 3-2:Use Case 1 Scenario 1 Flow Progress Tracking Table 32
Table 3-3 ROBUST-6G Involved Components implementing UC1 - Scenario 2 34
Table 3-4:Use Case 1 Scenario 2 Flow Progress Tracking Table 40
Table 4-1: ROBUST-6G Components implementing UC2 - Scenario 1 44
Table 4-2: UC2 Key Performance Indicator 52
Table 4-3: ROBUST-6G Components implementing UC2 - Scenario 3 57
Table 4-4: Use Case 2 Flow Progress Tracking Table 62
Table 5-1 Security capabilities available 65
Table 5-2 : Use Case 3 Flow Progress Tracking Table 73

List of Figures

Figure 2-1: Unified Testbed overview 14
Figure 3-1 Architecture mapping of UC1_1 18
Figure 3-2 Components for the demonstration of the main functionalities of the use case 19
Figure 3-3 Collaborative and privacy-preserving model training flow diagram, benign nodes 22
Figure 3-4 Sequence diagram for evaluating DFL system's robustness 24
Figure 3-5 Sustainable and efficient evaluation of the model lifecycle 25
Figure 3-6 Continuous monitoring of explainability within the federated training lifecycle 27
Figure 3-7 Privacy-enhanced collaborative model training flow diagram 29
Figure 3-8 Architecture mapping of UC1_2. Red squares indicate the involved functionalities 34
Figure 3-9: High-level description of the functional flows and their involved components within the physical layer closed loop 36
Figure 3-10:PHY layer trustworthiness evaluation flow diagram 36
Figure 3-11:Mutual authentication flow diagram 37
Figure 3-12:(Fast) Secret key agreement flow diagram 38
Figure 3-13:Inputs of NYUsim channel model simulator 39
Figure 3-14:Diagram of the integration of deployable and simulated -based components 39
Figure 4-1: ROBUST-6G architecture validated in UC2 Scenario 1 44
Figure 4-2: Proactive Security Deployment (UC2.1 - flow1) 47
Figure 4-3: CL-Monitoring and CL-Analysis functions run (UC2.1 - flow2) 48
Figure 4-4: CL-Decision and CL-Execution functions run (UC2.1 – flow3 49
Figure 4-5: Testbed implementing UC2 - Scenario 1 51
Figure 4-6: Investigative loop deployment and execution (UC2.2 loop 1) 54
Figure 4-7: Resolutive loop deployment and execution (UC2.2 loop 2) 55
Figure 4-8: ROBUST-6G architecture validated in UC2 Scenario 3 57
Figure 4-9: Internal loop for field A/B (UC2.3 loop1) 59
Figure 4-10: External loop for verification (UC2.3 loop2) 60
Figure 4-11: Planned testbed implementing UC2 - Scenario 61
Figure 5-1: UC3 scenario overview 64
Figure 5-2: General Security Capability exposure scenario 65
Figure 5-3: UC3 architecture mapping 67
Figure 5-4: Access Governance data exposure flow 68
Figure 5-5: Security Capabilities discovery data exposure flow 69
Figure 5-6: XAI Analytics Data flow 70
Figure 5-7: SSLA simplification process flow 71
Figure 5-8:UC3 testbed planification 72

Acronyms and abbreviations

Term	Description
6G	Sixth Generation (wireless networks)
AI	Artificial Intelligence
API	Application Programming Interface
AoA	Angle of Arrival
AKA	Authentication and Key Agreement
CAMARA	Refers to CAMARA project for APIs
CL	Closed Loop
CSI	Channel State Information
DFL	Decentralized Federated Learning
DL	Deep Learning
DoA	Description of Action
ETSI	European Telecommunications Standards Institute
GMR	Global Model Repository
GNSS	Global Navigation Satellite System
IDS	Intrusion Detection System
IoT	Internet of Things
IP	Integration Point
KPI	Key Performance Indicator
LoS/NLoS	Line of Sight / Non-Line of Sight
LSTM	Long Short-Term Memory
ML	Machine Learning
M2M	Machine-to-Machine
NOMA	Non-Orthogonal Multiple Access
PHY	Physical Layer
PLA	Physical Layer Authentication
PLS	Physical Layer Security
PTA	Partner Testbed Asset

QoS	Quality of Service
RAN	Radio Access Network
RF	Radio Frequency
SDN	Software Defined Networking
SKG	Secret Key Generation
SNN	Spiking Neural Network
SSLA	Security Service Level Agreement
TBC	To Be Confirmed
UC	Use Case
XAI	Explainable Artificial Intelligence

1 Introduction

This deliverable provides a detailed report on the validation activities conducted within the ROBUST-6G project since the publication of D6.1. It summarises the progress made in implementing and testing the validation plan, with a particular focus on executing functional flows, integrating components and carrying out a preliminary assessment of key performance indicators (KPIs) across the defined use cases.

Building directly on the methodology and testbed design introduced in D6.1 'Use Case Validation Plan and Testbed Design' [ROB25-D61] this document moves from planning to practical execution. While D6.1 defined the validation framework and the unified testbed concept, D6.2 reports on the outcomes of these activities so far, highlighting the current integration status, results and challenges.

This deliverable is structured to provide transparency and traceability of the validation process. It includes a description of the applied methodological framework, validation progress for each use case and scenario, and preliminary KPI measurements. It also identifies gaps, dependencies and lessons learned to inform the final validation phase, which will be reported in D6.3.

1.1 Objective of the document

The main objective of this document is to report the intermediate validation results of the ROBUST-6G project. Specifically, it aims to:

- Present the progress achieved in validating the use cases and scenarios defined in D6.1.
- Describe the methodologies, tools, and testbeds employed for component, flow, and scenario validation.
- Provide preliminary KPI measurements and analyse their alignment with the targets defined in the Description of Action (DoA).
- Highlight integration challenges, gaps, and lessons learned during this phase.
- Serve as a reference for the final validation activities to be reported in D6.3.

This deliverable does not redefine the overall validation strategy but rather complements D6.1 by focusing on the operational aspects and early outcomes of the validation process.

1.2 Structure of the document

The document is organized as follows:

- Section 2 – Methodological Framework for Validation: Summarizes the validation methodology, including the multi-level approach (component, flow, and scenario validation) and the role of the unified testbed.
- Section 3 – Use Case 1: AI Model Trustworthiness Evaluation for 6G Distributed Scenarios: Details the validation activities and intermediate results for UC1, covering decentralized federated learning and physical layer trustworthiness.
- Section 4 – Use Case 2: Automatic Threat Detection and Mitigation in 6G-Enabled IoT Environments: Reports the validation progress for UC2, focusing on closed-loop security workflows and their performance under different scenarios.
- Section 5 – Use Case 3: Security Capabilities Exposure with Network-Security-as-a-Service (NetSecaaS): Presents the validation of exposure mechanisms and APIs for secure interaction with third-party applications.
- Section 6 – Conclusions and Next Steps: Summarizes key findings, outlines the plan for final validation, and identifies open issues to be addressed in D6.3.

2 Methodological Framework for Validation

This chapter outlines the methodological refinements that guide validation activities at the current stage of the ROBUST-6G project. While Deliverable D6.1 established the overall validation framework, including the multi-level approach and testbed catalogue, the purpose here is not to restate that baseline. Instead, the focus is on the delta: the specific methodological updates introduced during the execution phase covered by D6.2. These updates reflect the project's progression from planning towards integration and validation, with emphasis on three aspects: (i) the adoption of flows as the central validation unit, linking components to scenarios through well-defined inputs, interactions, and expected outcomes; (ii) the transition from isolated Partner Testbed Assets (PTAs) to a federated, unified testbed supporting cross-partner execution; and (iii) the operationalisation of KPIs and the introduction of a systematic gap analysis to maintain alignment with the architecture defined in D2.2. Together, these refinements demonstrate that although full KPI measurements are not yet available, the consortium has established the structures, artefacts, and processes needed to ensure consistent and auditable validation in the next phase.

2.1 Scope of Updates

The scope of this chapter is limited to reporting on the methodological updates introduced since Deliverable D6.1. The baseline validation plan, including the description of validation levels and the catalogue of Partner Testbed Assets (PTAs), is already documented in D6.1 [ROB25-D61] and is not repeated here. Instead, D6.2 highlights the refinements that have been applied as the project moves from planning into execution.

These updates can be summarised as follows:

- **Flow-based validation:** Flows have been established as the primary unit of validation, extending beyond isolated component testing to cover end-to-end interactions. Each flow definition now includes not only inputs and integration points, but also the expected outcomes linked to specific KPIs.
- **Unified testbed federation:** The focus has shifted from cataloguing individual PTAs to specifying and realising their federation into a cohesive environment. This step is essential for enabling flows that span multiple partners and technical domains.
- **Operationalisation of KPIs:** Project-level KPIs have been defined with clear initiation and completion criteria, allowing for consistent tracking and data collection once the workflows are executed.
- **Gap analysis against D2.2:** A systematic process has been introduced to map flows to architectural functions, identify missing or misaligned elements, and document gaps for resolution in subsequent iterations.

By concentrating on these deltas, the chapter demonstrates how the consortium has progressed from abstract planning towards practical readiness. The absence of complete numerical results at this stage is mitigated by the delivery of structured flows, measurable KPI definitions, and a federated testbed framework that together provide a transparent and auditable pathway towards the final validation activities reported in D6.3. Flow-Based Validation Approach.

A major methodological refinement in this phase of the project is the adoption of flows as the central unit of validation. While D6.1 emphasised the classification and standalone testing of components, D6.2 shifts the focus to how these components interact within the context of use cases and scenarios. A flow represents a chain of interactions between multiple components that collectively realise a specific functionality within a use case. Each flow can be seen as a self-contained slice of the architecture, combining validated components, integration points, and testbed resources. For D6.2, each flow

description has been extended to cover not only technical inputs but also the expected outcomes. The common elements now include:

- **Inputs and prerequisites:** involved components, integration points, and initial configuration.
- **Interactions:** the sequence of steps or messages exchanged across the components.
- **Expected outputs:** the artefacts or observable behaviours expected from the flow (e.g., a trained model, a detection alert, or a mitigation action).
- **Linked KPIs:** the specific performance indicators associated with the flow, that will be applied once executions take place.

This approach ensures that validation is no longer confined to proving that components work in isolation, but that they operate coherently when combined. Even if some flows have not yet been executed due to component readiness, the process of defining their inputs, expected outputs, and KPI linkage already provides value. It allows early identification of dependencies, clarifies testbed requirements, and ensures traceability to the architecture defined in D2.2.

2.2 Unified Testbed Federation

Another key evolution since D6.1 is the transition from a static catalogue of Partner Testbed Assets (PTAs) to the establishment of a federated testbed environment, shown in Figure 2-1. The Figure 2-1 illustrates the current deployment of Partner Testbed Assets, the integration of individual components, and their grouping into capability clusters such as Distributed Federated Learning, Privacy-Enhanced Services, Sustainable AI, Threat Mitigation, and Orchestration frameworks. The diagram also highlights how these assets and components are mapped onto the different use cases, serving as a blueprint for validation activities and ensuring consistency across partner contributions. This unified design provides a consolidated view of the testbed architecture and its role in supporting use-case–driven experimentation and evaluation. In fact, whereas D6.1 documented the available assets and their individual capabilities, D6.2 reports on the progress made towards interconnecting them into a cohesive and distributed validation platform. The goal of federation is to enable flows that span multiple partners and technical domains, ensuring that validation reflects realistic multi-stakeholder 6G environments. A federated testbed allows components hosted in different PTAs to interact as if they were co-located, while still preserving local autonomy and security.

The federation design is based on three methodological principles:

- **Connectivity:** PTAs must support secure, standardised inter-partner networking, allowing remote execution of flows without compromising data or infrastructure.
- **Deployment flexibility:** containerisation and orchestration mechanisms are encouraged to simplify replication and reduce integration overheads across heterogeneous environments.
- **Exposure of resources:** each PTA is expected to expose its assigned components through well-defined interfaces, ensuring that flows can be instantiated and executed without manual reconfiguration.

At this stage, the main requirements for federation have been identified and implementation has begun. Several PTAs are already capable of hosting project components in a containerised manner and exposing them over secure connections. Flow definitions have been mapped to specific PTAs, clarifying which environments are responsible for hosting which parts of the architecture. By moving beyond individual testbed catalogues towards a federated environment, D6.2 demonstrates that the project is laying the operational foundation for executing cross-partner flows and scenario-level validations in the next phase.

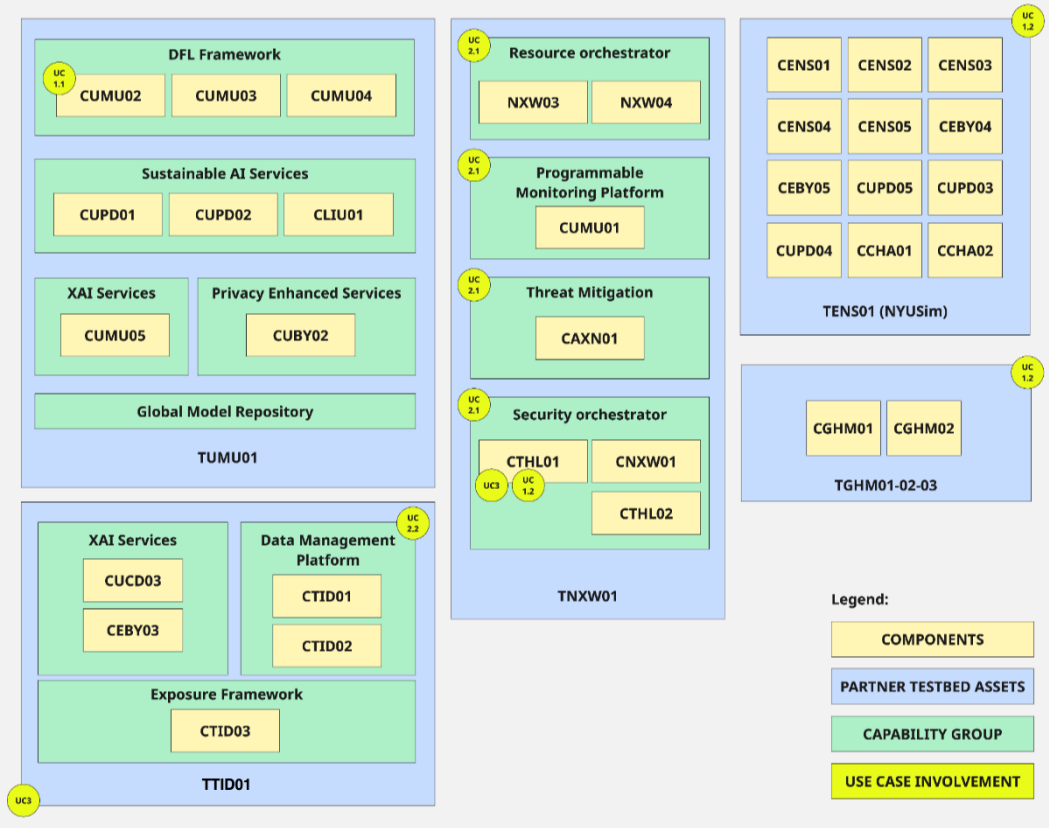

Figure 2-1: Unified Testbed overview

2.3 Operationalisation of KPIs

A further refinement introduced in D2.2 and D6.2 is the translation of project-level KPIs into measurable and executable definitions. In D6.1, a subset of KPIs were identified as high-level indicators to guide validation (e.g., detection accuracy, mitigation time, model trustworthiness). In this deliverable, these abstract metrics have been operationalised so they can be applied consistently once flows are executed. Each KPI is defined not only by its target value, but also by the procedure used to measure it. This ensures comparability across partners and reproducibility of results. The operationalisation is based on three methodological elements:

- **Measurement criteria:** precise definition of what is being quantified and how it is calculated.
- **Start event:** the log entry, API call, or system trigger that signals the beginning of the measurement.
- **Stop event:** the log entry, API call, or system trigger that marks the end of the measurement.

While complete numerical results are not yet available for most flows, the measurement criteria and evidence collection mechanisms have already been defined. In some cases, preliminary or partial executions have been carried out, providing early indications of feasibility. These definitions will allow the consortium to collect results in a systematic and auditable way during the next validation phase. By introducing these detailed criteria now, the project ensures that once integration reaches maturity, validation will not require additional methodological alignment. Instead, results will be directly comparable across testbeds and scenarios, and traceable to the architecture defined in D2.2. This provides transparency for reviewers and ensures that the final validation (D6.3) will deliver not only numbers but also verifiable evidence of how those numbers were obtained.

2.4 Gap Analysis and Alignment with the Architecture

To ensure architectural alignment and readiness for validation, D6.1 introduced the structured Gap Analysis methodology and D6.2 shows the gap analysis results focused on detecting and resolving inconsistencies between the practical implementation and the reference architecture defined in D2.2. This process acts as an early warning system, enabling proactive identification of issues that could compromise scenario-level outcomes and guiding corrective actions before they escalate.

The gap analysis is organized into three key steps:

1. **Mapping Flows to Architectural Components:** Each functional flow is systematically linked to the corresponding functions defined in D2.2. This ensures that the implementation accurately reflects the planned capabilities and that no critical elements are overlooked.
2. **Cross-Scenario Interaction Checks:** Flows are analysed in combination to verify that interdependencies between use cases and scenarios are properly captured. This step confirms the technical feasibility of integration and highlights any sequencing or coordination challenges.
3. **Gap, Overlap, and Ambiguity Identification:** The analysis flags missing components or functional blocks, duplicated responsibilities, and unclear interface specifications. Each issue is documented internally and assigned to the relevant partners for resolution, ensuring accountability and traceability.

This structured approach transforms architectural validation from a static checkpoint into a dynamic, iterative process embedded throughout the project lifecycle. It complements the qualitative assessment with a systematic framework that supports transparency, consistency, and readiness for the final validation phase.

At this stage of the gap analysis, the review covered the following scenarios and flows:

- UC1.1: Decentralized federated learning: 5 flows examined
- UC1.2: Physical and sensing layer trustworthiness: 3 flows examined
- UC2.1: Device violation to cause economic harm (a): 3 flows examined
- UC2.2: Fraudulent usage of device resources: 2 flows examined
- UC2.3: Device violation to cause economic harm (b): 2 flows examined
- UC3: Security Capabilities Exposure (NetSecaaS): 4 flows examined

In total, 19 flows across three use cases were examined.

At this stage, the analysis has identified the following items affecting architectural alignment and validation readiness:

- **Use Case 1.1:** Flows are well aligned with the reference architecture; no major gaps identified. The Global Model Repository is consistently positioned within the AI Service Management Layer; an open point remains regarding potential interactions with external consumers.
- **Use Case 1.2:** Flows are well aligned with the reference architecture; no major gaps identified.
- **Use Case 2:** Flows are technically consistent; clarifications are required for closed-loop data interactions and the harmonization of monitoring responsibilities across IoT and RAN layers.
- **Use Case 3:** Flows are consistent with the reference architecture; an open point remains on the identification of all the modules that can be exposed, to be clarified in the next iteration.

All identified issues are systematically logged in to ensure traceability. Each entry includes:

- **Reference to Affected Flows:** Clear mapping of the impacted flows or processes to understand the scope and implications.
- **Involved Partners:** Identification of the responsible or affected stakeholders to facilitate ownership and collaboration.
- **Corrective Actions:** Defined steps, timelines, and responsible parties for resolution, ensuring that mitigation measures are actionable and trackable.

While the current assessment is primarily qualitative rather than quantitative, the introduction of a systematic gap analysis process marks a significant improvement. It transforms architectural alignment from a one-off milestone into a continuous, iterative process embedded in the project lifecycle. This approach delivers key benefits:

- **Transparency:** Provides reviewers and stakeholders with a clear, real-time view of progress, challenges, and mitigation plans, enabling informed decision-making.
- **Consistency:** Ensures that the architecture evolves in a coherent and controlled manner, reducing the risk of misalignment and technical debt.
- **Readiness:** Guarantees that the final validation phase will be based on an up-to-date, fully aligned architecture that reflects both technical and organizational.

The outcomes of the gap analysis will also feed into the refinement process. Depending on the nature of each finding, this may involve updating the reference architecture (Version 8 → Version 9) or adapting the relevant flows to ensure proper alignment. These updates will strengthen consistency between architecture and flows and will guide the upcoming use case - level validations to be reported in D6.

3 UC1: AI model trustworthiness evaluation for 6G distributed scenarios

The sixth generation (6G) of wireless networks is envisioned as a deeply integrated and intelligent fabric, underpinning a vast ecosystem of distributed applications and services. A foundational characteristic of this AI-native 6G architecture is its inherent decentralisation, which introduces formidable challenges in generating, managing, and validating Machine Learning/Deep Learning (ML/DL) models in a manner that is both privacy-preserving and unequivocally trustworthy. This Use Case (UC1) is designed to directly address these challenges by establishing a comprehensive framework for the end-to-end evaluation of AI model trustworthiness within distributed 6G environments.

Leveraging Decentralised Federated Learning (DFL), this use case enables collaborative model training across multiple, distinct administrative domains without necessitating the centralisation of sensitive, raw user data, thereby upholding principles of data sovereignty and privacy by design. The evaluation of trustworthiness extends beyond model accuracy to encompass a multi-faceted assessment based on crucial pillars: robustness against adversarial threats, sustainability in terms of computational efficiency, explainability for transparent decision-making, and fairness to prevent algorithmic bias.

Furthermore, this use case adopts a holistic approach by considering the trustworthiness of the entire training ecosystem. This includes the implementation of dynamic, reputation-based mechanisms to evaluate the behaviour of participating domains and nodes, as well as the integration of trustworthiness measures derived from the infrastructure layer (i.e., physical and sensing layers) to propose and validate advanced mitigation techniques against emerging security threats. With respect to the latter, we demonstrate how this evaluation can be mapped to the physical layer closed loop proposed for the security architecture and we showcase the choice of specific physical layer security schemes for authentication and key agreement based on the outcome of the infrastructure trustworthiness assessment.

The ultimate objectives of UC1 is i) to deliver a robust and verifiable framework for creating and deploying AI models that are not only high-performing but also demonstrably secure, transparent, and ethically aligned with the stringent requirements of future 6G networks; and ii) to showcase how an AI/ML driven PHY trustworthiness analysis can enable the use of physical layer security functionalities, and further, how such functionalities can be integrated in the 6G security architecture to offer low latency and low computational complexity alternatives.

3.1 Scenario 1 - Decentralized federated learning for joint privacy-preserving ML/DL model training

This scenario is focused on the architectural design, implementation, and validation of a fully decentralized federated learning framework, specifically engineered for the highly distributed and dynamic network topologies characteristic of 6G. Conventional federated learning paradigms, which rely on a central server for model aggregation, present significant limitations in the 6G context, including potential performance bottlenecks, a single point of failure, and scalability challenges. This scenario overcomes these limitations by adopting a serverless, peer-to-peer approach where network nodes directly collaborate to train a shared model.

The core innovation lies in demonstrating how ML/DL models can be collaboratively generated and refined while stringently preserving data privacy, as only model parameter updates—not raw data—are exchanged among participants. A primary goal of this scenario is the rigorous assessment of AI trustworthiness for the models produced within this framework. This involves a granular analysis of key trust pillars—accountability, fairness, explainability, robustness, and privacy—to ensure the resultant models are suitable for deployment in critical applications. Additionally, this scenario investigates the enhancement of model performance and convergence speed by integrating a reputation-based system. This system empowers participants to dynamically weigh or even discriminate against model updates from other entities based on their observed historical behaviour, thereby fostering a resilient and cooperative training environment.

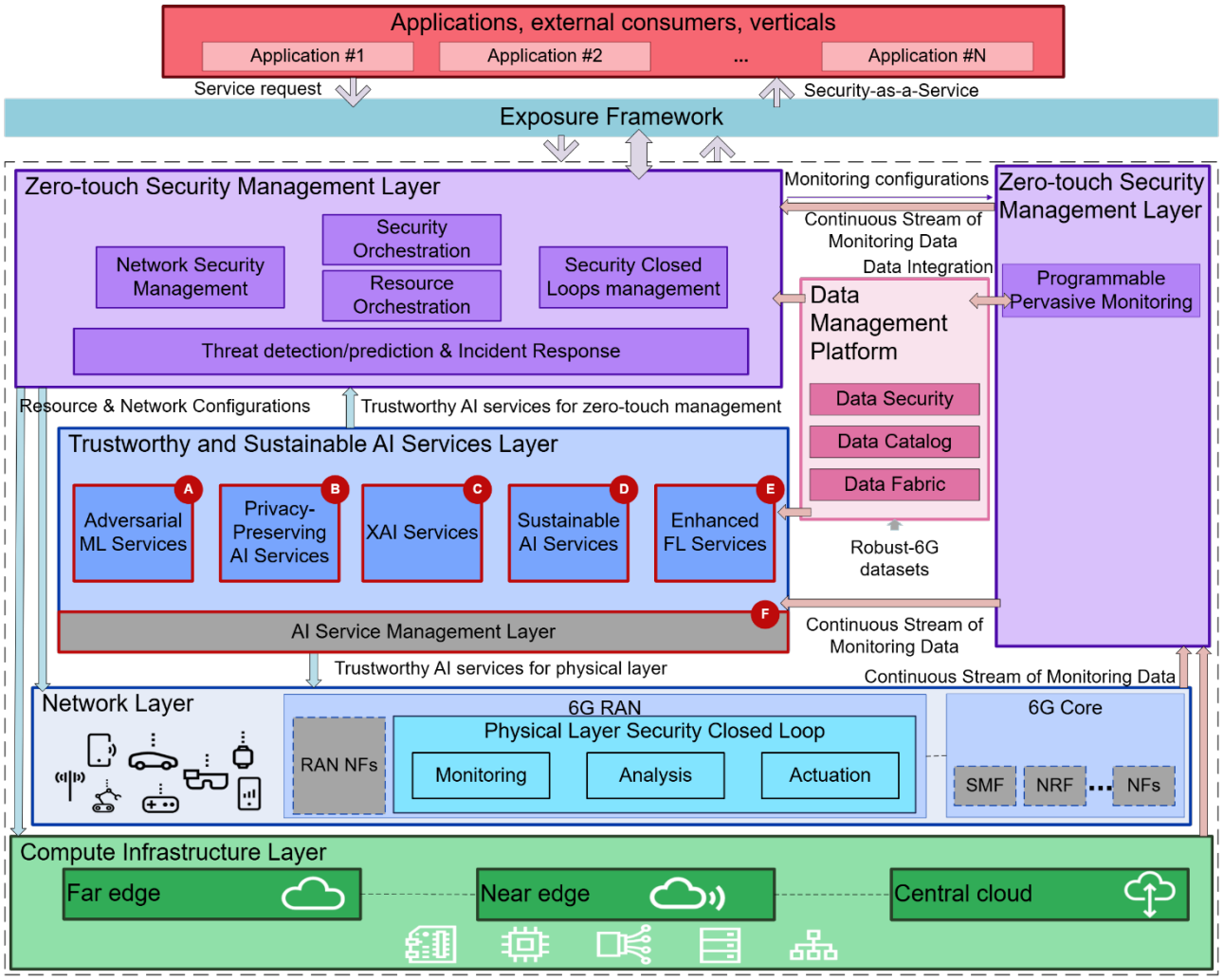

Figure 3-1 Architecture mapping of UC1_1

3.1.1 Functional Flows description and mapping

This section delineates the functional workflows that underpin the decentralized federated learning scenario. The objective is to provide a detailed account of the integration, exposition, and interaction of each trustworthiness capability within the DFL lifecycle. The flows are structured to represent a progressive maturation of the system's capabilities, commencing with a foundational baseline for privacy-preserving training and incrementally incorporating advanced layers for robustness, sustainability, and explainability. By mapping these interactions, we bridge the conceptual design with the practical implementation, enabling stakeholders to understand how security, privacy, and trust are orchestrated efficiently within the framework.

The primary components involved in these functional flows, as highlighted in the ROBUST-6G architecture shown in Figure 3-1, are as follows. Each of these flows incorporates one or more components that operate their functionalities in the architecture elements marked in Figure 3-1 with {A, B, C, D, E, F} in Trustworthy and Sustainable AI Services Layer, widely detailed in deliverable D2.2 [ROB24-D22].

1. **DFL Framework (CUMU02, CUMU03)** – {B, E} in the architecture mapping of Figure 3-1: This unified component is the core element of the scenario, integrating the orchestration engine (CUMU02) with an intrinsic Reputation & Trust Management System (CUMU03). Their responsibilities include system initialization, federation lifecycle management, enforcement of the network topology, and the execution of trust-aware distributed model aggregation algorithms.

2. **Robustness Service (CUMU04)** – {A} in the architecture mapping: This service provides a suite of tools and methodologies for evaluating and enhancing the robustness of the ML/DL models. It integrates with the DFL Framework through predefined templates and policies to apply advanced adversarial attack detection and mitigation techniques.
3. **Sustainable AI Service (CUPD01, CUPD02, CLIU01)** – {D} in the architecture mapping: These components are responsible for the assessment of sustainability aspects of the AI lifecycle. They provide functionalities for tracking, analyzing, and optimizing the energy consumption and resource footprint of the training process. The integration with the DFL Framework is currently a work in progress (WIP).
4. **XAI Services (CUMU05)** – {C} in the architecture mapping: This service offers critical explainability and trustworthiness capabilities. It interacts with the Global Model Repository to retrieve models and metrics, applying advanced techniques to provide deep model analysis, uncertainty quantification, and auditable confidence metrics for predictions.
5. **Global Model Repository (GMR) –** {F} in the architecture mapping: A logical entity responsible for the secure storage of models, performance metrics, and resource logs. It serves as a centralized point for auditing and for providing the necessary artifacts to the XAI services.
6. **Privacy-enhanced DFL services (CEBY02) –** {B, E} in the architecture mapping: This service provides a method for enhancing the privacy during the decentralized FL model training process.

Figure 3-2 illustrates the main functionalities that compose the scenario of Use Case 1, Scenario 1. It focuses on DFL for training AI models, as well as on the evaluation of the trustworthiness of these models based on three essential pillars: Robustness, Sustainability, and XAI models (explainability).

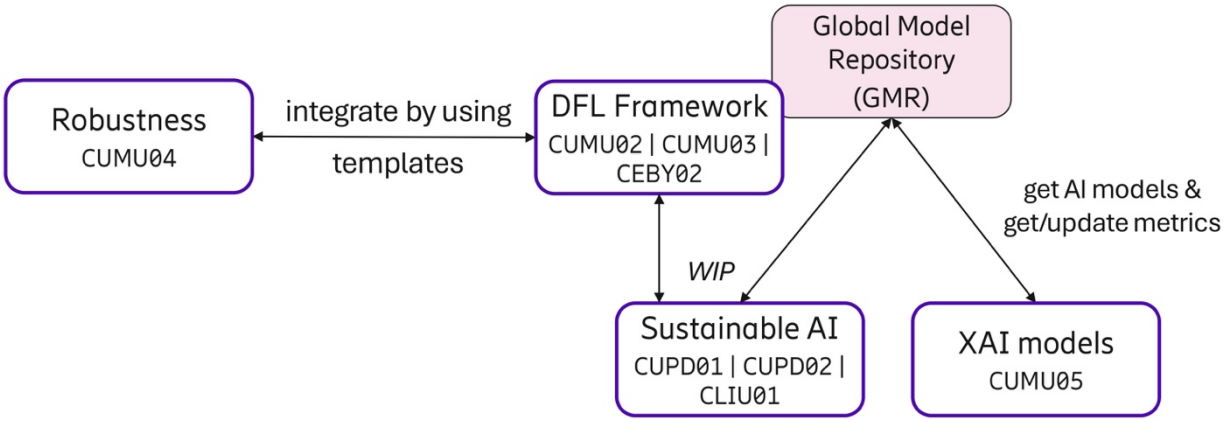

Figure 3-2 Components for the demonstration of the main functionalities of the use case

In a more descriptive way, Table 3-1outlines the trustworthiness capabilities identified for development and validation within this scenario. Capabilities highlighted in green are already under development, with partial results available. The partial results of the first two rows in green in the table were presented at the ROBUST-6G booth during the EuCNC 2025 celebration, a video demo of which is available at [ROB25-DFL]. On the other hand, rows marked in orange represent features assessed as feasible and valuable, with development planned for a subsequent phase. It should be noted that the first column ID in the table refers to the architecture mapping of components.

Table 3-1 ROBUST-6G Components implementing UC1 - Scenario 1

ID	Component Name	Description
B, E	CUMU02 CUMU03	**DFL Framework**: This is the core component of the scenario. It provides a unified orchestration engine for Decentralised Federated Learning (CUMU02) that is intrinsically coupled with a Reputation & Trust Management System (CUMU03). Its main functionalities

		include managing the federation lifecycle, defining network topologies, and executing trust-aware model aggregation. It interacts directly with the Robustness Service by applying its templates and with the Sustainable AI Service to exchange performance metrics. It also populates the Global Model Repository with trained models.
A	CUMU04	**Robustness Service**: This component is responsible for evaluating and enhancing the resilience of the AI models against adversarial attacks. It offers a set of configurable templates that define threat detection and mitigation strategies. It integrates with the DFL Framework to analyse model updates during the training process and provides feedback that influences the trust scores calculated by the reputation system.
D	CUPD01 CUPD02 CLIU01	**Sustainable AI Service**: This component focuses on the sustainability aspects of the AI lifecycle. Its role is to track, analyse, optimise, and provide insights on the energy consumption and resource footprint of the DFL process. It is designed to interact with the DFL Framework to collect real-time metrics (e.g., CPU/GPU usage, network traffic) from the Federation Nodes and to evaluate the efficiency of the final models stored in the Global Model Repository.
E	CUMU05	**XAI Services**: This component gives advanced explainability and trustworthiness analysis. Their primary interaction is with the Global Model Repository, from which they retrieve trained models and their associated metrics. They then apply sophisticated techniques (e.g., latent space analysis, Conformal Prediction) to generate comprehensive explainability reports, visualizations, and quantifiable confidence scores for the model predictions.
B, E	CEBY02	**Privacy-enhanced Service**: This component integrates privacy-enhancing technologies and security measures to safeguard distributed federated learning from data leakage and poisoning attacks. Even though it was not initially proposed, our algorithm is partially extended to safeguard decentralized FL from data leakage during the training process.
F	Global Model Repository	This logical component acts as a centralized and secure storage for all artifacts generated during the DFL process. It stores versioned ML/DL models, performance metrics, resource logs, and trust scores. It serves as the primary data source for the XAI Services and enables post-hoc auditing and analysis of the entire training history.

The following are the workflows for the demonstration of Use Case 1, Scenario 1, in accordance with the components and functionalities distribution shown in Figure 3-2:

- **Flow UC1_1_01 "Privacy and decentralization"**: This foundational flow describes the end-to-end process of decentralized model training within a trusted, benign-only network, establishing the baseline for privacy-preserving collaborative learning.
- **Flow UC1_1_02 "Evaluation of model robustness"**: This flow introduces adversarial conditions to test the framework's resilience. It details how the reputation system is used to detect and mitigate threats like model poisoning.

- **Flow UC1_1_03 "Sustainability evaluation"**: This flow focuses on assessing the energy efficiency and resource footprint of the DFL process, from the training algorithms and client scheduling to the final models themselves.
- **Flow UC1_1_04 "Explainability of the models obtained"**: This advanced flow details how to use XAI techniques to evaluate model explainability, measure prediction uncertainty, and provide a comprehensive trustworthiness assessment.
- **Flow UC1_1_05 "Enhanced privacy during DFL model training"**: This foundational flow describes the end-to-end process of privacy-enhanced decentralized model training where participants in decentralized FL are honest but curious, enabling privacy-enhanced collaborative learning.

The following paragraphs provide a comprehensive overview of each defined flow.

UC1_1_01 "Privacy and decentralization"

Objective: This foundational flow establishes the end-to-end operational workflow for collaborative ML/DL model training across distributed nodes, with a primary focus on ensuring data privacy within a trusted environment. It operates under the assumption that all participating nodes are benign and collaborative, with no adversarial actors present. The objective is to validate the core mechanics of the DFL protocol, from initial configuration to final model convergence, ensuring that local data remains private throughout the process.

Figure 3-3 provides a comprehensive overview of the workflow.

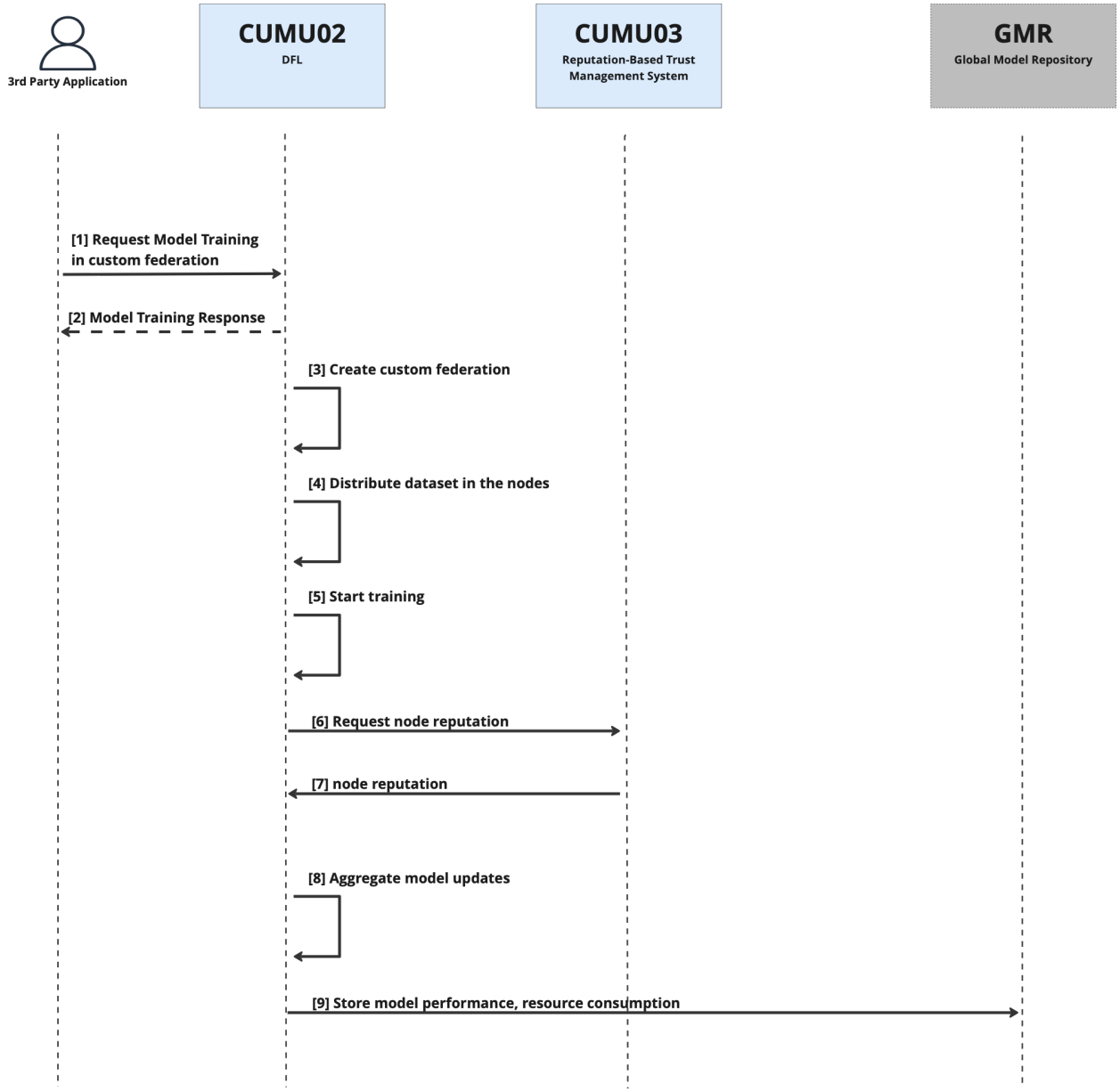

Figure 3-3 Collaborative and privacy-preserving model training flow diagram, benign nodes

Process Description: The process, as illustrated in the sequence diagram of Figure 3-3, is initiated by an external actor and managed through a coordinated interaction between the core components:

1. **Training Request [Steps 1-2]**: The process is initiated by an external *3rd Party Application*, which submits a request to the **DFL Framework** (CUMU02) to start a new model training session (Step 1). This initial request specifies the parameters for the custom federation, such as the desired model architecture, dataset, and training configuration (e.g., number of rounds, aggregation algorithm). The DFL Framework acknowledges the request and responds, confirming the initiation of the training process (Step 2).
2. **Federation Setup [Steps 3-5]:** Upon receiving the request, the DFL Framework proceeds with the internal setup. It first creates the custom federation of nodes as specified (Step 3), then orchestrates the balanced and equitable distribution of the dataset to the participating nodes (Step 4). This strict adherence to local data partitioning is the cornerstone of the framework's privacy-by-design architecture. Once the environment is prepared, the framework issues the command to start the iterative training process (Step 5).

3. **Trust-Aware Aggregation Cycle [Steps 6-8]**: This triggers the core DFL cycle which repeats for each round. Each node performs local model training on its private dataset and shares the resulting model updates with its peers. Before aggregating updates, the DFL Framework (CUMU02) queries the **Reputation-Based Trust Management System** (CUMU03) to retrieve the current reputation scores for the participating nodes (Steps 6 and 7). In this benign baseline flow, although the reputation data is retrieved, it is primarily used for logging and establishing a baseline of trustworthy behavior; no defensive filtering or weighting is applied. Following reputation check, the framework proceeds with the aggregation of model updates according to the configured algorithm (e.g., FedAvg) (Step 8).
4. **Results Storage [Step 9]**: At the conclusion of each training round, the DFL Framework stores key artifacts in the **Global Model Repository (GMR)**. This includes the updated model parameters, aggregated performance metrics (e.g., accuracy, loss), and resource consumption data (e.g., CPU usage, training time). This step ensures that a complete and auditable record of the training process is maintained for monitoring and post-hoc analysis.

This cycle of local training, reputation querying, aggregation, and storage repeats for the configured number of rounds, culminating in a converged global model built from distributed knowledge while preserving data privacy.

UC1_1_02 "Evaluation of model robustness"

Objective: This flow extends the baseline scenario by introducing adversarial conditions to rigorously evaluate the DFL system's robustness. The primary objective is to assess the framework's ability to detect and mitigate threats, specifically model poisoning attacks, by integrating the Enhanced AI/ML Model Robustness service (CUMU04) and leveraging the Reputation-Based Trust Management System (CUMU03). The threat model considers an insider attack where one or more registered federation nodes behave maliciously.

Figure 3-4 provides a comprehensive overview of the workflow.

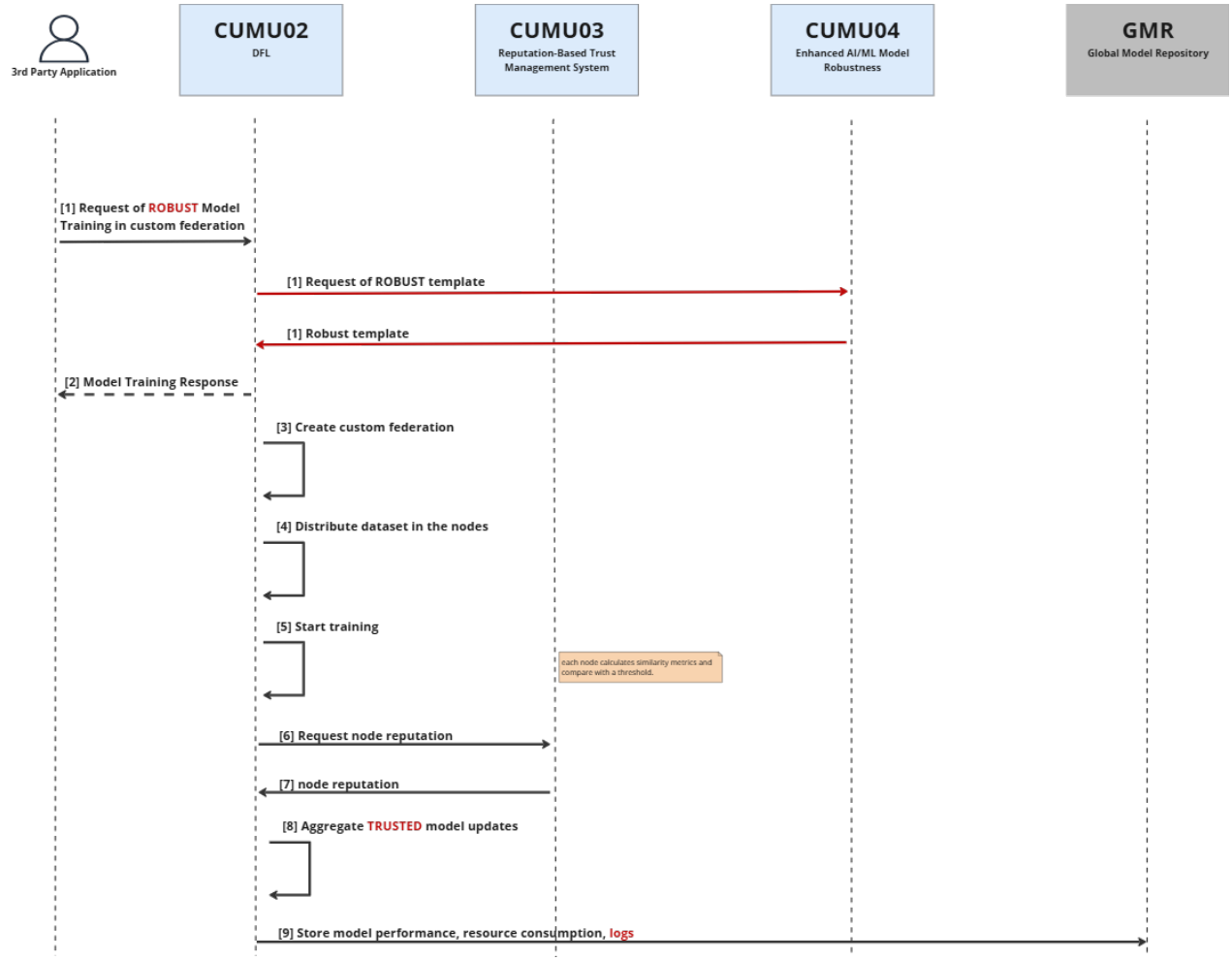

Figure 3-4 Sequence diagram for evaluating DFL system's robustness

Process Description: The process, as depicted in the sequence diagram of Figure 3-4 integrates a robustness-aware workflow from the very beginning:

1. **Robust Training Request [Step 1]**: The workflow is initiated when a *3rd Party Application* sends a request for a "ROBUST" model training session to the **DFL Framework** (CUMU02). This explicit request signals the need for enhanced security measures. In response, the DFL Framework immediately queries the **Enhanced AI/ML Model Robustness service** (CUMU04) to fetch a suitable "Robust template". This template contains a certain number of specific policies, metrics (e.g., similarity-based calculation methods), and thresholds that will be used to evaluate the trustworthiness of node contributions during the training process.
2. **Federation Setup with Adversaries [Steps 2-5]**: After receiving the template, the DFL Framework confirms the training initiation to the 3rd Party Application (Step 2). It then proceeds with the internal setup, creating a custom federation that includes both benign and malicious nodes (Step 3). The dataset is distributed (Step 4), and the training process is started (Step 5). Malicious nodes will attempt to inject poisoned updates to disrupt the model's convergence or integrity.
3. **Robustness-Enhanced Aggregation Cycle [Steps 6-8]**: This triggers the core DFL cycle, now enhanced with robustness checks. During each round, after local training, each node calculates similarity-based metrics for the updates it receives, comparing them against the thresholds defined in the Robust template. This distributed analysis allows for the early detection of anomalous updates.
    - The DFL Framework (CUMU02) then queries the **Reputation-Based Trust Management System** (CUMU03) for the current reputation of each node (Step 6).

- CUMU03 provides reputation scores (Step 7), which are influenced by the anomaly detection results derived from the CUMU04 template. Those updates from nodes that are flagged as anomalous will result in a lower reputation score.
- In the final step, the DFL Framework performs a **trust-aware aggregation** (Step 8). It only aggregates **TRUSTED** model updates, meaning it will either filter out (discard) or down-weight (reduce the influence of) updates from nodes with low reputation scores, effectively mitigating the impact of the malicious participants.

4. **Enhanced Logging for Auditing [Step 9]**: At the end of each round, the DFL Framework (CUMU02) stores comprehensive data in the **Global Model Repository (GMR)**. In this flow, this includes not only model performance and resource consumption but also detailed logs of the reputation scores and mitigation actions taken. This enhanced logging is crucial for post-hoc analysis, auditing the effectiveness of the robustness measures, and identifying persistent threats.

UC1_1_03 "Sustainability evaluation"

Objective: This flow's objective is to evaluate the sustainability and energy efficiency of the ML model training and inference processes. The evaluation focuses on three key areas: optimizing model training on edge devices, scheduling resource-constrained nodes, and assessing the energy impact of model outputs. The goal is to ensure the project's models are not only performant but also environmentally conscious by minimizing their computational and energy footprints throughout their lifecycle.

Figure 3-5 provides a comprehensive overview of the workflow.

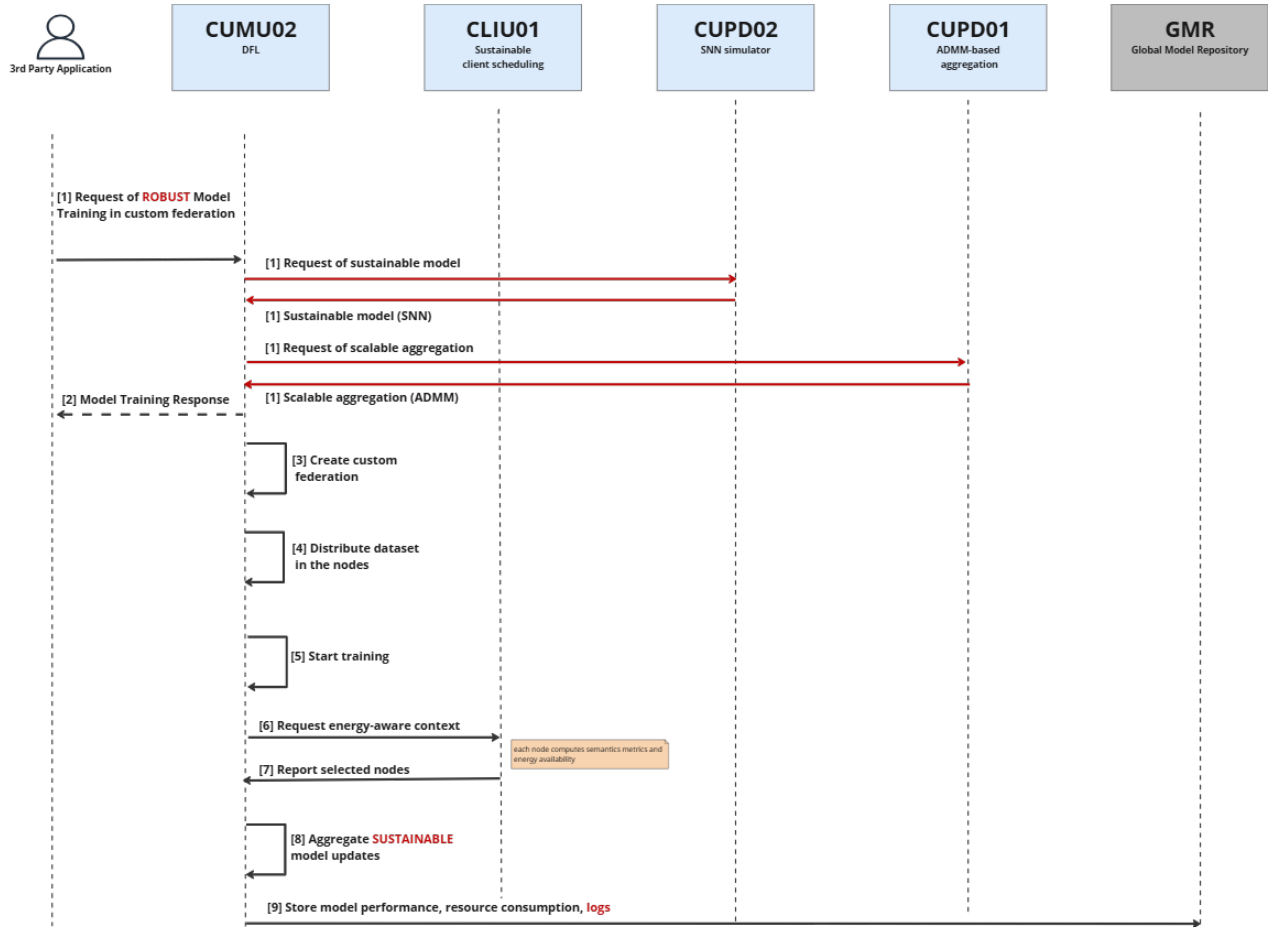

Figure 3-5 Sustainable and efficient evaluation of the model lifecycle

Process description: The sustainable flow is initiated by a third-party application submitting a request for robust model training in a custom federation. The workflow proceeds as follows:

1. **Robust training request [Steps 1-2]**: The process starts when a *3rd Party Application* sends a request for "ROBUST" model training to the **DFL Framework** (CUMU02). The framework analyses the request to determine the appropriate training method. If a sustainable model is needed, it requests one from the **SNN Simulator** (CUPD02); if scalable aggregation is required, it requests it from the **ADMM-based Aggregation System** (CUPD01). Upon receiving the sustainable model from the SNN Simulator or the scalable aggregation method from the ADMM system, the DFL Framework confirms the impending model training launch by sending a response back to the requesting application.
2. **Federation setup and data distribution [Steps 3-5]**: The DFL Framework then initiates a custom federation (Step 3), distributing the training dataset across the selected edge nodes (Step 4). Each node begins its local training using its assigned data subset (Step 5). This allows for distributed computation and privacy preservation, as raw data never leaves the device.
3. **Energy-aware client scheduling [Steps 6-7]**: In each global communication round, the server interacts with the Sustainable **Client Scheduling Scheme** (CLIU01) to determine which nodes will participate (Step 6). Each node autonomously assesses its energy availability and computes the significance of its update using semantics-aware metrics. Based on these evaluations, the scheme returns a selection of nodes that will participate in the current round (Step 7).
4. **Global aggregation and model finalization [Steps 8-9]**: The server aggregates the model updates from the selected, energy-efficient nodes (Step 8). Upon completion of the federated learning process, the DFL Framework saves key metrics including model performance, resource consumption, and detailed logs in the Global Model Repository (Step 9).

UC1_1_04 "Explainability of the models obtained"

Objective: This flow details a continuous monitoring and evaluation process for model explainability, fully integrated within the DFL training lifecycle. The primary objective is to generate specific trustworthiness artifacts for the model at each round of the federation, using exclusively Shapley values for quantitative feature attribution analysis and t-SNE for qualitative visualization of data representations. Instead of acting as a post-processing step, the XAI services are invoked during training to analyze the evolving global model. This produces a rich log of its learning behavior over time, allowing developers and auditors to analyze the entire training process retrospectively, identify at which stage a model may have developed biases or instabilities, and ultimately certify the trustworthiness of the final artifact based on its complete evolutionary history.

Figure 3-6 provides a comprehensive overview of the workflow.

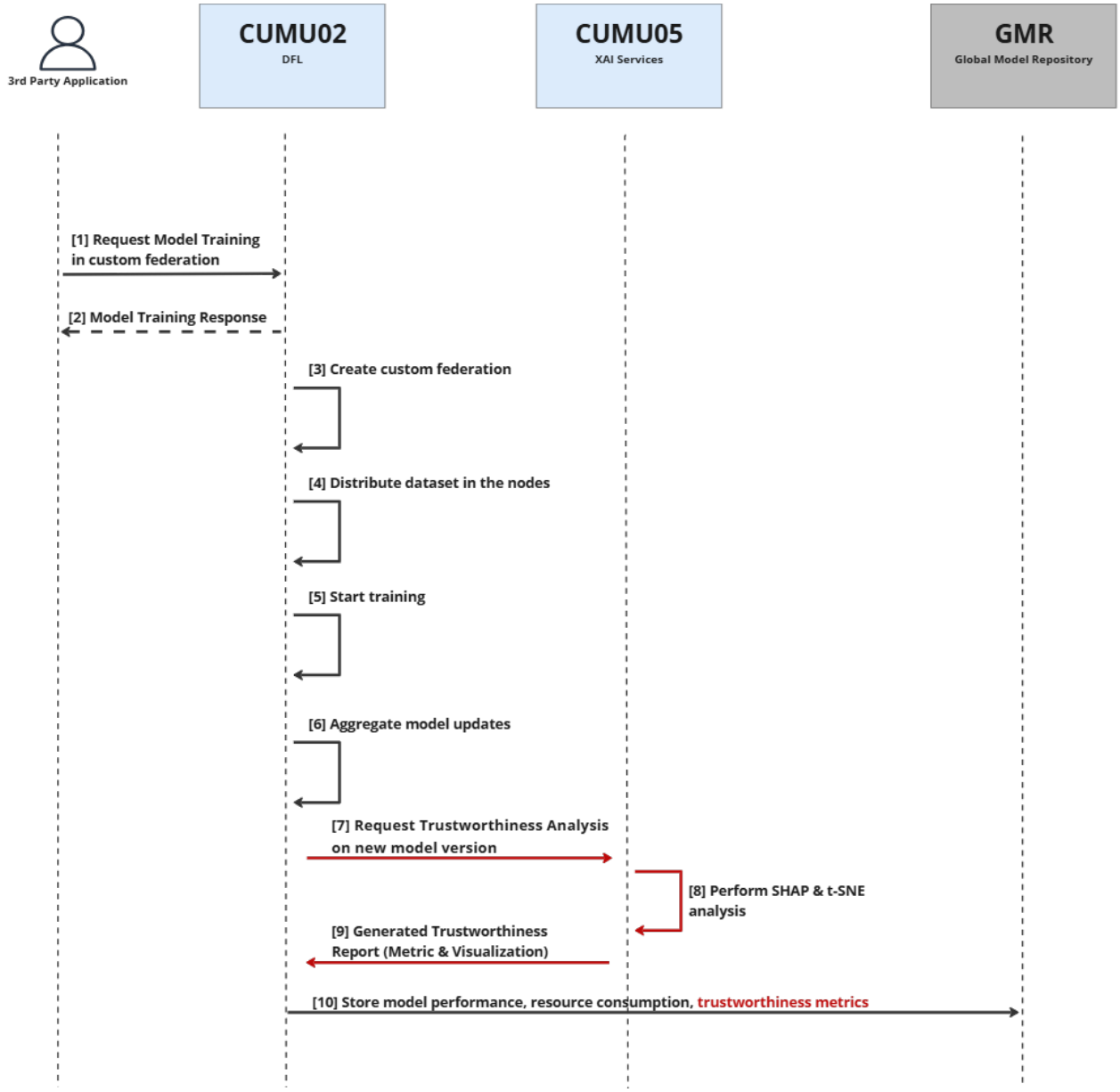

Figure 3-6 Continuous monitoring of explainability within the federated training lifecycle

Process Description: The process, as illustrated in the sequence diagram of Figure 3-6 integrates XAI analysis as a systematic monitoring step within each federated training round.

1. **Federation Initiation and Model Training [Steps 1-5]**: The flow assumes that a standard DFL training process has been initiated, as described in **UC1_1_01**, and the federated models are being trained. The **DFL Framework** (**CUMU02**) has created the federation (Steps 1-3), distributed the data (Step 4), and started the iterative training loop (Step 5).
2. **Model Aggregation [Step 6]**: At the end of each federated round, the DFL Framework performs the standard aggregation of model updates received from all participating nodes. This creates a new, updated version of the global model for that specific round.
3. **On-the-fly Trustworthiness Analysis [Steps 7-9]**: Immediately after a new global model version is created, the DFL Framework sends it to the **XAI Services** (CUMU05) for a trustworthiness assessment (Step 7).
    - XAI Services (CUMU05) perform a comprehensive analysis on this specific model version (Step 8), generating one quantitative metric and one qualitative visual artifact:

- **Metric 1 – Feature Attribution Stability (from SHAP)**: This metric provides a quantitative measure of the model's reasoning stability. The XAI Service calculates the mean absolute **Shapley values** for each feature over a reference dataset, producing a feature importance vector for the current model version. The **cosine similarity** is then computed between this vector and the one generated for the model from the previous round. A high similarity score (close to 1.0) indicates that the model is learning in a stable, predictable manner. A low or fluctuating score can signal training instability, indicating that the model's internal logic is changing drastically between rounds, which reduces its trustworthiness.
- **Artifact 2 – Data Representation Visualization (from t-SNE)**: This artifact provides a powerful qualitative tool for human-in-the-loop analysis. The XAI Service uses the current model version to generate latent space embeddings for a validation dataset and then creates a **2D t-SNE visualization** of these embeddings. This plot allows auditors to visually inspect whether the model is learning to form distinct and meaningful clusters for different data classes over time. The progressive separation of clusters from round to round is a strong visual indicator of healthy and effective training. The output is the plot image itself, which is stored as a visual artifact.

  - The XAI Services then compile the quantitative stability metric and the t-SNE visualization into a structured **Trustworthiness Report** and return it to the DFL Framework (Step 9).

4. **Comprehensive Logging in GMR [Step 10]**: The DFL Framework takes the new global model version, its standard performance metrics (accuracy, loss), and the newly generated Trustworthiness Report from the XAI services and stores them all together in the **Global Model Repository (GMR)**. This includes the calculated Feature Attribution Stability score and the generated t-SNE plot for that round.

This cycle repeats for every round of the federation. The outcome is a complete, versioned history of the training process stored in the GMR. Each model version is enriched with a corresponding report containing its feature attribution stability score and its data representation visualization. This allows for an unprecedented level of transparency, enabling auditors to not only inspect the final model but also to understand and visually verify its entire learning trajectory.

UC1_1_05 "Privacy-enhanced DFL"

Objective: This flow's primary focus is to enhance privacy in model training across distributed nodes where nodes are trusted but curious. Meaning that all nodes follow the rules of the protocol correctly but will try to learn as much as possible from the model updates that are exchanged among nodes during model training. No adversarial attacks are considered in this flow.

Figure 3-7 provides a comprehensive overview of the workflow.

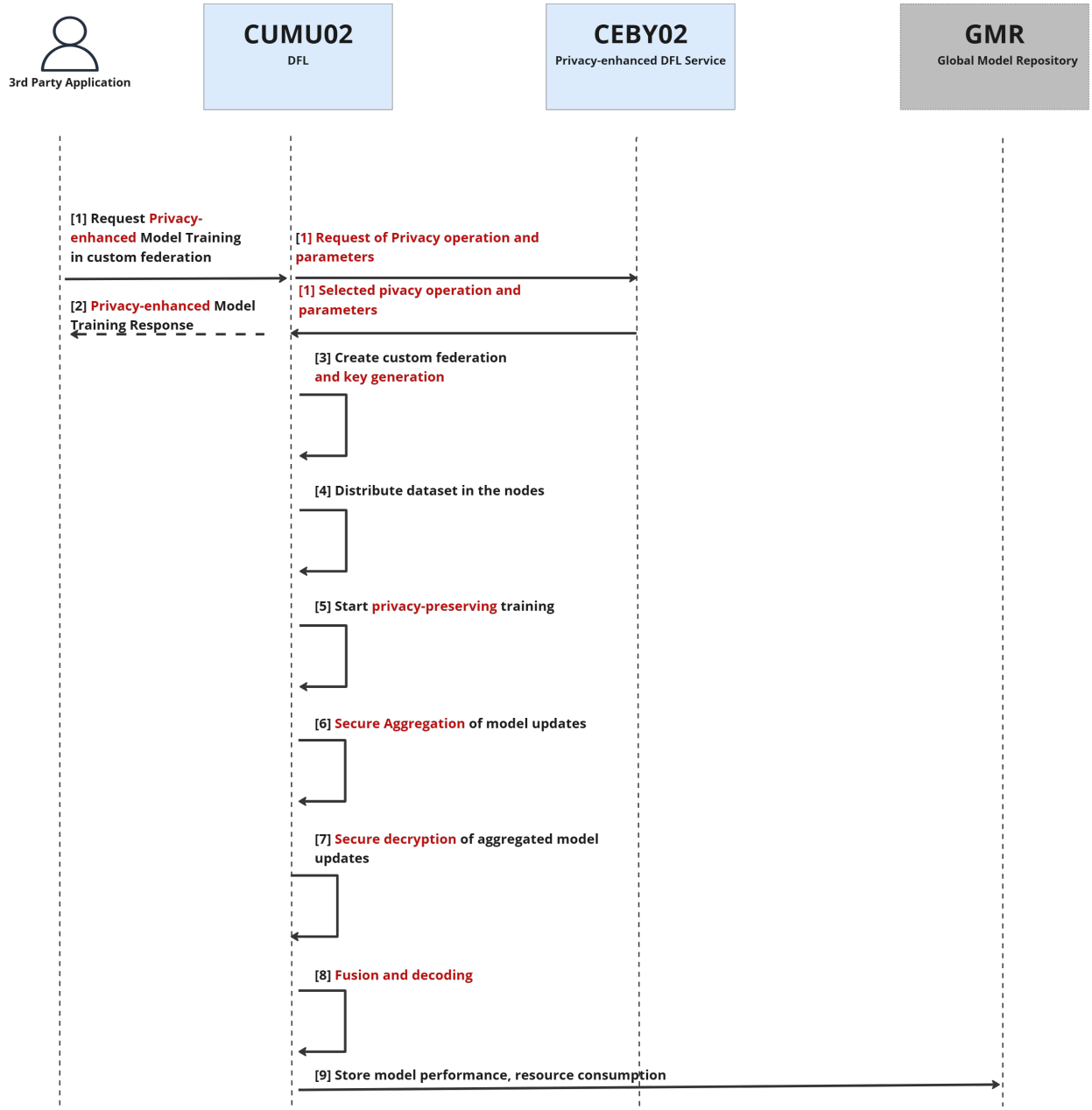

Figure 3-7 Privacy-enhanced collaborative model training flow diagram

Process Description: The process, as illustrated in the sequence diagram of Figure 3-7, is initiated by an external actor and managed through a coordinated interaction between the core components:

1. **Training Request [Steps 1-2]**: The process is initiated by an external *3rd Party Application*, which submits a request to the **DFL Framework** (CUMU02) to start a new privacy-enhanced model training session This initial request specifies the parameters for the custom federation, such as the desired model architecture, dataset, and training configuration (e.g., number of rounds, aggregation algorithm). Also, requesting the suitable privacy operation and the required parameters (e.g. homomorphic encryption) for the privacy operation through the CEBY02 (Step 1). DFL Framework acknowledges the request and responds, confirming the initiation of the training process (Step 2).
2. **Federation Setup and Key Generations [Steps 3-4]:** Upon receiving the request, the DFL Framework proceeds with the internal setup. It first creates the custom federation of nodes considering their privacy operation capability, and then each participating node by getting specified parameters for the privacy operation, generates its own secret and private keys. All nodes engage in

a protocol to compute a joint public key (step 3). Finally, the DFL framework orchestrates the balanced and equitable distribution of the dataset to the participating nodes (Step 4).

3. **Privacy-preserving Model Training [Steps 5]**: Once the environment is prepared, the framework issues the command to start the iterative training process (Step 5). Each node trains a local model using only its private dataset. Each node encrypts its model updates using joint public key. Nodes share their encrypted model updates with neighbouring nodes based on the network topology.
4. **Collaborative Secure Aggregation [Steps 6]:** Each node aggregates received encrypted model updates from neighbours using secure aggregation method. This produces an encrypted sum of all client model updates.
5. **Collaborative Decryption [Step 7]:** Each node computes a partial decryption share using its own secret key. This partial decryption does not leak any information related to the node's model update because the result still has other nodes' contributions mixed in. Each node sends its share to all other nodes.
6. **Fusion and decoding [Step 8]:** Each node combine all received shares by running multi-party decryption fusion function on all shares. So, all nodes obtain the plaintext result. The plaintext is decoded to obtain the approximate aggregated sum.
7. **Results Storage [Step 9]**: At the conclusion of each training round, the DFL Framework stores key artifacts in the **Global Model Repository (GMR)**. This includes the updated model parameters, aggregated performance metrics (e.g., accuracy, loss), and resource consumption data (e.g., CPU usage, training time).

This cycle of local training, reputation querying, aggregation, and storage repeats for the configured number of rounds, culminating in a converged global model built from distributed knowledge while preserving data privacy.

3.1.2 Testbed Requirements and Deployment

The experimental validation of this scenario will be exclusively conducted in the UMU testbed (TUMU01), which provides a versatile and controllable environment for simulating distributed 6G network scenarios. The testbed consists of a cluster of virtualized servers with software-defined networking (SDN) capabilities, allowing for the flexible instantiation and interconnection of the required software components.

Deployment Architecture

The components of this Use Case 1-Scenario 1 will be deployed using a container-based architecture to ensure portability, scalability, and ease of management. The DFL Framework (CUMU02) orchestrator, the Reputation & Trust Management System (CUMU03), and the Global Model Repository will be deployed as services within a Kubernetes cluster running on TUMU01. The Federation Nodes will be instantiated as individual Docker containers, with their network connectivity managed by the SDN controller to dynamically create the desired topologies (e.g., ring, mesh). This setup allows for the precise control of network conditions, such as latency and bandwidth, between nodes, enabling the simulation of realistic wide area 6G deployments.

Current Status and Future Work

The foundational implementation of the DFL Framework (CUMU02) has been finalized and validated against its initial criteria as defined in D6.1 (Table 11). This validation confirmed the framework's capabilities in terms of Model Performance, Scalability, and Data Distribution.

The baseline flow (UC1_1_01) has been deployed and tested within the UMU testbed using two simulated datasets. Preliminary tests of this flow have yielded promising results against the initial KPIs outlined in D6.1. For instance, in a controlled experiment using the CIFAR-10 dataset, the framework achieved a model accuracy improvement of over 6% compared to standalone training, surpassing the initial target of 5% (KPI 2). This successful execution of the baseline flow establishes the necessary foundation for evaluating the more advanced trustworthiness and robustness metrics.

The current development focus is on the implementation of the Reputation & Trust Management System (CUMU03). The next milestones involve the development of secure APIs for integrating CUMU03 with CUMU02, which will enable the execution of the reputation-based filtering flow (UC1_1_02) and the formal validation of KPI 3 (Adversarial Robustness). Subsequent work will focus on integrating the XAI and Sustainability services to enable the execution of flows UC1_1_03 and UC1_1_04, which are essential for the formal validation of KPI 1 (Trustworthiness Score) and KPI 4 (Sustainability Score).

3.1.3 KPIs and Validation Criteria

The validation activities will be conducted in a phased, iterative manner, directly aligning with the progressive implementation of the functional flows. The baseline for KPI 2 has already been established with the validation of flow UC1_1_01. Quantitative results for the remaining KPIs (KPI 1, KPI 3, and KPI 4) will be systematically gathered as their enabling flows (UC1_1_02 to UC1_1_04) are finalized and tested. Consolidated findings will be formally reported in subsequent project deliverables.

KPI 1: Trustworthiness Score

- **Target**: The final collaboratively trained AI/ML models shall achieve a composite trustworthiness score of ≥ 80%.
- **Validation Methodology**: This composite score will be calculated as a weighted average of metrics from several trust pillars. Each pillar will be assessed independently:
  - **Robustness**: Evaluated by simulating a range of adversarial attacks and measuring the model's performance degradation.
  - **Explainability**: Assessed quantitatively using metrics such as Fidelity and Comprehensibility on the outputs of integrated XAI services.
  - **Fairness**: Measured using standard fairness metrics like demographic parity and equalized odds on a dataset with known sensitive attributes.
  - **Sustainability**: Quantified by measuring the average CPU, memory, and network bandwidth consumed per node during the training process.

KPI 2: Model Accuracy Improvement

- **Target**: The federated model must demonstrate an average accuracy improvement of ≥ 5% over standalone models.
- **Validation Methodology**: A baseline will be established by training a separate, standalone ML model on the isolated dataset of each participating node. The accuracy of the final federated model, trained for the same number of epochs, will be compared against the average accuracy of these standalone models. The test will be conducted on a held-out, global test dataset to ensure an unbiased evaluation.

KPI 3: Adversarial Robustness

- **Target**: The AI/ML models must achieve a minimum robustness score of ≥ 85% against a defined set of adversarial attacks.
- **Validation Methodology**: The system's resilience will be tested by injecting malicious nodes into the federation that perform common attacks, such as data poisoning (e.g., label-flipping) and model poisoning. The robustness score will be measured using metrics like the Attack Success Rate (ASR), which quantifies the adversary's ability to degrade model performance or cause targeted misclassifications. The evaluation will be conducted both with and without the reputation and XAI-based mitigation mechanisms actively quantify their effectiveness.

KPI 4: Sustainability score

- **Target 1**: The trained model must run with inference power reduced by 30% if standard NN, or three orders of magnitude less than standard NN if SNN.
- **Target 2**: The training process should lower energy consumption by 30% concerning not considering energy optimization or increasing the model accuracy (+5%) when dealing with power constraints.

- **Validation methodology**: Models and training procedures will be validated by comparing them with not optimized ones for energy consumption and sustainability (e.g., non-quantized models, vanilla FedAvg, etc.).

The validation activities will be conducted iteratively, aligning with the progressive implementation of the functional flows. The results and detailed analysis of these KPIs will be formally reported in subsequent project deliverables.

3.1.4 Flow Progress Tracking

Table 3-2 reports the intermediate status of four flows addressing decentralized federated learning in trusted 6G environments. All flows are defined and mapped to the relevant components, with integration work in progress. Some components (e.g., CUMU02, CUMU04) have been partially integrated, while others remain under preparation. The main challenges identified relate to inter-component integration and the need for alignment between local and federated testbeds. These results demonstrate steady progress at the flow level, while also highlighting dependencies that must be resolved before full validation and KPI measurement in the next phase. The different integrated components illustrated in Table 3-2 will be deployed in PTA TUMU01, although they can also be implemented in local environments with virtualized capabilities.

Table 3-2:Use Case 1 Scenario 1 Flow Progress Tracking Table

ID	Included Components		Integration %	Status
UC1.1_1	3rd Party	CUMU02	10%	First integrations performed using a Frontend (dashboard)
	CUMU02	CUMU03	0	Components work individually, but they are not integrated yet
	CUMU02	GMR	0	Analysis of different tools for implementing this component
UC1.1_2	3rd Party	CUMU02	10%	First integrations performed using a Frontend (dashboard)
	CUMU02	CUMU03	0	Components work individually, but they are not integrated yet
	CUMU02	CUMU04	25%	First data poising attack integrated in the DFL Framework (CUMU02)
	CUMU02	GMR	0	Analysis of different tools for implementing this component
UC1.1_3	CUPD01	CUMU02	0	Under discussion for integration
	CUPD02	GMR	0	Analysis of different tools for implementing this component
	CLIU01	GMR, CUMU02	0	First meetings held, with and integration process underway
UC1.1_4	3rd Party	CUMU02	10%	First integrations performed using a Frontend (dashboard)
	CUMU02	CUMU05	0	Components work individually, but they are not integrated yet

	CUMU02	GMR	0	Analysis of different tools for implementing this component
UC1.1_5	3rd Party	CUMU02	10%	First integrations performed using a Frontend (dashboard)
	CUMU02	CEBY02	0	Under discussion for integration
	CUMU02	GMR	0	Analysis of different tools for implementing this component

3.2 Scenario 2 - Physical and sensing layer trustworthiness and resilience

Use Case 1 – Scenario 2 (UC1.2) aims on enhancing the trustworthiness of the physical and sensing layers in 6G networks, focusing on the integration of Physical Layer Security (PLS) and trust mechanisms to ensure integrity, privacy, and resilience of 6G environments. In this framework, as already outlined in deliverable D6.1 [ROB25-D61], this approach involves mechanisms such as angle of arrival based physical layer authentication (AoA-PLA), challenge response physical layer authentication (CR_PLA), RF fingerprinting, secret key generation (SKG), keyless confidential communication and PHY monitoring, targeting on establishing comprehensive framework that secures the physical and sensing layers of 6G networks while supporting high performance and scalability. Furthermore, as outlined in D6.1, the current UC scenario involves several challenges that need to be addressed. In this regard, to ensure RF fingerprinting accuracy and robustness adaptive ML models were introduced, e.g, a novel adaptive robust principal component analysis (A-RPCA) that also reduces the error probability after reconciliation. Additionally, to enhance resilience to physical-layer attacks improved ML-based techniques for jamming/spoofing identification are developed. Also, particular focus was put to ensure the requirements of real-time operation and scalability, e.g., in the implementation of the fast SKG for privacy amplification. Finally, the incorporation of adaptive algorithms to provide adjustments to dynamic environmental conditions remains a task for future work. In this context, UC1.2 envisions a smart factory setting where 6G communication enables connectivity between devices and access points, as well as direct device-to-device (M2M) communication, all under the demand for fast and highly reliable connections. As described in the next subsections, UC1.2 mainly exploits the functionalities of the physical layer closed loop of the ROBUST-6G architecture, focusing on the validation of three critical requirements: i) trustworthiness evaluation of the PHY layer, ii) mutual authentication of the devices and the access points as well as device pairing, and iii) secret key generation. In the rest of this subsection, a short description regarding the status of the functional flows and setup specifications of the UC1.2 is provided.

3.2.1 Functional Flows description and mapping

We first introduce the three functional flows of the UC1.2:

- **Flow UC1_2_01 “PHY layer trustworthiness evaluation”**: The first flow focuses on PHY trustworthiness evaluation by comprehensive monitoring of physical layer parameters, traffic analysis, jamming attack detection, and secrecy map generation.
- **Flow UC1_2_02 “Mutual authentication”**: This flow handles mutual authentication of a device (robot, drone, sensor, etc.) and an access point and secure device pairing in M2M setups (e.g., robot paired with another robot).
- **Flow UC1_2_03 "(Fast) Secret key generation”:** SKG meeting the proposal’s KPIs: more than 99% reconciliation rate and less than 5 msec run time for the overall AKA time (work in progress). These KPIs have been met in experiments with real datasets reported in D5.1 [ROB25-D51], and will be detailed further in D5.2 (to be released on December 2025), as well as a demonstrator of a real time SKG that was presented in the 6G Summit in Dresden in May 2025 and in the IEEE Conference on Standards for Communications and Networking [CSCN-2025].

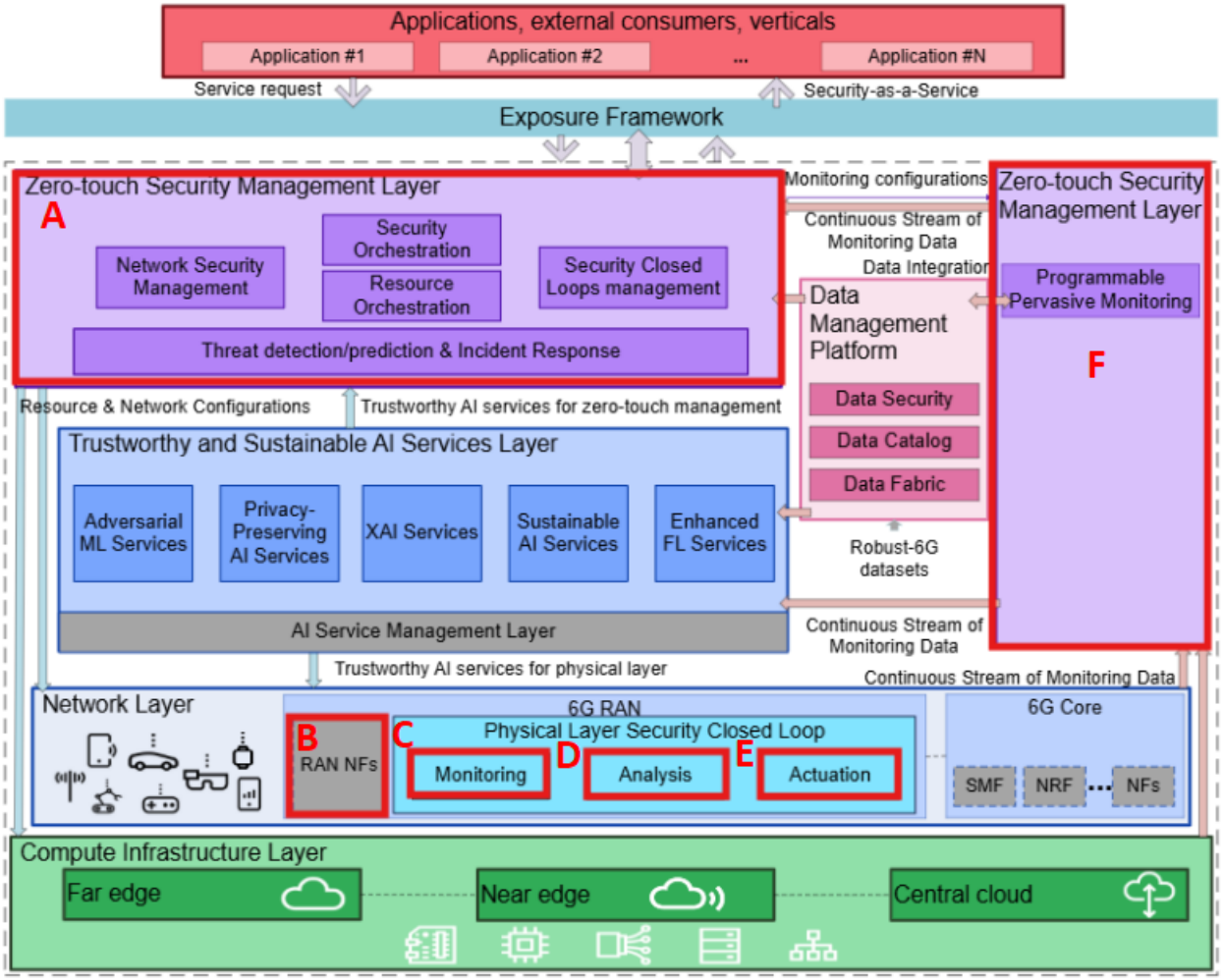

Figure 3-8 Architecture mapping of UC1_2. Red squares indicate the involved functionalities

In Figure 3-8, the mapping of the functionalities being validated within the functional flows of UC1_2, aligned with the overall ROBUST-6G architecture (highlighted with red squares) is illustrated, while Table 3-3 provides a brief description of the components involved in the architecture's functionalities.

Table 3-3 ROBUST-6G Involved Components implementing UC1 - Scenario 2

ID	Component Name	Description
A	CNXW01	Zero-touch Security Orchestrator (CNXW01) is the responsible component for requesting and managing security services, composed of a Security Service Orchestrator module and a Closed-Loop (CL) Governance/Coordination module. The Security Orchestrator oversees translating the consumer requests (e.g.: SSLA) into commands to the specific orchestrators (e.g.: Resource Orchestrator, Network Orchestrator). The Closed-Loop component oversees handling and coordinating the respective functionalities (monitoring, analysis, decision, and action) in the appropriate environment. In UC1_2 scenario, CNXW01 is utilized to provide the necessary Upper layer context inputs to the PHY layer closed loop, e.g., security configuration parameters and contextual data such as GNSS-based localization information and orchestration alerts.

B	PHY Layer	In UC1_2, PHY Layer is used to provide the appropriate physical layer inputs to the PHY layer closed loop, such as CSI and radar-based sensing metrics, representing RF signals and sensing observations from the radio environment. Additionally, PHY Layer is also fed from the PHY layer closed loop to update the RAN specifications (RAN NFs).
C	CENS01, CCHA02	Monitoring stage receives the PHY and Upper layer inputs and exploits the following components: i) PHY monitoring (CENS01), utilized to estimate core channel metrics, e.g., SNR and determination of LoS/NLoS conditions, and, ii) Data sets generation and fingerprinting for Physical Layer Security (CCHA02), employed to generate RF datasets for fingerprinting-based research.
D	CUPD03 CENS02 CEBY04 CGHM02 CENS03 CUPD05	Analysis stage primary focuses on i) PHY attack identification, and, ii) overall trustworthiness evaluation of the physical layer. In this perspective, it exploits the below components: Secrecy and Information Leakage (CENS02), Jamming Detection (CUPD03), Signal/Attack Identification solution to Classify different types of EM Signals (CEBY04), RF-Predict (CGHM02), Trustworthy Sensing and Localization (CENS03) and Cross-layer Holistic Anomaly Detection System (CUPD05).
E	Activation CCHA01 CENS04 CEBY05 CUPD04 CENS05 RAN control CGHM01	Actuation stage implements decisions based on the outputs of the first two stages to support PHY resource control and provisioning. Based on the outputs of the PLS activation (Activation component), these decisions include: i) the utilization of the appropriate PLS scheme, i.e., enable/disable security features depending on the required trustworthiness level; employed components include the PLS in NOMA MIMO Systems (CCHA01), AoA-based PLA (CENS04), Identification/Authentication of Legitimate Devices (CEBY05), PHY-layer based enhanced AKA Protocols (CUPD04) and Fast SKG using LSTM networks for Privacy Amplification (CENS05), and, ii) the activation of PHY control operations to adjust parameters regarding power/resource allocation, modulation schemes, synchronization sequences etc; involved components include RAN Control and RF Fingerprinting Migration (CGHM01). This stage closes the PHY layer closed loop by feeding back to the RAN and the orchestrator.
F	CUMU01	The Programmable Monitoring Platform (PMP) is an automated solution for managing service and resource health, possible anomaly detection by a rule-based IDS, aggregating data from multiple sources, and enabling dynamic configuration through virtualization techniques. It enables efficient monitoring and closed-loop management of service performance. In the context of UC1_2 scenario, Actuation stage interfaces with the orchestrator through the PMP, raising potential alerts to the upper layers.

Finally, in Figure 3-9, we aim on a high-level presentation of the interconnection among the utilized components for each individual functional flow, within the physical layer closed loop and the interaction of the physical layer closed loop to the zero-touch security management and the network layer. Note that, orange coloured components refer to Flow 1, light blue coloured components to Flow 2, and, red coloured components to Flow 3.

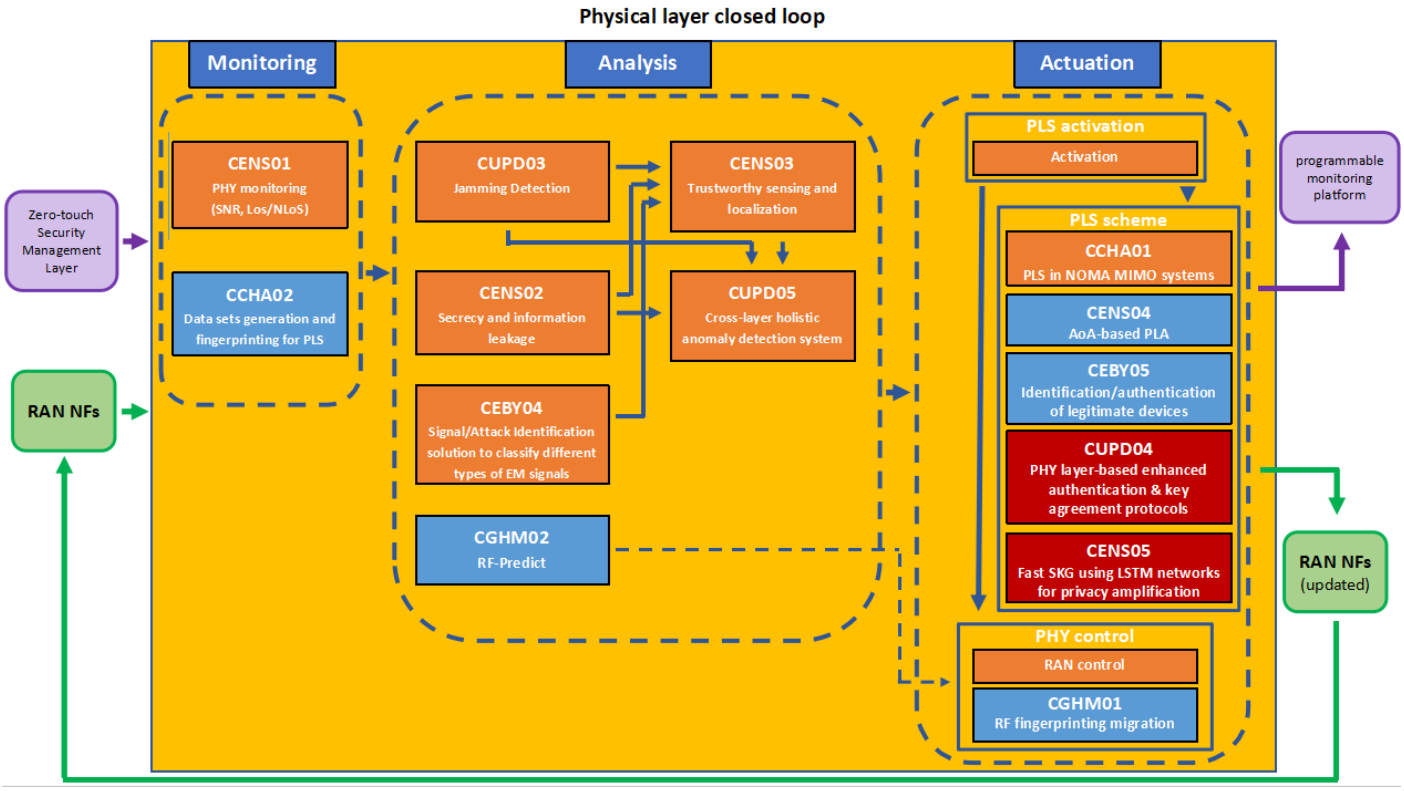

Figure 3-9: High-level description of the functional flows and their involved components within the physical layer closed loop

Flow UC_1_2_01 "PHY layer trustworthiness evaluation"

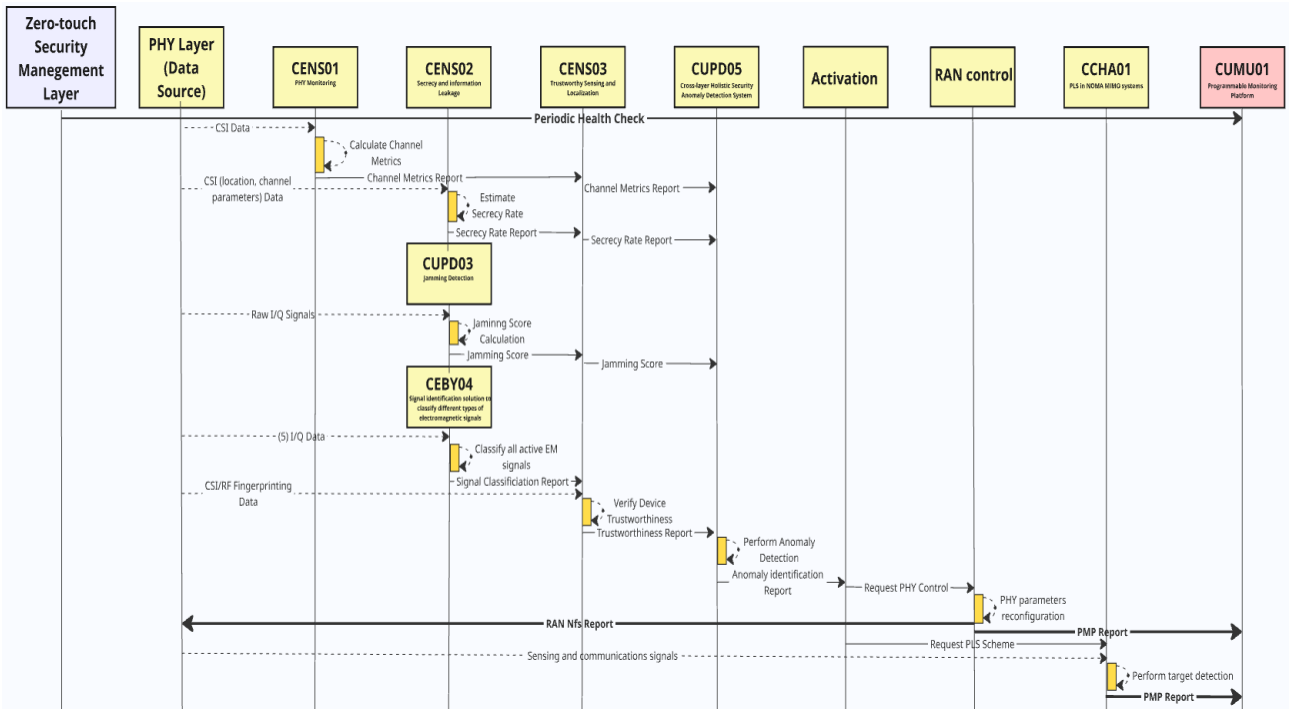

Figure 3-10:PHY layer trustworthiness evaluation flow diagram

Description: The flow in Figure 3-10 spans the monitoring, analysis, and actuation phases of the closed loop. On the reception of raw physical measurements from the PHY layer (CSI, Radar) and upper layer inputs from the zero-touch security management layer (e.g., configuration parameters, orchestration alerts etc), the process begins with CENS01 that monitors (the frequency of monitoring is to be determined) important physical layer parameters like Signal-to-Noise Ratio (SNR) and Line-of-Sight / Non-Line-of-Sight (LoS / NLoS) conditions to establish baseline trustworthiness indicators. Next, upon analysis: i) CUPD03 identifies likely RF jamming attacks and interference patterns, ii) CENS02 generates secrecy maps through analysing probabilistic estimates of information leakage and privacy at the physical layer, also focusing on preventing potential eavesdroppers from intercepting confidential information, and, iii) CEBY04 analyses RF signatures and patterns to differentiate different classes of electromagnetic signals and identify possible attacks. These components feed CENS03 to provide trusted positioning and sensing data. All the previous components feed CUPD05 to integrate trust information from several layers to provide holistic anomaly detection, correlating physical layer indicators with higher-layer security events. The flow proceeds to the actuation stage through the activation in PLS activation, integration of the NOMA MIMO scheme for PLS by CCHA01, and RAN control leverages RF forecasting insights to apply real-time reconfiguration of PHY parameters. The resultant actuation decisions are fed to the radio access network functions (RAN NFs) through the PHY Layer component, which execute the necessary control actions and close the loop by directly translating them to new PHY layer inputs. Furthermore, the output of this flow provides alerts and trustworthiness measures to the orchestrator by communicating with the programmable monitoring platform (CUMU01). In brief, the flow provides the complete sensing, analysis, and actuation decisions that ensure physical layer trustworthiness throughout the entire closed loop, interacting with both lower and upper layers.

Flow UC1_2_02 "Mutual authentication"

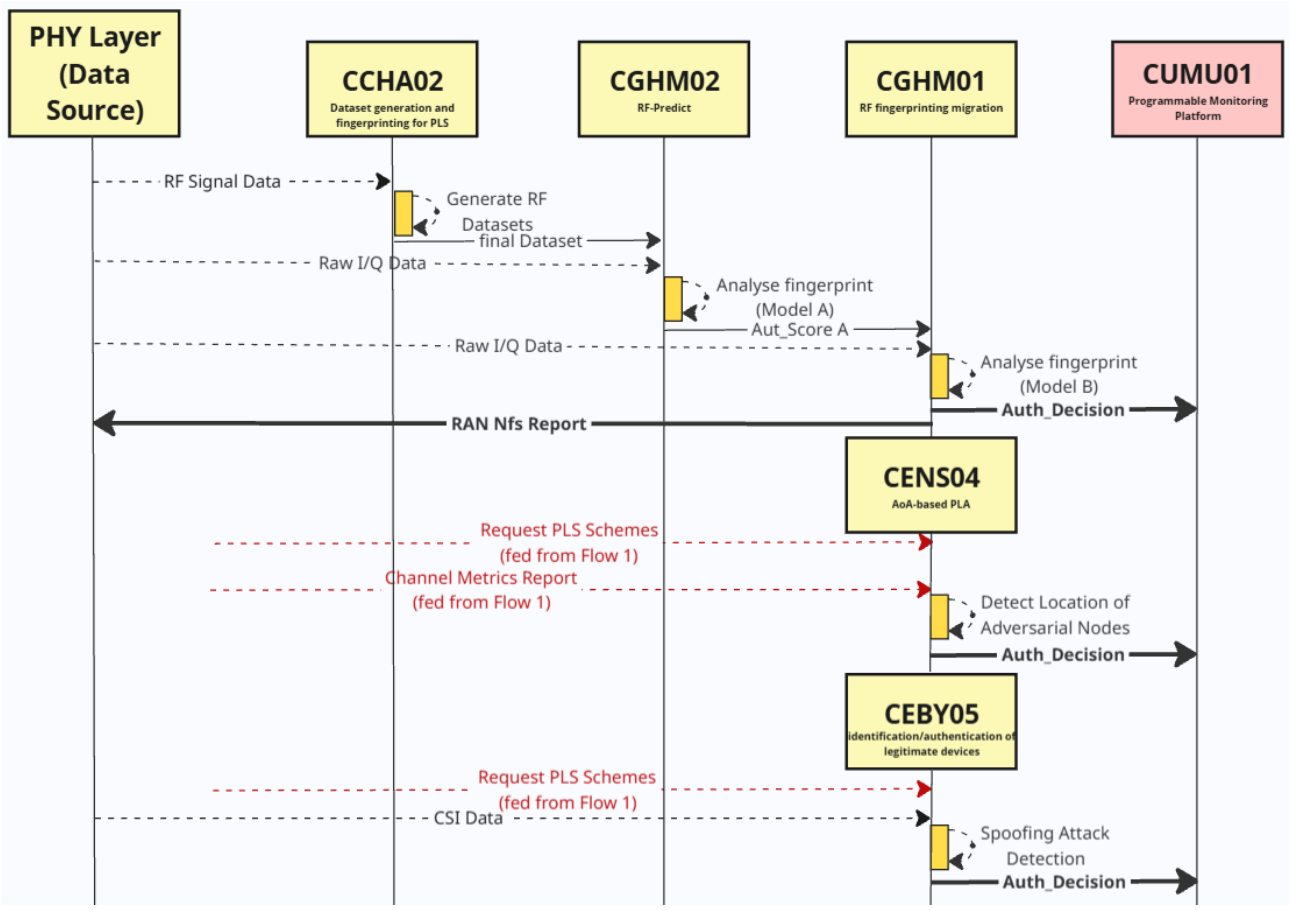

Figure 3-11:Mutual authentication flow diagram

Description: As illustrated in Figure 3-11, flow UC1_2_02 encompasses all stages of the closed loop, i.e., monitoring, analysis, and actuation. In the monitoring stage, CCHA02 uses a RF digital twin to generate comprehensive datasets and create RF fingerprints that can be used for PLS mechanisms. In the analysis phase, CGHM02 provides predictive analytics for RF behaviour (evolution of fingerprints) and potential security threats based on historical trends and real-time conditions, enhancing authentication decisions through the prediction of potential impersonation or degradation threats. Actuation phase involves CENS04, which uses spatial features and Angle-of-Arrival (AoA) measurements to authenticate legitimate devices and detect impersonation attacks. Another option for authentication is to employ CEBY05 which authenticates device legitimacy and accommodates both access point-device authentication and M2M device pairing use cases. Further, CGHM01 manages RF fingerprints migration between base stations to offer continuity of authentication while roaming devices across the network. Depending on the scenario, the configuration parameters, etc., the choice of running all or a subset of the authentication components is made by Flow 1 (PLS Activation), that in this sense feeds Flow 2.

Flow UC1_2_03 "(Fast) Secret key agreement"

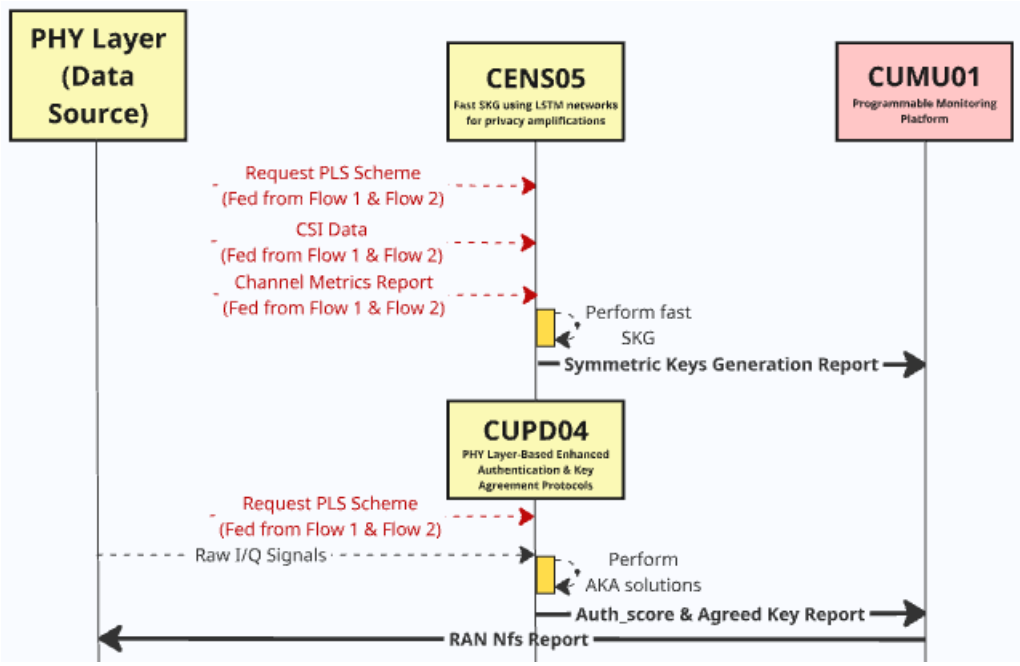

Figure 3-12:(Fast) Secret key agreement flow diagram

Description: As shown in Figure 3-12, the activation of components for fast SKG depends on the output of the PLS activation component, specifically the results from both Flow 1 and Flow 2. This ensures that the action occurs only between mutually authenticated devices. In the future we can choose to allow over-riding this step (if for example upper layer authentication schemes are used instead). In the actuation phase, CUPD04 utilizes advanced PHY-layer-based key agreement and authentication protocols. CENS05 delivers fast secret key generation based on LSTM networks (other possibilities using convolutional neural networks will be investigated to capture spatiotemporal information when available), meeting key performance targets, namely >99% for reconciliation rate and runtime less than 5 msec for the overall AKA time [CSCN-2025] in which a run time of 0.2 msec was reported for the privacy amplification. Our approach for fast privacy amplification using LSTM networks was incorporated in a real-time live demonstrator for context-aware SKG, which received the best demo paper award at IEEE CSCN 2025 [CSCN-2025]. The generated keys are aggregated and coordinated by the Orchestrator, that forwards them to WP4's programmable monitoring platform through CUMU01. The WP4 through zero-touch security management layer feeds back the Upper Layer, to close the loop forwards them to WP4's programmable monitoring platform. The WP4 through zero-touch security management layer feeds back the Upper Layer, to close the loop.

3.2.2 Testbed Requirements and Deployment

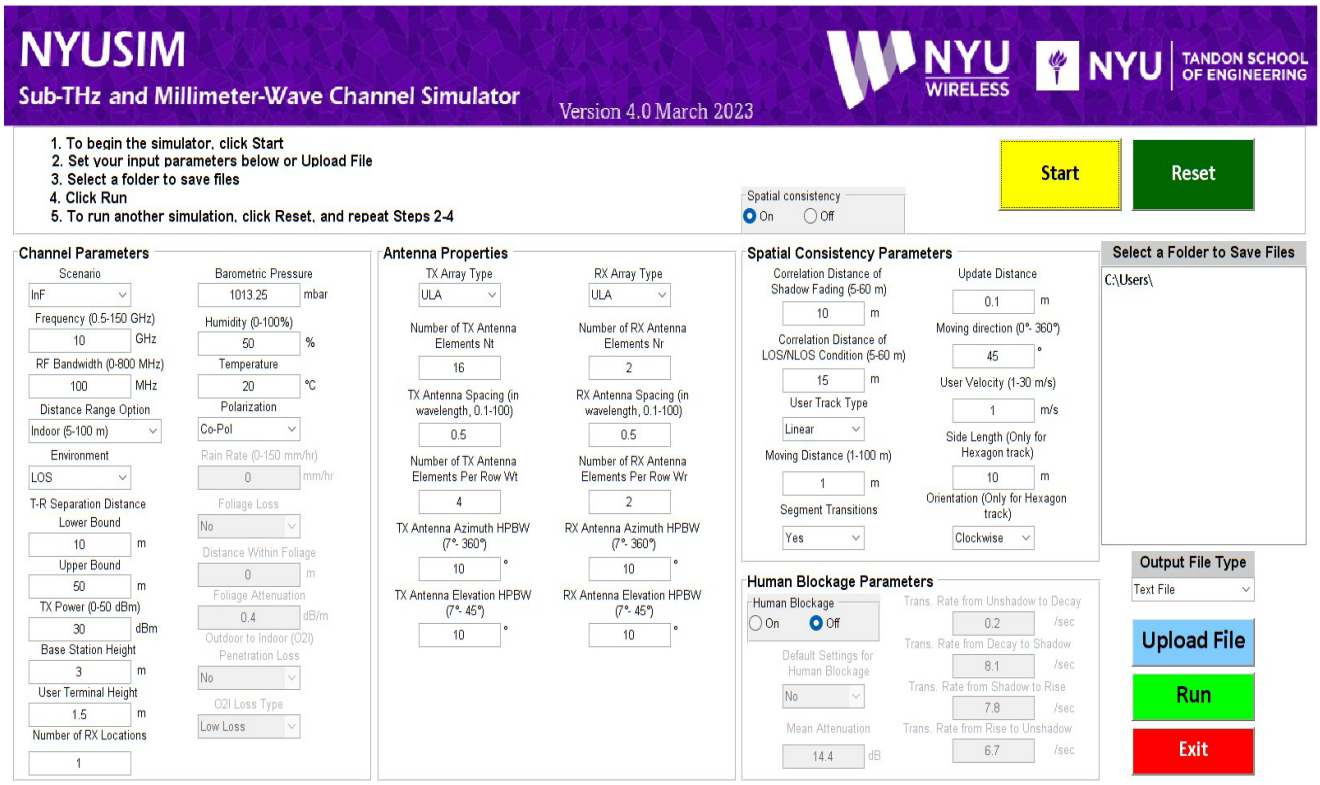

Figure 3-13:Inputs of NYUsim channel model simulator

For the validation of UC1.2, a hybrid approach is adopted, combining both deployable (i.e., CGHM01 and CGHM02) and simulation-based components (i.e., the rest of UC1.2 components). The latter utilizes the NYUSIM channel simulation platform[1][NYU] to generate CSI parameters tailored to the indoor factory scenario, with the main input parameters shown in Figure 3-13.

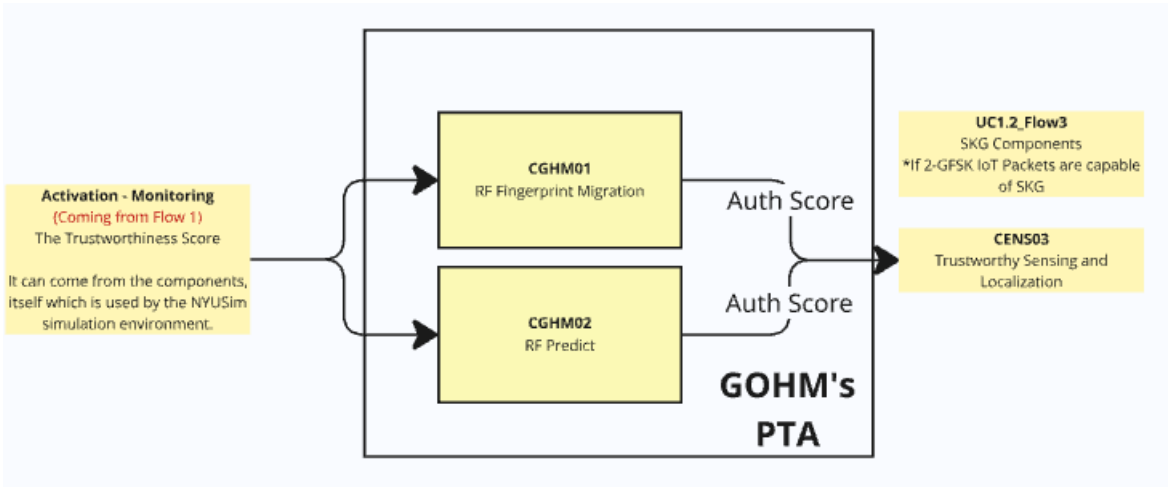

Figure 3-14:Diagram of the integration of deployable and simulated -based components

[1] https://wireless.engineering.nyu.edu/nyusim/

The deployable components are hosted on the GOHM's PTA (i.e., TGHM01, TGHM02 and TGHM03), and their integration with the simulation-based components will be achieved through dedicated interfaces. Figure 3-14, briefly presents an example of the interaction between the simulation and the deployable based components.

3.2.3 KPIs and Validation Criteria

At this stage, we are performing initial validations of individual components against a wide range of RAN specifications (e.g., RF bandwidth, transmission power, number of Tx/Rx antennas, mobility type). The next step will be to assess design choices for interconnecting the different flow components, including the exact determination of component's inputs/outputs.

UC1.2 will put emphasis on the following KPIs, which are also linked to the objectives and the quantifiable targets of WP5:

i) **Evaluation of PHY-layer trustworthiness** through continuous monitoring of physical parameters (e.g., SNR, LoS/NLoS conditions), traffic analysis, jamming attack detection and generation of secrecy maps. A critical KPI will be the system's capacity to accurately identify attacks and dynamically adjust PHY security controls in response. Quantifiable targets: Detection of jamming/interference denial of service attacks with accuracy higher than 90%. Reach a detection accuracy higher than 70% for Sybil attack with the aid of source and device localization and RF fingerprinting.
ii) **Mutual authentication** between access points and devices (such as robots, drones, or sensors), alongside device-to-device pairing in M2M scenarios (e.g., robot-to-robot communication). The associated security guarantees will also be communicated to the security orchestrator as part of the KPI assessment. Quantifiable targets: more than 90% accuracy in PLA, 6G resilience 9e.g., to impersonation and jamming attacks) will be increased at least by 20% through the proposed mitigation techniques (compared to a benchmark without any PLS solutions for identification and mitigation).
iii) **Fast secret key generation (SKG)**, with over 99% reconciliation success and under 5 msec execution times. These metrics, initially defined in the ROBUST-6G proposal, are intended to highlight the advantages of physical-layer security (PLS) over conventional cryptographic approaches.

The possible extension of the quantitative targets and KPIs along with their validation will be further discussed in WP5. Note that, KPI development will progress incrementally, in this context potential updates and / or methodological clarifications will be reported in the next deliverable of WP5 (D5.2).

3.2.4 Flow Progress Tracking

The Table 3-4 presents the intermediate progress results for three flows addressing physical and sensing layer trustworthiness. The flows target PHY-level robustness evaluation, mutual authentication, and secret key generation. All flows have been defined with their corresponding components assigned and mapped to the NYUSim Platform for testing. Integration work has not yet started, but the flows are structurally prepared for execution once component readiness is achieved. These results confirm that the foundations for PHY-layer validation are in place, with the next phase focusing on actual deployment, inter-component integration, and KPI-based assessment.

Table 3-4:Use Case 1 Scenario 2 Flow Progress Tracking Table

ID	Included Components		Integration %	PTA	Status
UC1.2_1	CENS01	CENS03	30	NYUSim	Partial integration ongoing
	CENS01	CUPD05	5	NYUSim	Integration not yet started

	CENS02	CENS03	10	NYUSim	Integration planned
	CENS02	CUPD05	5	NYUSim	Integration not yet started
	CUPD03	CENS03	0	NYUSim	Integration not yet started
	CUPD03	CUPD05	10	NYUSim	Integration planned
	CEBY04	CENS03	0	NYUSim	Integration not yet started
	CENS03	CUPD05	5	NYUSim	Integration planned
	CUPD05	Activation	0	NYUSim	Integration not yet started
	Activation	RAN Control	0	NYUSim	Integration not yet started
	Activation	CCHA01	0	NYUSim	Integration not yet started
	RAN Control	CUMU01	0	NYUSim	Integration not yet started
	RAN Control	PHY Layer	0	NYUSim	Integration not yet started
	CCHA01	CUMU01	0	NYUSim	Integration not yet started
UC1.2_2	CCHA02	CGHM02	0	NYUSim – TGHM03	Integration not yet started
	CGHM02	CGHM01	10	TGHM01 & TGHM02	Integration planned
	CGHM01	CUMU01	0	NYUSim	Integration not yet started
	CGHM01	PHY Layer	10	TGHM03 – NYUSim	Integration planned
	CENS04	CUMU01	0	NYUSim	Integration not yet started
	CEBY05	CUMU01	0	NYUSim	Integration not yet started
UC1.2_3	CENS05	CUMU01	5	NYUSim	Integration planned
	CUPD04	CUMU01	0	NYUSim	Integration not yet started

	CUMU01	PHY Layer	0	NYUSim	Integration not yet started

4 UC2: Automatic threat detection and mitigation in 6G-enabled IoT environments

The Use Case 2 (UC2) of the ROBUST-6G project focuses on the automatic detection and mitigation of threats in 6G-enabled IoT environments, as already described in previous documents D2.2 [ROB24-D22] and D6.1 [ROB25-D61]. The aim of this UC is to enhance the security of IoT systems, rapidly evolving in several sectors, which presents an increasing wide surface of attack and vulnerabilities to be exploited by attackers.

To overcome these challenges, UC2 focuses on three main key points. The automation of the security management process in 6G networks, the development of mechanisms to detect and mitigate threats in real time, as well as the programmability and flexibility of 6G to improve security orchestration. Additionally, according to the D6.1, several challenges have been faced during the implementation of the UC and addressed via targeted design and implementation strategies. To manage the high complexity of closed-loop processes and coordination a new dedicated component was introduced as well as standardized workflow enabling easier integration. Similarly, for an accurate threat detection, advanced AI algorithms were studied as well as the study of the simulation of ad hoc datasets. The objective was to improve the anomaly recognition and to reduce false positives. For the threat mitigation lightweight strategies were proposed, meeting time requirements.

The methodology proposed is based on a "closed-loop" of monitoring, analysis, decision, and action already introduced both in D4.1 [ROB24-D41] and in D6.1. In particular, in this use case, there are three scenarios, but the idea behind them is similar. The proposed solution continuously monitors system log, user activity, and sensor measurement in order to detect, as soon as possible, anomalies and apply resolutive mitigation plans. The implementation of UC2 foresees the use of programmable CL-functions that operate as "agents" in the target infrastructure in order to achieve automatic security proactive and reactive actions.

The validation of the UC2 is measured through KPIs such as the accuracy in the detection and the mitigation time. For each of the following scenarios, the validation criteria and, whenever possible, also the relative measures achieved with the use of ROBUST-6G components are reported. The functionalities developed in UC2, in fact, are integrated with others from different work packages. In particular, data management, security orchestration, and resource orchestration are mainly from WP4, which are very detailed in D4.1 and in the successive D4.3. Additionally, the proposed solution includes both a "zero-touch" automated solution and a human-Supervised solution, depending on the severity or just as a validation by an expert of the proposed mitigation plan.

4.1 Scenario 1 - Device violation to cause an economic harm (a)

The scenario has already been discussed extensively in previous deliverables such as D2.2 and D6.1, but this document aims to provide a more technical version of the implementation, and its validation achieved through the joint work of the other technical work packages. To briefly remind the context of this scenario, it is enough to think that it takes place in a small office environment, which uses centralised IoT platforms to manage devices such as gateways, heating systems, and temperature sensors. The main threat is an attacker who takes advantage of IoT platform vulnerabilities, like a weak password, and, for example, uses legit commands to turn on a heating system during holidays, which could cause unnecessary energy consumption, equipment damage, and financial losses.

The security orchestration capability of the ROBUST-6G framework is designed to identify these anomalies by combining sensor measurements, such as temperature, and network data, such as IoT platform logs, and

mitigate the threat with the most suitable remediation plan. To address this challenge, the scenario applies a single closed-loop process, composed of four closed-loop functions, as shown in the following flows.

4.1.1 Functional Flows description and mapping

According to the methodology proposed in Section 2, the implementation of this scenario is achieved through three "functional flows".

- Flow 1 can be identified as "**Proactive Security Enforcement**". The objective of this flow is to configure the target environment (smart home) to have a minimum of security up-and-running. As the name suggests, this is a proactive step since it is executed as a prerequisite, and technically speaking, it consists of the deployment of the four CL-functions in the target infrastructure.
- Flow 2 can be identified as "**Monitoring and Analysis**". It consists of the execution of the reactive part of the previous defined two CL-functions. Basically, it consists of threat detection by combining network and IoT data properly monitored.
- Flow 3 can be identified as "**Decision and Execution**". Similarly to the previous flow, it consists of the reactive part of the other two CL-functions defined in the first flow. It is the execution of the selection of the best actions to apply in order to mitigate the identified threat.

To achieve the proposed scenario, the ROBUST-6G framework uses several components implemented by different partners that interact with each other in order to guarantee the security of the environment. In particular, Figure 4-1 reports the global ROBUST-6G architecture with a mapping, highlighted in red, of the functionalities validated through this scenario. Additionally, Table 4-1 reports the description of the components in the architecture as well as a brief explanation of the main interactions with other components.

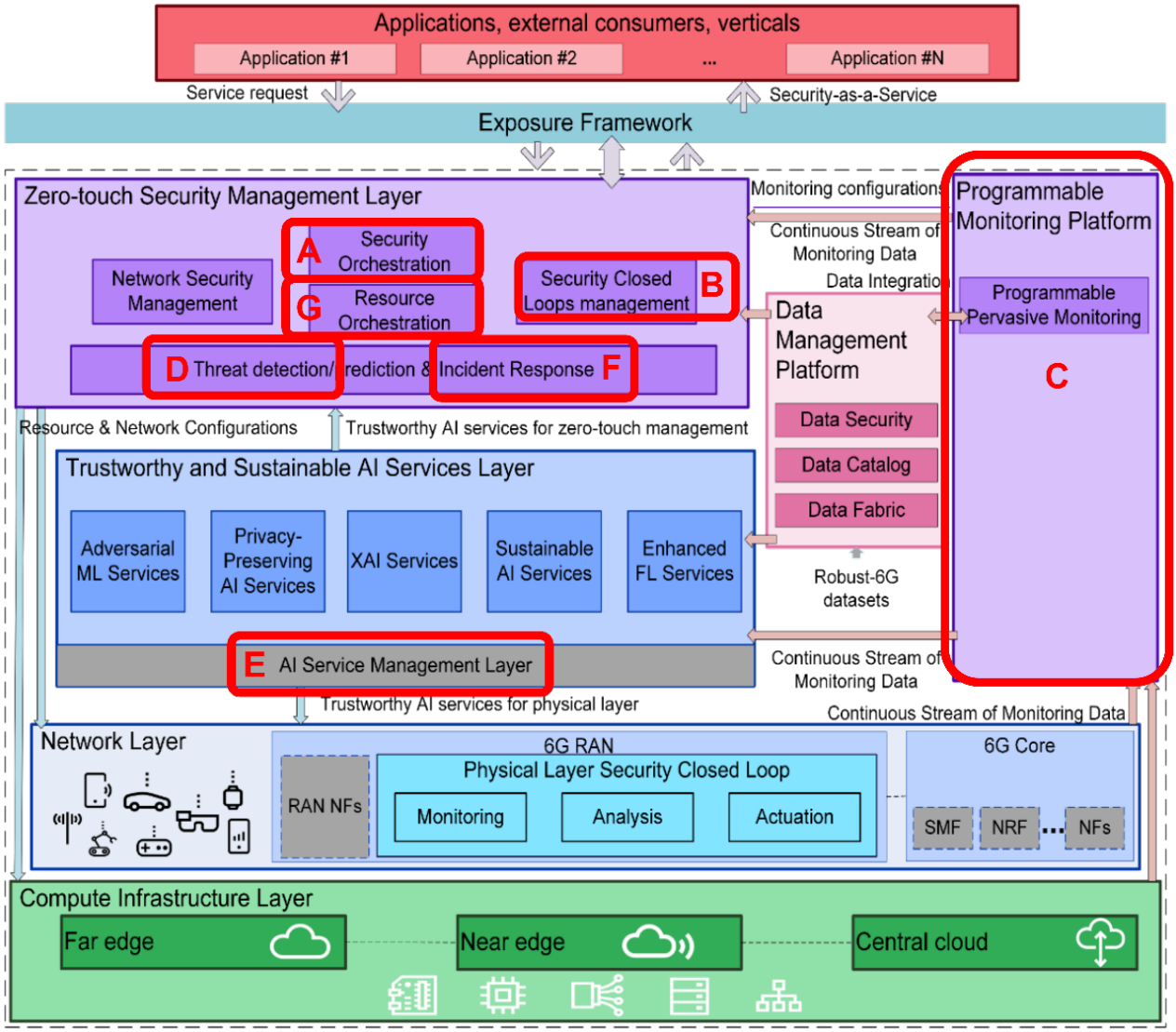

Figure 4-1: ROBUST-6G architecture validated in UC2 Scenario 1

Table 4-1: ROBUST-6G Components implementing UC2 - Scenario 1

ID	Component Name	Description
A	CTHL 01	The core functionalities of this security orchestrator are developed by both Thales and Nextworks and will work together to fulfil the scope described above. This component from Thales addresses the core features responsible for SSLA ingestion from the 6G exposure framework, SSLA translation to security policies and security KPIs, and finally develop a dynamic and adaptative preparation and response workflows for security closed loops.
A	CNXW01	The core functionalities of this security orchestrator are developed by both Thales and Nextworks and will work together to fulfil the scope described above. This component from Nextworks addresses the core security orchestration ontology, security functions and infrastructure catalogues, and the ingestion of security plans (both proactive and reactive) with their required security levels, to dynamically

		compose and deploy security functions and security level monitoring closed loop with a semantic translation approach based on the ontology and catalogues
B	CNXW04	This component takes care of the management of the functions requested by a closed-loop. It registers and continuously checks the status of the Monitoring, Analysis, Decision, Execution "agent" of a loop as well as the cohesion of multiple functions from different loops.
C, D	CUMU01	The Programmable Monitoring Platform (PMP) is an automated solution for managing service and resource health, possible anomaly detection by a rule-based IDS, aggregating data from multiple sources, and enabling dynamic configuration through virtualization techniques. It enables efficient monitoring and closed-loop management of service performance. In addition, it also delivers information, which may be extracted from other collection entities or directly gathered in them, to be consumed by other components.
D, F	CAXN01	This component employs network traffic data and telemetry data collected from IoT devices in IDS datasets, in conjunction with multi-label AI models, to identify threats and generate appropriate mitigation actions.
E	Global Models Repository	This component is a unified component exposed by the AI layer. Robust models could be queried from such repository to be used during the analysis stage.
G	CNXW03	The Resource Orchestrator component manages computing resources in a target environment. It facilitates the efficient deployment and allocation of resources, ensuring optimal performance and scalability in the system's infrastructure.
F	CTHL02	The Monitoring and Closed-Loop Remediation System's component is part of the *zero-touch security management layer*. It ingests security alerts and is assisted with AI to detect security policy violations and propose a remediation plan.

UC2_1_01 "Proactive Security Deployment"

The workflow represented in Figure 4-2 illustrates the orchestration of a security service deployment across multiple specialized components, each responsible for a distinct aspect of the process. It begins with an application requesting a security service, like an SSLA, through the Security Orchestrator (CTHL01). This first step establishes the functional demand and triggers a chain of orchestration activities (CNXW01, CTHL02). The orchestrator then validates the request, ensuring that all prerequisites are satisfied before moving forward. Additionally, it creates a list of plans to execute in case of predefined attacks (step 7). Once validated, the orchestrator asks the deployment of the security service automation, coordinating with the CL-management (CNXW04). At this stage, the workflow explicitly incorporates zero-trust principles, where access permissions and policies are hardly checked, until the proper security posture is confirmed. In particular, the above figure essentially depicts the iterative requests for the closed-loop functions deployment. These requests are made to the Closed Loop Governance component (CNXW04), responsible for enforcing compliance policies and ensuring that deployments respect the governance rules. Governance validation acts as a safeguard that prevents unauthorized or non-compliant deployments from propagating further. During its operation, the CL-management may interact with external components like the Global Model Repository for deploying a ML model compliant with the security service to apply (step 12). Continuing the inspection in the flow, once governance has given the green light, the request is handed over to the Resource Orchestrator (CNXW03). This component manages the underlying compute, network, and storage resources, effectively translating abstract policies into concrete resource commands and allocations. The smart building may be implemented in Kubernetes and represents the computing resources of the infrastructure available. It acts as the execution layer where the actual containerized agents of the CL-functions are running.

The result of the workflow is a deployment pipeline that is not only automated but also strongly aligned with zero-trust and governance-driven paradigms, minimizing risk and guaranteeing compliance by design. This preliminary pipeline prepares the ground for the successive flows

UC2_1_02 "Monitoring and Analysis"

This second workflow, depicted in Figure 4-3, focuses on the operation performed by the closed-loop monitoring and analysis functions. The sequence starts with a reminder of the deployed CL-Monitoring function deployed from the previous flow 1. This deployment request was done from the Resource Orchestrator (CNXW03), which configured and provided the necessary information such that the Programmable Monitoring Platform (CUMU01) can be activated. At this point, the monitoring platform begins by enabling various monitoring resources. According to the UC2 scenario, it starts first IoT monitoring, and successively a network monitoring. This function, once activated, continuously collect telemetry and make the data available in the CL system for the second part of the flow.

Continuing the sequence of the flow from step 8, it emerges the reminder of the CL-Analysis function deployment from the previous flow 1. It is worth to mention that together with the CL-Analysis function the CL-management takes care of deploying the ad-hoc ML models suitable for such security services (see flow 1). Once the ML models, retrieved from the Global Model Repository, are active, the CL-Analysis function can pull IoT and network data streams (step 10, 11). Using such stream of data, it is performed an inference, and the result is forwarded to the successive step of the loop.

In short, this workflow demonstrates how closed-loop monitoring integrates real-time telemetry and ML-driven analytics. The outcome of this workflow is a dynamic system capable of adapting its monitoring depth while simultaneously ensuring that decisions are backed by intelligent, data-driven insights.

UC2_1_03 "Decision and Execution"

This third workflow continues the previous flow 2 by introducing the closed-loop decision and execution functions. The sequence starts with some reminders from the flow 1 about the deployment of the target CL-Decision function and from the flow2 about the results of the analysis on the monitored data. These functions are coordinated by the CL-Management (CNXW04) and executed in the target infrastructure (Smart Building). As discussed in the previous flow, once monitoring and analysis are in place, data from IoT and network domains are processed, and ML models are applied to infer potential threats. If a threat is detected (light red block) the workflow moves into a decision-making phase concerning the mitigation strategies to be chosen. Step 4 of Figure 4-4 shows an interaction with CAXN01 for getting the mitigation actions to execute for mitigate the specific threat according to their AI models. At this point, the CL-execution function comes into play, and some remediation measures are injected directly into the IoT Platform. This branching illustrates a "zero-touch" basic plan can be applied quickly, but for more complex ones a confirmation loop from an expert to ensure correctness is required. Furthermore, to assure a good level backlog and security, notifications are issued to inform the 3rd party application (step 12). Similarly, it is important to note that this dynamic mechanism allows also to distinguish between a cyber-attack and a sensor's failure (step 3, 13).

To summarize, these three workflows depict a fully automated closed-loop system that does not stop at detection. It completes the cycle by generating, validating, and executing mitigation strategies as well as ensures transparency via the constant feedback of notifications. All of this creates a dynamic defence mechanism capable of responding in real time to evolving threats.

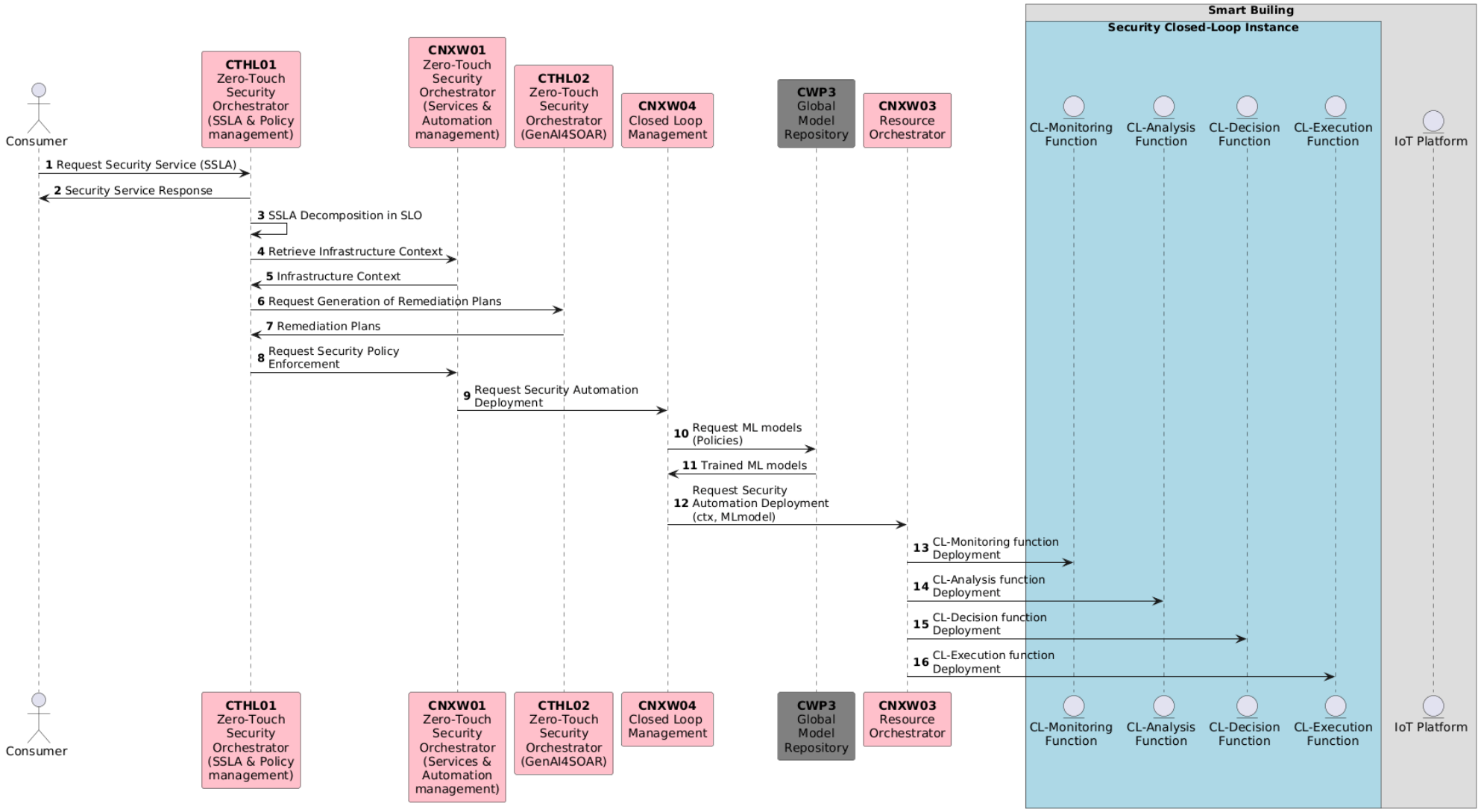

Figure 4-2: Proactive Security Deployment (UC2.1 - flow1.

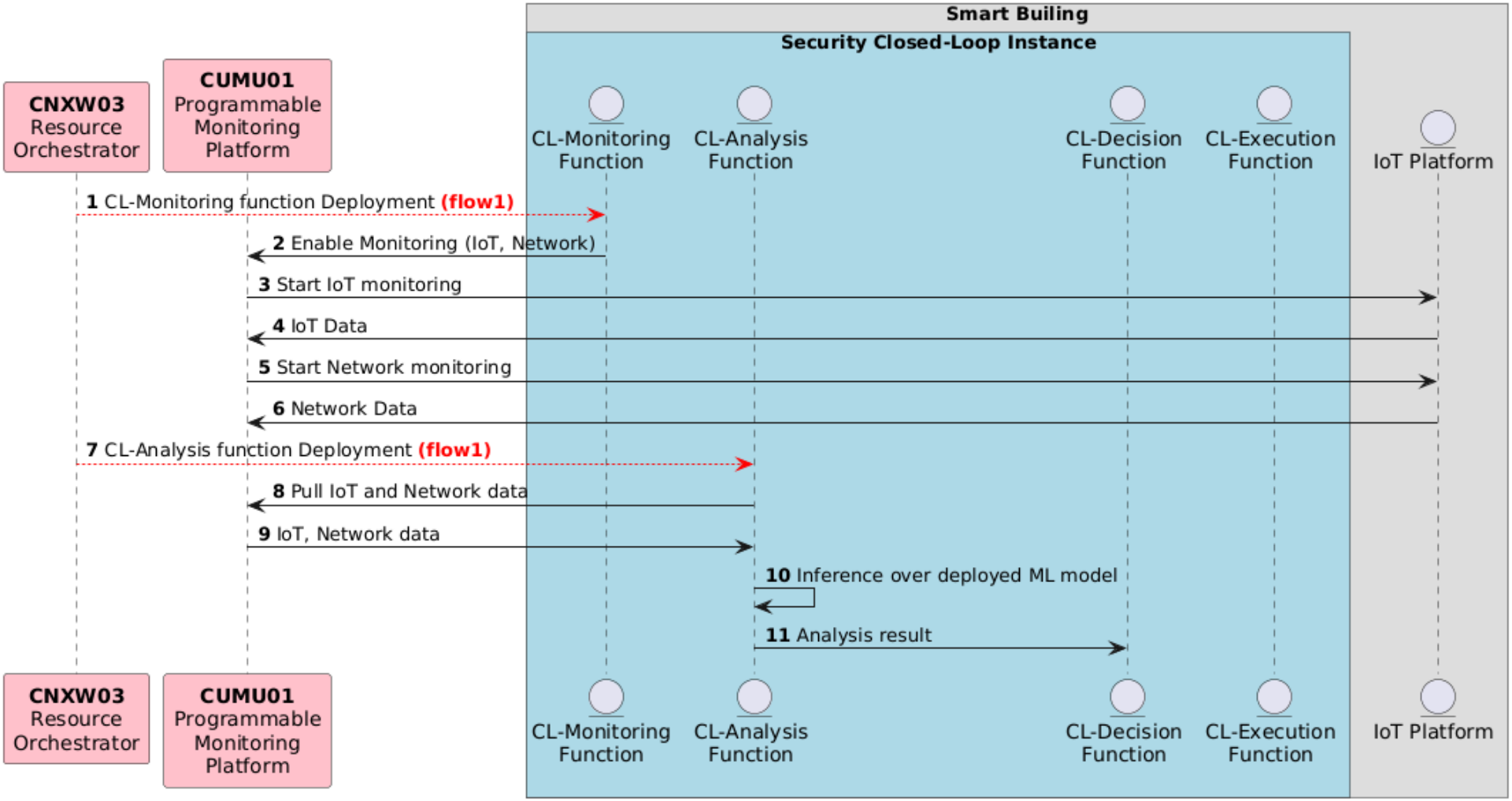

Figure 4-3: CL-Monitoring and CL-Analysis functions run (UC2.1 - flow2).

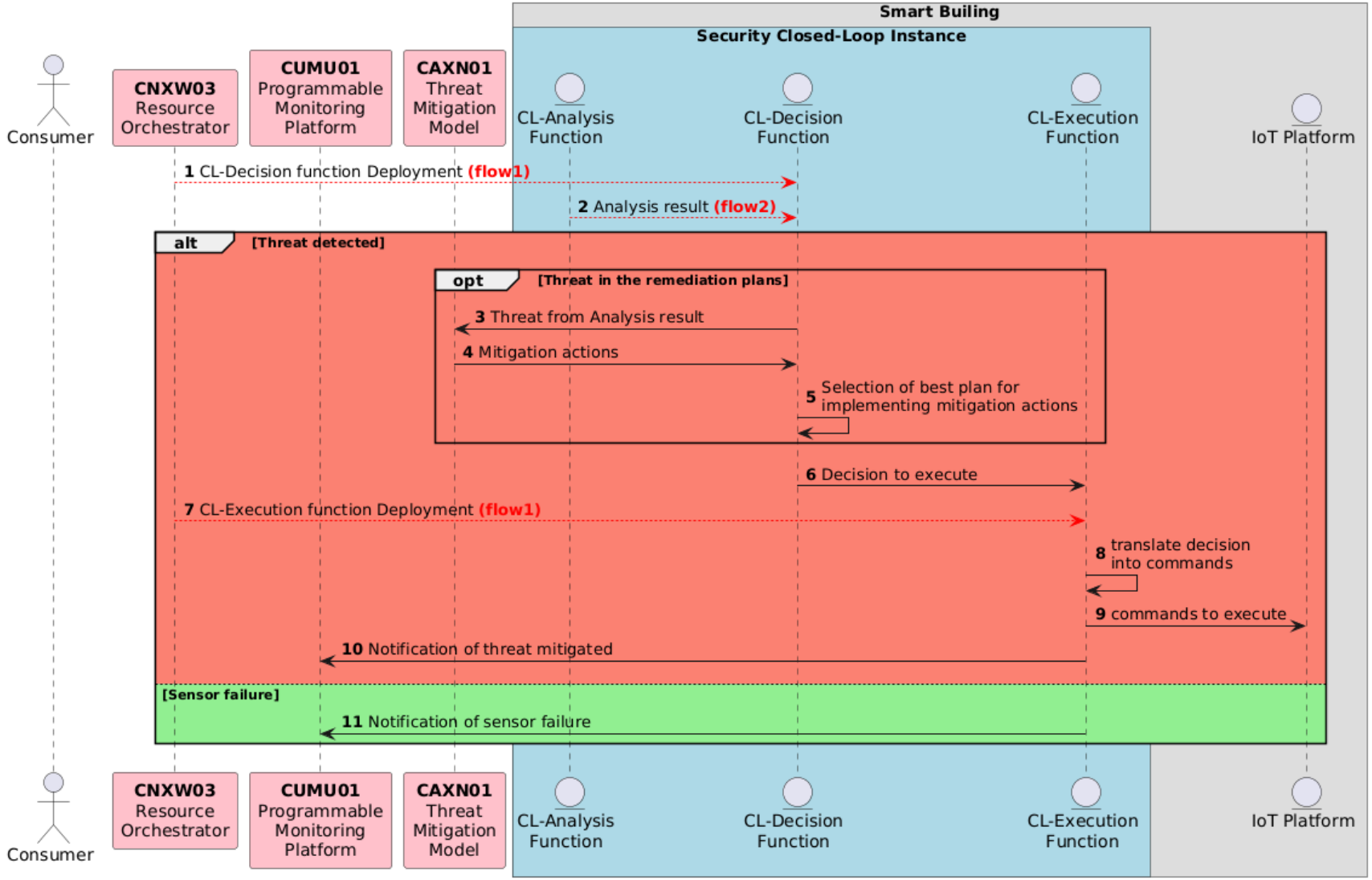

Figure 4-4: CL-Decision and CL-Execution functions run (UC2.1 – flow.

4.1.2 Testbed Requirements and Deployment

The diagram in Figure 4-5 provides a clear overview of the infrastructure and its interconnections within the testbed environment in Nextworks (TNXW01). This architecture focuses on the deployment and management of various components, crucial to the operation and validation of the Scenario 1 of the UC2. For an implementation point of view, the infrastructure – identified by the Smart Building – is realized by Kubernetes while the other components are Docker containers. The components, identified by their respective codes (e.g., CNXW03, CTHL02), interact each other to implement the steps discussed in the previous three flows.

Each component is deployed in the specific testbed (TNXW01) to serve its function:

- **Resource Orchestrator** (CNXW03) and **CL-Management** (CNXW04) handle the orchestration of resources and governance of closed-loop functions respectively, ensuring that the system operates within the configured parameters.
- **Programmable Monitoring Platform** (CUMU01) belongs to the infrastructure layer, closely integrated with other components to provide detailed monitoring capabilities and early detection functionalities.
- **Threat (Prediction and) Mitigation Model** (CAXN01) is deployed within the same infrastructure, contributing to the overall security strategy by providing mitigation recommendations.
- **Security Orchestrator** (CTHL01, CNXW01, CTHL02) plays a pivotal role in managing security-related activities such as SSLA, policies, automations, and remediation plans.

Additionally, at the right of , it is depicted the **Global Models Repository**. It is a component which provides a location from where retrieve ML models to be used in the CL-functions. At the time of this deliverable, the interaction with this component was still under finalization and it was used a local repository containing the necessary ML models.

The system is designed for a highly automated environment where continuous monitoring, threat prediction, and remediation are key operational pillars. The integration of these modules in this testbed ensures that the system remains both responsive and resilient to the simulated threats.

The components are already validated individually, and their results are available in the technical deliverables like D4.3 for WP4 (Zero-Touch Security Management Layer) components. In the context of UC2, not all the listed components have been deployed in the described testbed. However, initial tests confirm stable integration inside the testbed with no significant issues to be reported in the interaction between all the modules. The next phase, described in the successive D6.3 deliverable, will involve complete integration with numerical results.

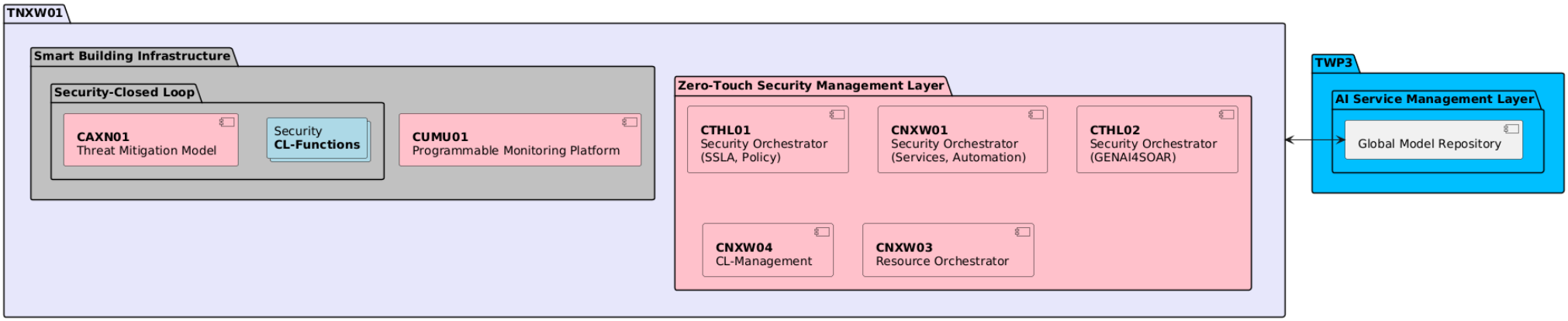

Figure 4-5: Testbed implementing UC2 - Scenario 1

4.1.3 KPIs and Validation Criteria

The KPIs are already described in previous documents D2.2 [ROB24-D22] and D6.1 [ROB25-D61]. These indicators suit well for the workflow described earlier and are still valid for the validation of the scenarios of the use case 2. As a reminder, the list of UC2's KPIs is reported in the Table 4-2.

Table 4-2: UC2 Key Performance Indicator

KPI	Value	Notes
Detection Accuracy	95%	Correctly identified threats ((TP+TN)/(FP+FN)) against for example simple sensor failure.
Detection Time	< 2 minutes	Delta time between injection of threat and its detection.
Mitigation Accuracy	95%	Proposed actions that actually reports the environment in a correct state.
Mitigation Velocity	<= 3 closed-loop	How many "explorative" corrections should be applied before mitigating the threats?
Mitigation Time	< 10 minutes	Delta time between detection of threat and its mitigation.

In order to validate such indicators, the execution of the three workflows described earlier come in place. In other words, the validation of all these KPIs is automatically done once the integration of all the components is completed. On the other hand, the individual component validation as well as the implementation details of each single component will be reported in the following technical deliverables from WP3, WP4, WP5. This deliverable focuses on the integration of all of the listed components for the specific scope of the UC, but since some integrations is not completed, it is not possible to report in this section the numerical results of the experiments. However, it is presented the methodology used in order to validate the scenario.

In particular, the flow 1 is "preparative" since it is used for deploying the CL-functions necessary for the reactive stages. Anyway, such deployment is going to validate the mitigation velocity for this scenario. By design in fact, such metric is satisfied since it includes the presence of a single closed loop which is resolutive for the injected threat.

Flow 2 is necessary for validating the KPIs related to the detection accuracy and the detection time. Such metrics are measured by the analysis result and by reading the logs of the components. In fact, at a certain point of time, the simulated traffic flowing in the IoT Platform will be converted in malicious traffic enabling the presence of network attacks. By measuring the time of the logs between the time such malicious traffic is injected and the time the analysis result state the presence of an attack (or not) it is possible to validate such KPIs.

Similarly, flow 3 is necessary for validating the KPIs related to the mitigation accuracy and mitigation time. By remembering such flow, it is immediate to imagine the validation of the KPIs. In particular, the mitigation accuracy strongly depends on the ML model used for generating the mitigation actions once the threat is identified. Similarly, by measuring the time between the attack injection and the execution of the mitigation commands in the IoT system it is possible to validate also the mitigation time.

4.2 Scenario 2 - Fraudulent usage of device resources

Similarly to the first scenario, also this scenario has already been discussed extensively in previous documents D2.2 [ROB24-D22] and D6.1 [ROB25-D61]. To briefly remind the context of this scenario, it discusses about the risks faced by a smart building, where hackers exploit the vulnerabilities of IoT smart lights for unusual activities like cryptocurrency mining rather than the original lighting functionality.

The system has similarities with the first scenario in terms of the components involved but differs in the nature of the threat. The flexibility of the proposed ROBUST-6G system allows addressing this apparent different scenario with minimal changes. For example, it gives the possibility of instantiating multiple closed loops, or to adapt and customize the CL functions for handling different needs without the need of introduce additional ad-hoc components.

4.2.1 Functional Flows description and mapping

As anticipated in the introduction of this section, this scenario is very similar to the previous one in terms of technical functionalities. The two scenarios belong to the same use case and for this reason they also share its implementation. In particular, the architecture mapping, the testbed setup, and the KPIs validation are valid as reported in the previous Section 4.1, and for simplicity are not duplicated in this section.

In terms of flows deployment, this scenario basically reuses the same principles behind the previous ones (UC2_1_01, UC2_1_02, UC2_1_03) about the proactive security enforcement and the run of the four CL-functions. What it really changes is not the interaction between the components, but the actual content of the request. It also means that the validation methods and the success criteria of these flows are also the same of the previous scenario. The CL-Management, in fact, allows the deployment of programmable CL-functions which may be different depending on the specific scenario. In this particular case, the main difference consists of the presence of two sequential loops:

- Flow 1 can be identified as "**1st investigative loop**". It basically consists in the union of the previous flows proposed in the scenario 1. The difference lies in the commands proposed for the mitigation because the CL-execution function is not resolutive in terms of threat but rather it is investigative because it triggers the scheduling of a second loop which analyse in depth the cause of the anomaly.
- Flow 2 can be identified as "**2nd resolutive loop**". It is dependent on the first flows, and combining the information from the two flows it is possible to declare the presence of the cryptocurrency mining threat and also to propose some resolutive actions to mitigate it.

UC2_2_01 "1st investigative Loop"

Figure 4-6 presents the end-to-end workflow for deploying a new security service in a target environment as well as the running of the CL-functions. Many of the operations are already well described in the previous Section 4.1.1. Briefly, it starts from the translation of a security request and terminates with the execution of a certain action. The CL-monitoring function, in step 12 reports an interaction with CUMU01. In this scenario the function is programmed to start only the monitoring of the CPU and energy consumption. The analysis result of such data brings the CL-decision function to ask for further investigation due to the anomalous pattern. This conservative approach is used to discard false positive threat. The actions executed by this loop translates in the scheduling of another loop. For this reason, the competition of this first loop cannot be considered resolutive, but rather investigative.

UC2_2_02 "2nd resolutive loop"

The second loop is scheduled right after the end of the first flow, as reported in Figure 4-7 (step 1). In this workflow, it is requested a second CL. The second CL-monitoring function is programmed to enable the collection of the network data. This network data, combined with the IoT data from the first flow, allows the second CL-analysis function to state the presence of a cryptocurrency mining attack or not. This triggers the second CL-decision function choose the best actions from the remediation plan to mitigate the threat.

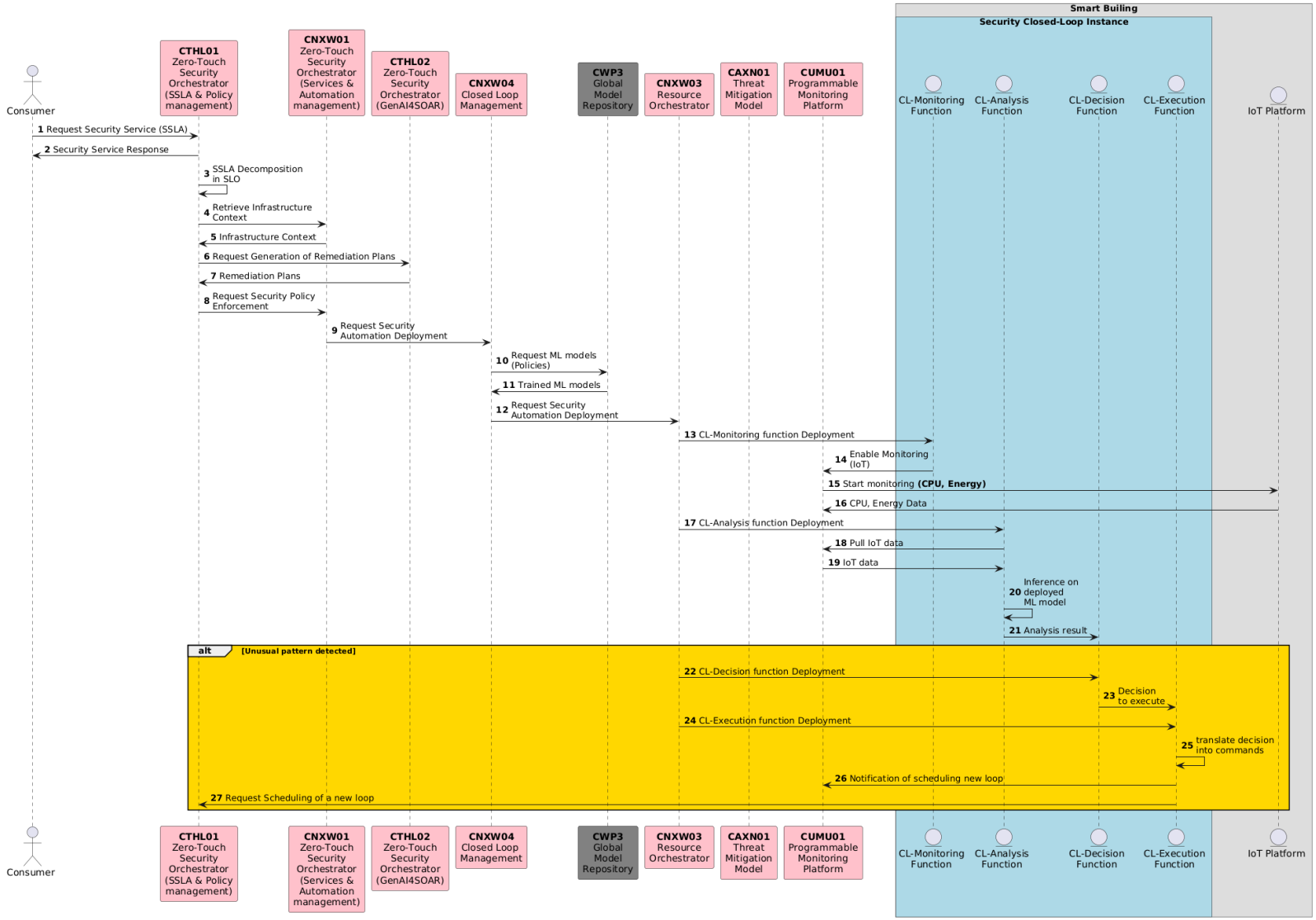

Figure 4-6: Investigative loop deployment and execution (UC2.2 loop 1)

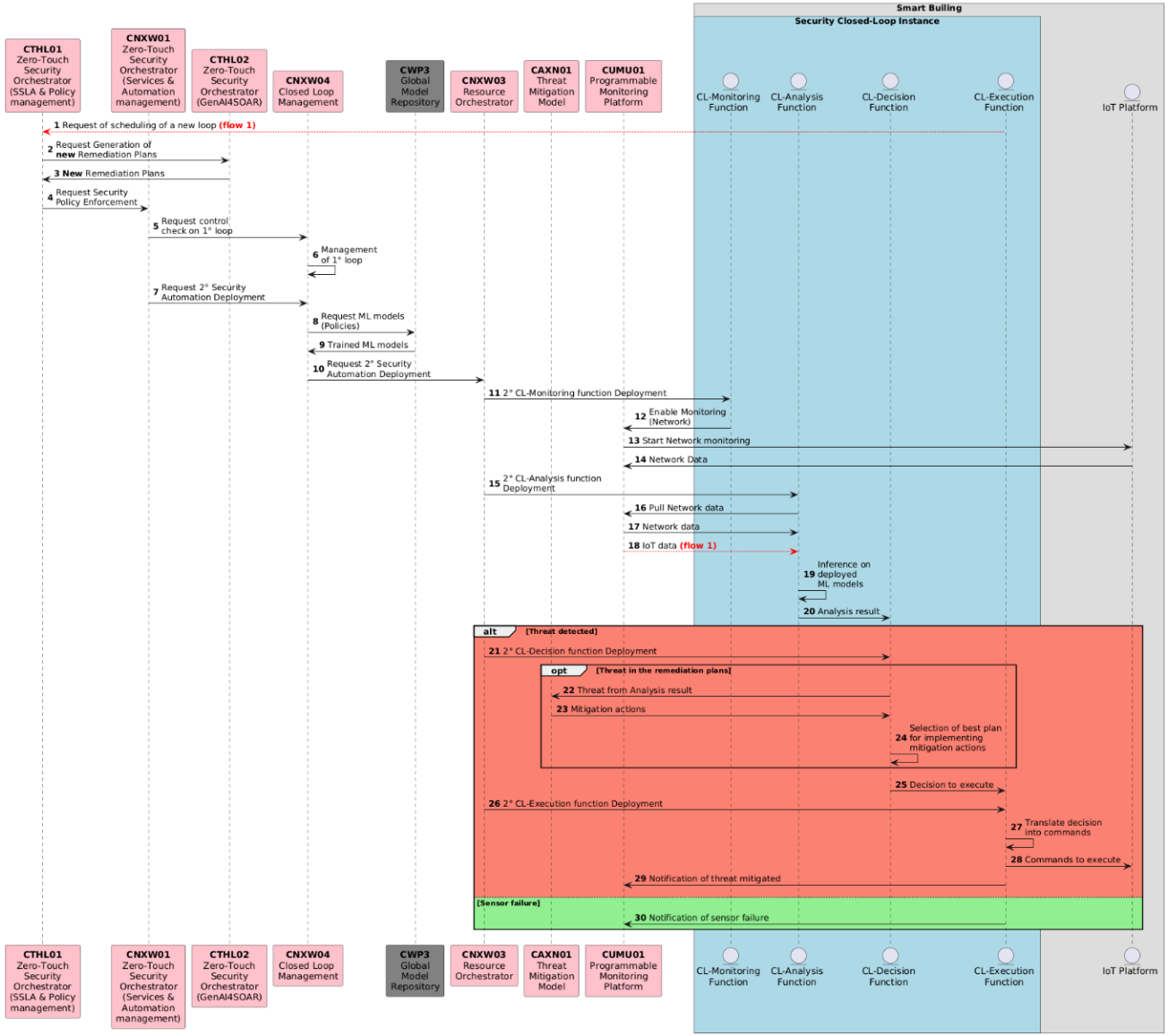

Figure 4-7: Resolutive loop deployment and execution (UC2.2 loop 2)

4.3 Scenario 3 - Device violation to cause an economic harm (b)

This scenario focuses on the consequences of cyber-attacks in smart agriculture as described in detail in the previous deliverables D2.2 and D6.1. The scenario is simulated by imagining two adjacent fields used for growing raw materials such as wheat and cereals. Each field has a temperature sensor, a humidity sensor, and an actuator that controls water irrigation. In this simulated scenario, an attacker exploits temperature sensor vulnerabilities to manipulate the measurements such that the irrigator is triggered even if it is not needed. This, as anticipated many times, leads to significant financial losses and causes environmental damage. In this scenario, to address such a challenge, it is proposed the use of multiple loops. Additionally, it is also evaluated the possibility to use interfaces exposed by the physical layer. For instance, advanced methods such as RF fingerprinting and AoA-based authentication can be introduced to identify unauthorized user attacks. While, jamming threats may be countered through frequency hopping and beamforming techniques enabled by distributed MIMO (dMIMO), designed to ensure reliable communications.

4.3.1 Functional Flows description and mapping

The proposed description of this flow is purely theoretical and may be revisited in future deliverables. The objective here is to formalize the flows even they are still under discussion and to postpone the final version in the future deliverable D6.3.

As introduced before, within each field several IoT sensors are available. Moreover, an “internal” closed loop is scheduled to monitor the status of the sensors and intervene if something is abnormal. In normal conditions, the data collected within each field are accumulated by the Data Fabric, which acts as an external entity that watch over all fields and could be used as comparator. On the other hand, if an anomaly is detected, an “external” loop is scheduled to address the problem.

This scenario aims at addressing the coverage of the majority of the ROBUST-6G platform functionalities as reported in Figure 4-8.

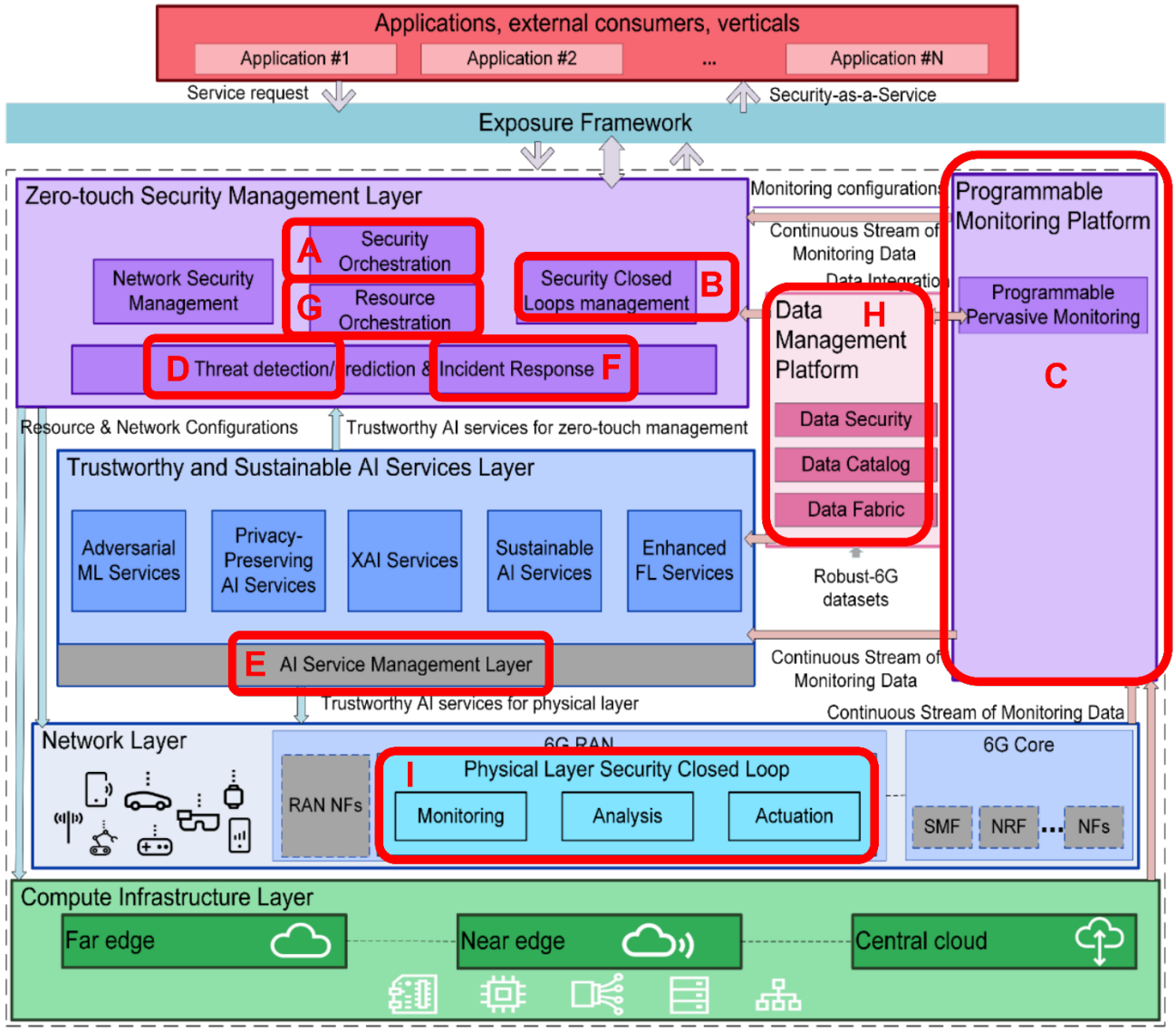

Figure 4-8: ROBUST-6G architecture validated in UC2 Scenario 3

Figure 4-8 presents many similarities with Figure 4-1. In the following Table 4-3 it is reported only the different components not described earlier. For the rest of the components, the description remains the same.

Table 4-3: ROBUST-6G Components implementing UC2 - Scenario 3

ID	Component Name	Description
H	CTID01	Data Fabric handles the collection, processing, and storage of security-related data while also ensuring this data is accessible to its intended consumers.
H	CTID02	The Data Governance module provides mechanisms for cataloguing and authorizing data access based on defined policies. It ensures high-quality data, privacy, and secure access in alignment with the requirements of data domain owners, while also facilitating data discovery.
I	Physical Security Layer	This component is a unified component exposed by the Physical layer. Physical closed loops could be accomplished using the interfaces exposed by this

		component. Such local loops could achieve functionalities like secret key agreement, mutual authentication and physical layer trustworthiness evaluation.

This flow, has been designed to be implemented with the following flows:

- Flow 1 about the "**internal**" closed loop inside the single field A (or B). This flow would imply the same structure and components reported in UC2_2_01, with the addition of the orange components for Data Management that will be used across multiple fields.
- Flow 2 about an "**external**" closed loop. This loop is triggered by the previous flow, and it is used to verify the presence of a sensor attack and to remediate to it like it was done in UC_2_02 by combining the information from adjacent field provided by the Data Management or by using the Physical Closed loop layer.

UC2_3_01 "Internal loop"

Figure 4-9 depicts the end-to-end workflow describing the deployment of a closed loop in a smart factory monitoring a field and reacting if something is anomalous. In particular as described in the previous similar flows, the consumer asks for the hardening of the system by requesting the deployment of a security service in each of the two fields available in the scenario identifies as "field A" and "field B". The security orchestrator (CTHL01, CNXW01, CTHL02) takes care of the incoming request and by interacting with the CL-Governance (CNXW04) and the resource orchestrator (CNXW03) it deploys the necessary entity in the target infrastructure (step 8, 13, 23, 26). The second part of the flow is the one tailored to the scenario. The CL-monitoring triggers the monitoring platform (CUMU01) for monitoring the sensors value in the field where it is deployed. Successively, the monitoring platform forwards such metrics to the data management (CTID01, CTID02) which will act as aggregator of information from data coming from different fields. The CL-analysis studies the pattern of the monitored data with a ML-model provided by the WP3 platform and provides the output to the CL-decision. The latter function decides the best actions to perform address the unusual pattern. In this particular scenario, the decision consists in deploying another "external" loop for compare the data with adjacent field or with some information coming from the physical layer

UC2_3_02 "External loop"

Once the CL-decision of the "internal" loop in a field (A or B) detects that some further investigation is required, it triggers the scheduling of another "external" loop which could be used as ground through for determine the presence of an attack and mitigate it. The workflow describing this resolutive loop is reported in Figure 4-10.

Figure 4-10 report the usual workflow in the initial part executed after the reaction of the previous loop (internal to a particular field). In the second part instead, two alternatives are proposed. Such alternatives consist in using the Data Management Layer (step 8 to step 33) which contains data from all fields, and this could be used to compare the data or using the Physical closed loop layer (step 34 to 43) to use low level functionalities.

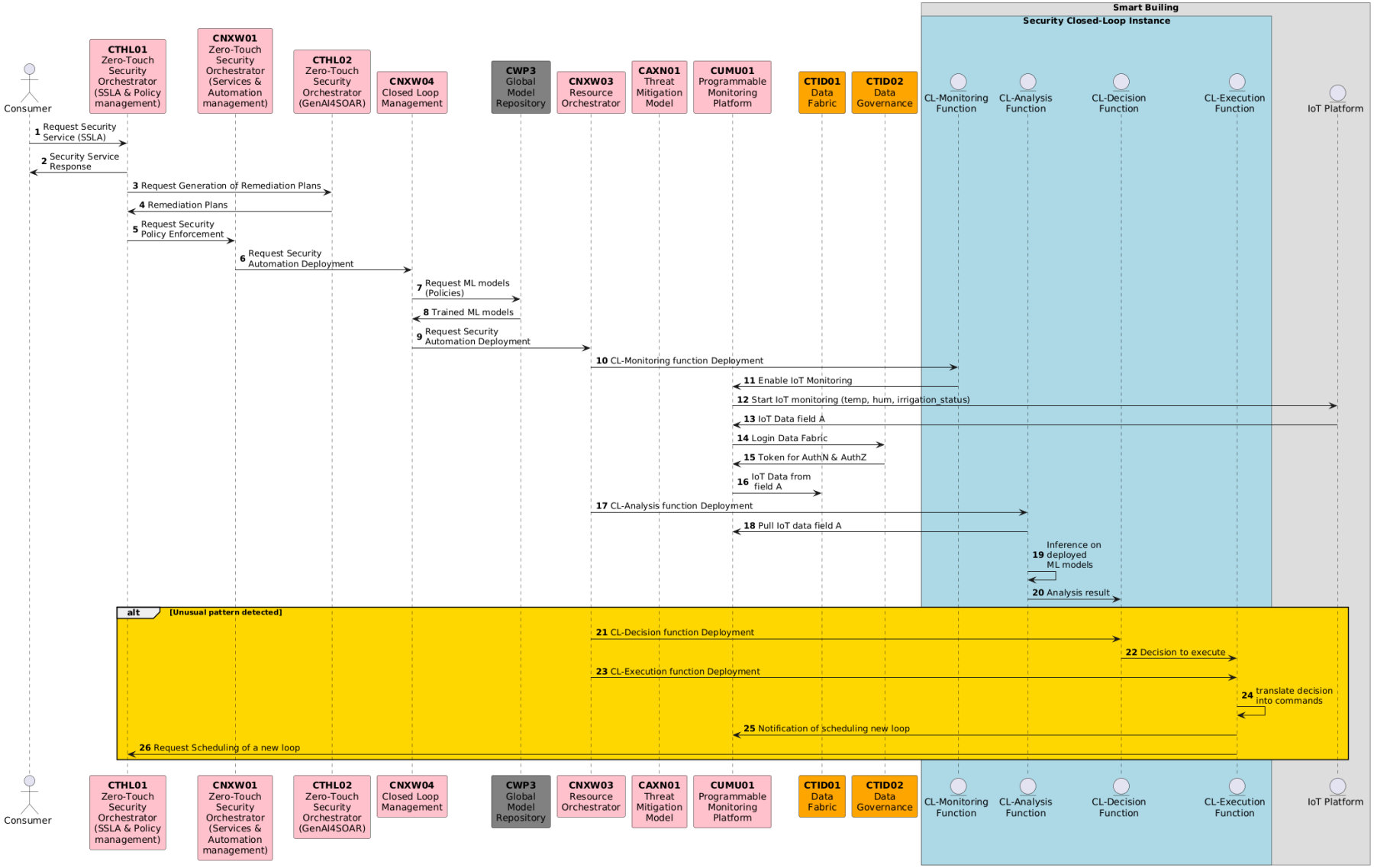

Figure 4-9: Internal loop for field A/B (UC2.3 loop1)

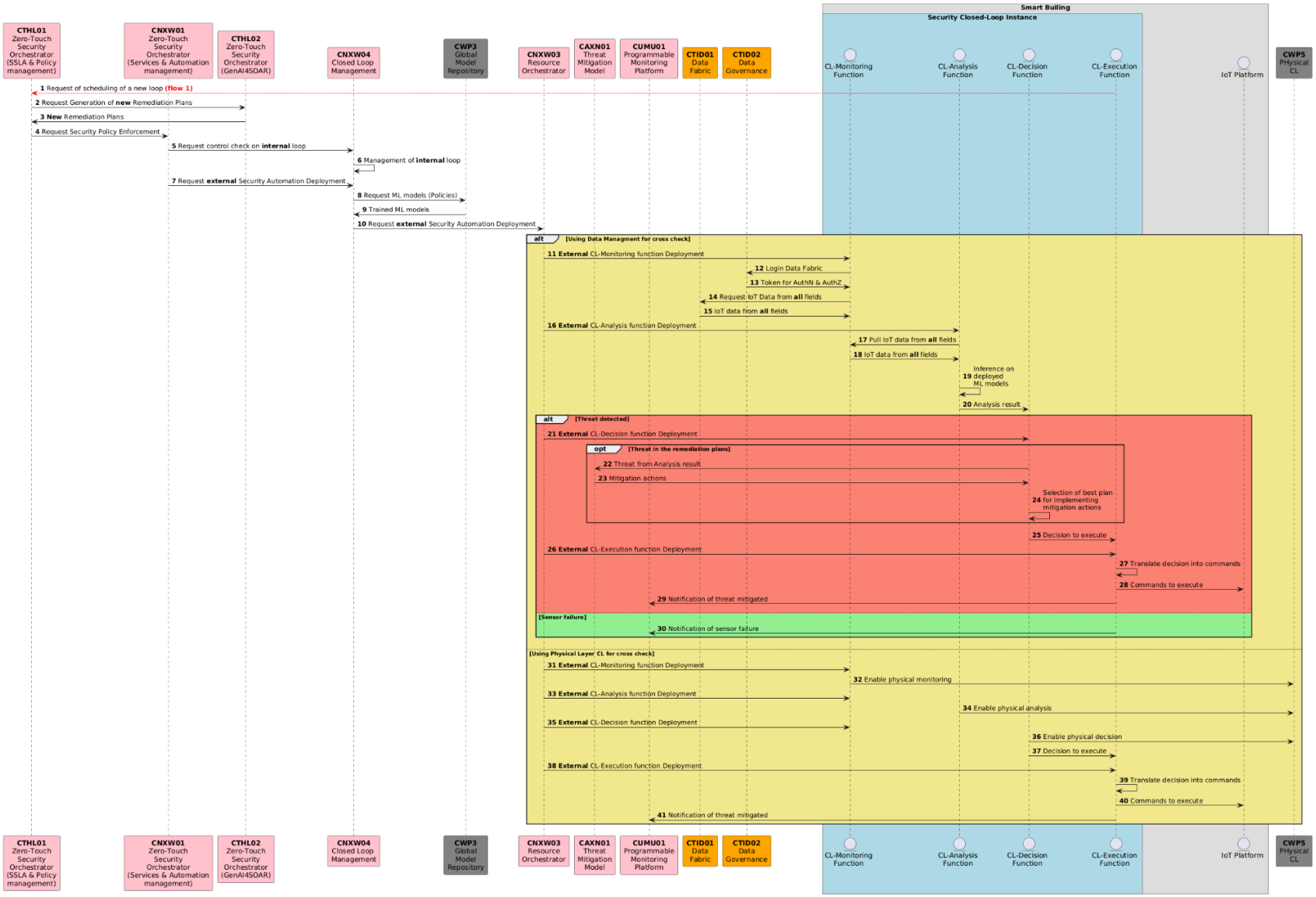

Figure 4-10: External loop for verification (UC2.3 loop2)

4.3.2 Testbed Requirements and Deployment

Figure 4-11 provides the planned interconnected testbed needed to demonstrate the scenario 3. Such theoretical testbed is still to be completed and agreed as anticipated earlier and some links may change in the future. It depicts the already introduced testbed from Nextworks within all the components introduced before, but in this case, it also contains the interconnection with TID01 to enable the presence of the Data Management Layer and the interconnection with WP3 and WP5 in order to exploit the functionality offered by the AI layer and the Physical layer, respectively.

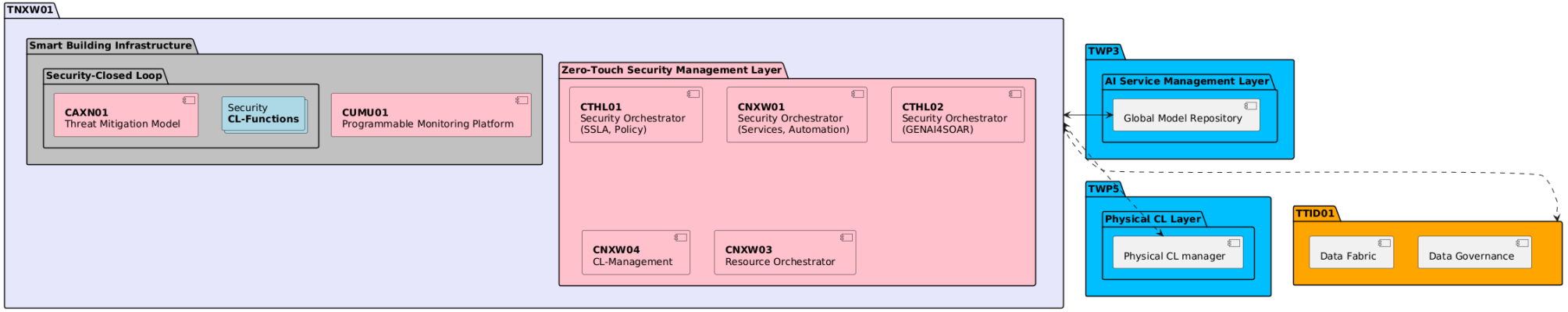

Figure 4-11: Planned testbed implementing UC2 - Scenario

4.3.3 KPIs and Validation Criteria

The KPIs are the same reported in Table 4-2 as they are generic for the whole UC2 and are applicable in all three scenarios. In particular, in order to validate such KPIs, the same validation criteria reported in Section 4.1.3 are still valid. The difference lies in the presence of additional components such as the data management layer and the physical layer. In order to verify also these new interconnections, additional criteria could be evaluated as follows.

1. Time to retrieve data from different fields (A, B) from the Data Management Layer
2. Time to retrieve data from certain asset at the physical layer
3. Time to request authentication at the physical layer
4. Time to request a secret key generation at physical layer

Unfortunately, at this time the numerical measures of such KPIs or of the criteria are not available since they are still under discussion of the technical implementation that will be available in the future. In any case, such numerical measurements or a revised version will be completed in the following months and will be reported in the next deliverable.

4.4 Flow Progress Tracking

The Table 4-4 shows the intermediate results for three flows within UC2 scenario 1, which address proactive enforcement, threat detection, and reactive mitigation. Several components (e.g., CTHL01, CTHL02, CUMU01) have reached partial or full integration within the TNXW01 testbed, while others are still under development or require clarification of APIs and datasets. Key open issues include policy management for proactive enforcement, dataset availability for detection, and the definition of remediation actions for mitigation. Overall, the flows are progressing, with partial functionality demonstrated, but further integration and coordination are required to achieve full closed-loop validation.

Table 4-4: Use Case 2 Flow Progress Tracking Table

ID	Included Components		Integration %	PTA	Status
UC2.1_1	Consumer	CTHL01	100%	TNXW01	Presented an SSLA for enhancing the security in an infrastructure
	CTHL01	CNXW01	75%	TNXW01	A single security orchestrator has been created. The functionalities are distributed among the partners
	CTHL01	CTHL02	100%	TNXW01	A single security orchestrator has been created. The functionalities are distributed among the partners
	CNXW01	CNXW04	25%	TNXW01	Ongoing integration
	CNXW04	CWP3	25%	TNXW01	Not clear which API is available
	CNXW04	CNXW03	25%	TNXW01	Ongoing integration

UC2.1_2	CL-Monitoring	CUMU01	75%	TNXW01	Working on a dataset compliant to the UC2 scenario
	CUMU01	IoT Platform	50%	TNXW01	Working on infrastructure configuration (ThingsBoard)
	CL-Analysis	CUMU01	25%	TNXW01	Can start after dataset is ready
	CL-Analysis	CL-Decision	25%	TNXW01	Can start after dataset is ready
UC2.1_3	CL-Decision	CAXN01	25%	TNXW01	To clarify how to put action in remediation plans
	CL-Decision	CL-Execution	25%	TNXW01	Can start after dataset us ready
	CL-Execution	Infrastructure	25%	TNXW01	Working on setting up the infrastructure (Kubernetes)

As anticipated during the previous sections regarding scenario 2 and scenario 3, it is premature to report the integration status in the table. This is because these scenarios are not mature enough at the moment and several discussions are still pending. However, the majority of components present in such flows are the same of the scenario 1 and they will be reused. This translated in considering the interconnection status similar to what it is reported in the above table also for the other table.

5 UC3: Security Capabilities Exposure with Network-Security-as-a-Service (NetSecaaS)

5.1 Scenario Summary

The objective of this UC is to implement and validate a secure exposition to third-party applications through a proof-of-concept deployment (Figure 5-1). This validation can demonstrate the practical application of the Network-Security-as-a-Service (NetSecaaS) concept within the Open Gateway framework [OPG25]. The exposable capabilities leverage intuitive APIs in accordance with the CAMARA project [CAM25] style and level of abstraction. The integration abstracts complex security features, thereby enabling users, such as application developers and enterprises, to apply security policies without requiring extensive network expertise.

The architecture, drawing inspiration from Open Gateway, incorporates key components such as the Exposure Gateway and the Transformation Function within the integration layer. These components serve to mediate interactions, enforce security standards, and ensure controlled access to network resources. Moreover, the Data Fabric platform administers data flows between ROBUST-6G and the exposure framework, thereby facilitating real-time security data exposition. In addition, it enables the execution of workflows based on user-defined intent declarations, should the necessity arise. This framework underscores the imperatives of efficiency and security, thereby aligning with the objectives of 6G networks.

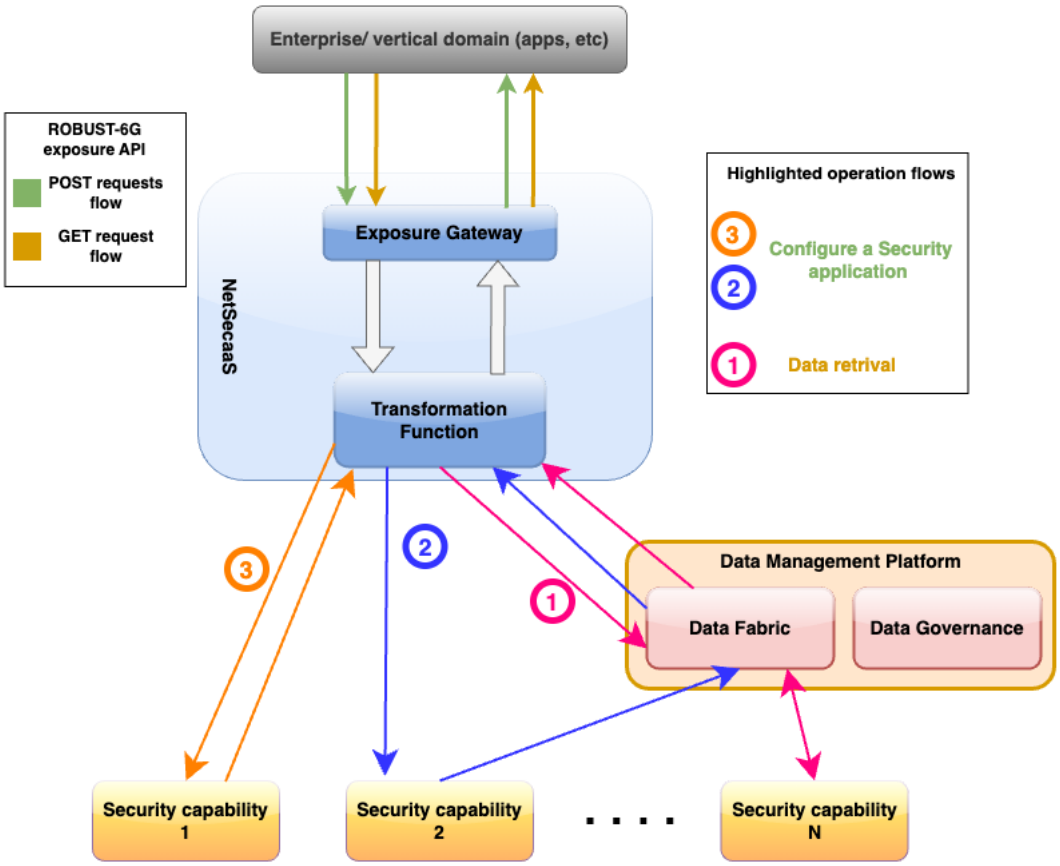

Figure 5-1: UC3 scenario overview

5.1.1 Functional Flows description and mapping

This section introduces the functional flows that underpin the Network-Security-as-a-Service (NetSecaaS) exposure use case. The objective is to provide an account of the integration, exposition, and interaction of each security capability with third-party software applications by means of intuitive APIs exposed through the Open Gateway like framework. By mapping these flows, the conceptual overview and the practical implementation are bridged, thereby enabling stakeholders to understand how security services can be abstracted and orchestrated efficiently.

The general use case, illustrated in Figure 5-2, captures a high-level functional flow and component mapping for the exposure of security capabilities as a service within a 5G/6G data management platform. The system is interacted with by a "3rd Party" user at the top, specifically with the CTID03 component, "Security capabilities exposure (NetSecaaS)". This interaction is characterised by two primary flows: configuration/trigger of security actions (orange) and retrieval of security data (green).

Within the "ROBUST-6G Scope," CTID03 communicates with the "ROBUST-6G Security Capability" involved in the exposition, to trigger or retrieve security actions and data. Concurrently, CTID03 establishes a connection to the internal data management platform, which consists of CTID01 ("Data Fabric") and CTID02 ("Data Governance"). These components interact closely to oversee core data governance and policy enforcement functions. The transmission of data between these components facilitates the effective functioning of the overall process.

The architectural design under consideration here highlights the seamless exposure and management of these security services through CAMARA-style APIs. This is a process which validates the efficiency and abstraction goals intrinsic to the Open Gateway vision.

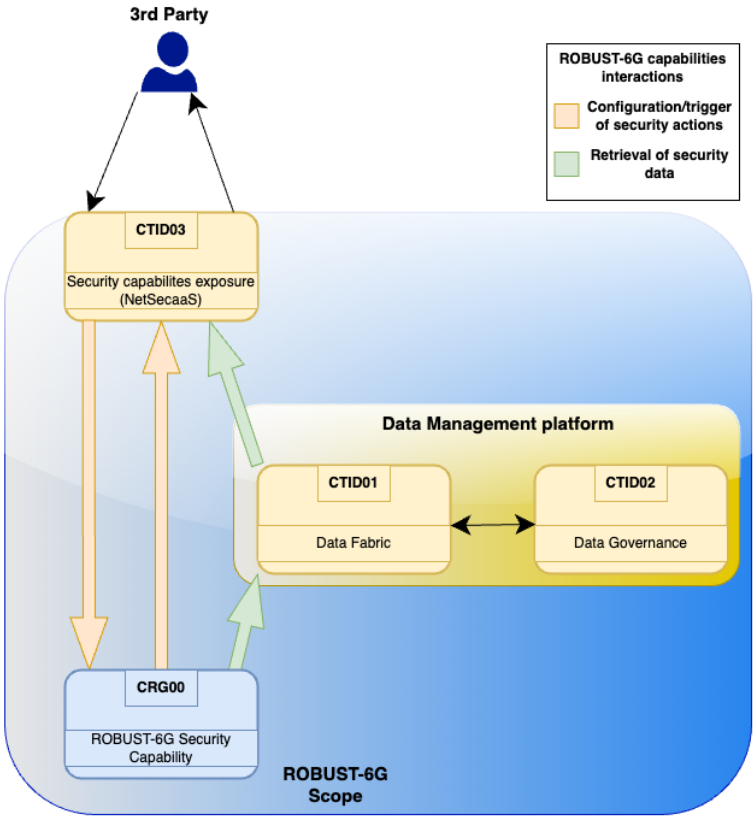

Figure 5-2: General Security Capability exposure scenario

The Table 5-1 outlines the security capabilities currently identified for exposure of the ROBUST-6G system. This table is the current latest version, but it will be refined in next phases, especially referring to the capabilities that are under development. In fact, capabilities highlighted in green are already under development, or with results partially available. These represent tangible functionality that can be validated and potentially integrated with other components. Of these, capability 4 (i.e. the SSLA simplified API) represents an initial approach to exposing all components that can be orchestrated by the Zero-Touch Security Management Layer. This means that, in future phases, this capability can be divided into more multiple specific capabilities.

Capabilities marked with orange refer to features assessed as feasible and valuable for the broader security ecosystem. However, these have not yet entered the development phase, and current development status of the system cannot ensure the technical development of them. As consequence, details regarding their design and delivery timeline remain to be defined. The purpose of this distinction is to provide transparency regarding current accessibility and planned future inclusions.

The table's clear delineation of each capability makes it a valuable resource for road mapping innovation and coordinating the broader effort towards robust, scalable NetSecaaS.

Table 5-1 Security capabilities available

N.	Capability Name	Description	Underlying Component IDs
1	Data Access Governance API	Exposes secure data cataloguing, policies and audit trails tracking of the data owner.	CTID02

2	Security Capabilities Discovery API	List and describe available security services and their status for third-party applications	CTID02, CTID01
3	XAI Analytics Data API	Expose historical explainability reports for AI/ML-driven security events and decisions	CUCD03, CTID02, CEBY03, CTID01
4	SSLA simplified API	This endpoint simplifies the process of defining Security Service Level Agreements (SLAs) by allowing users to specify only a subset of security capabilities and desired security levels, without having to define the entire SLA from scratch.	CTHL01
5	PHY Anomaly Detection API	Provides trustworthy sensing for radar localization integrity, while enhancing threat detection through early, cross-layer anomaly identification.	CENS03, CUPD05

Figure 5-3 shows the mapping of the UC3 components to highlight their alignment with the current version of the main ROBUST-6G architecture. The components are as follows:

1. Exposure Framework (CTID03): as the main gateway, it securely exposes network security features to third-party applications via user-friendly APIs. It manages interaction between external users and internal security services, enforces robust security standards, and controls access to network resources. The Exposure Framework simplifies complex security tasks, enabling users to configure or access security data with ease, regardless of their technical knowledge.
2. Data Fabric (CTID01): it is responsible for the seamless administration of data flows between the ROBUST-6G system and the exposure framework. This system facilitates the real-time exposition of security data and streamlines workflow execution. As the fundamental element for integrated data management, it ensures the consistent operation of all exposed security services.
3. Data Security (CTID02): it is designed to protect data access for data owners by enforcing policies and tracking audit trails. It also manages access governance, playing a crucial dual role in ensuring data exposure and service accessibility for third-party applications.
4. XAI-services (CUCD03, CEBY03): XAI-services provide access to explainable artificial intelligence analytics, including historical and on-demand reports for AI/ML-driven security events and decisions. These services enhance the transparency and auditability of security actions by offering underlying analytics and explainability features. These features can be integrated with other components to support end-to-end explainability and trust for third-party stakeholders.
5. Security Orchestration (CTHL01): This component addresses the core features responsible for SSLA ingestion from the 6G exposure framework, SSLA translation to security policies and security KPIs, and finally develop a dynamic and adaptative preparation and response workflows for security closed loops.

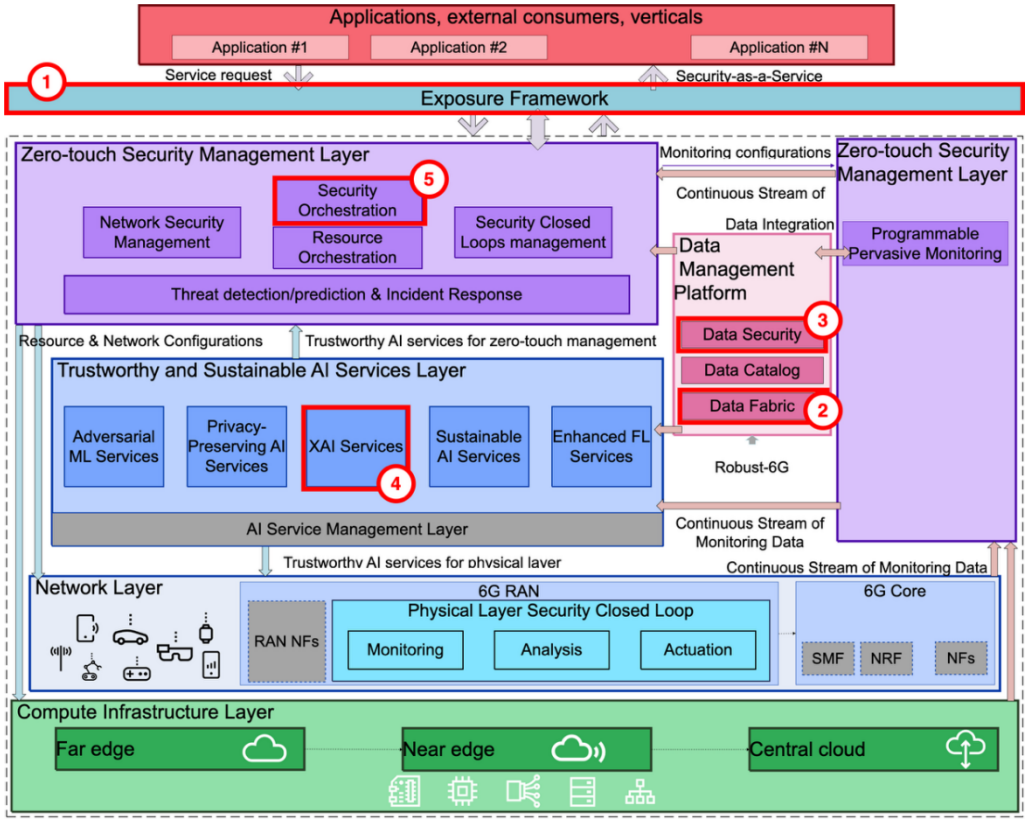

Figure 5-3: UC3 architecture mapping

The components described here are involved in different ways and to different extents, in order to facilitate the exposition of network capabilities. In order to assess how these components work together to deliver the functionalities targeted by UC3, several workflows have been defined to describe these behaviours. The following flows were analysed:

- Flow UC3_01 **"Access Governance data exposure"**: This flow enables secure data cataloguing, policy enforcement, and audit trail tracking for data owners when using third-party applications. It guarantees that external users can access data in a regulated manner via APIs, thereby supporting data security and transparency for all interactions defined by the access governance framework.
- Flow UC3_02 **"Security Capabilities discovery data exposure**": This flow enables third-party applications to access the list and current status of available security services. The API allows external users to discover and understand the exposed security capabilities, while maintaining robust security standards.
- Flow UC3_03 **"XAI Analytics Data":** This flow provides external users with historical and featured explainability reports for AI or ML-driven security events and decisions. The exposure makes analytics data accessible via dedicated APIs, contributes to greater transparency, and allows for the validation and auditing of security actions carried out by AI/ML models within the system.
- Flow UC3_04 **"Simplified SSLA enforcement":** This flow provides external users with a simplified interface to instantiate security function through the Security Orchestrator. The process abstracts the process as translation between high level requirements from user to SSLA, to be enforced by the orchestrator.

The following paragraphs provide a comprehensive overview of the flows, including all the necessary steps.

UC3_01 "Access Governance data exposure"

The process (shown in Figure 5-4) is initiated when a third-party user submits a request to access security-relevant data using a user-friendly API exposed by the Security Capabilities Exposure (CTID03). This API serves as the central interface, providing a clear definition and management of external requests. Upon receipt

of a request, the Exposure Framework immediately applies initial security rule checks, such as identity verification and basic compliance requirements, to filter out unauthorised or malformed requests at an early stage. These checks are supported by the Data Governance component (CTID02).

If authorisation is granted, the Exposure Framework transforms the high-level user request into the appropriate queries that the internal system can process. These queries are then directed to Data Governance itself, ensuring that only the authorised subset of information is extracted and delivered according to the user's permissions.

In this instance, the data are referred to the Data Security module, which is responsible for conducting detailed analyses of access requests and cross-referencing them with internal security policies and user privileges. The Data Security module is responsible for authorising or rejecting requests, as well as for securely cataloguing interactions and maintaining audit logs for compliance and traceability. Subsequently, the data is securely delivered back to the third-party user via the same API. In the event of a denial of the request, the Exposure Framework will respond with an unambiguous access-denied message to the third party, ensuring that no data is exposed.

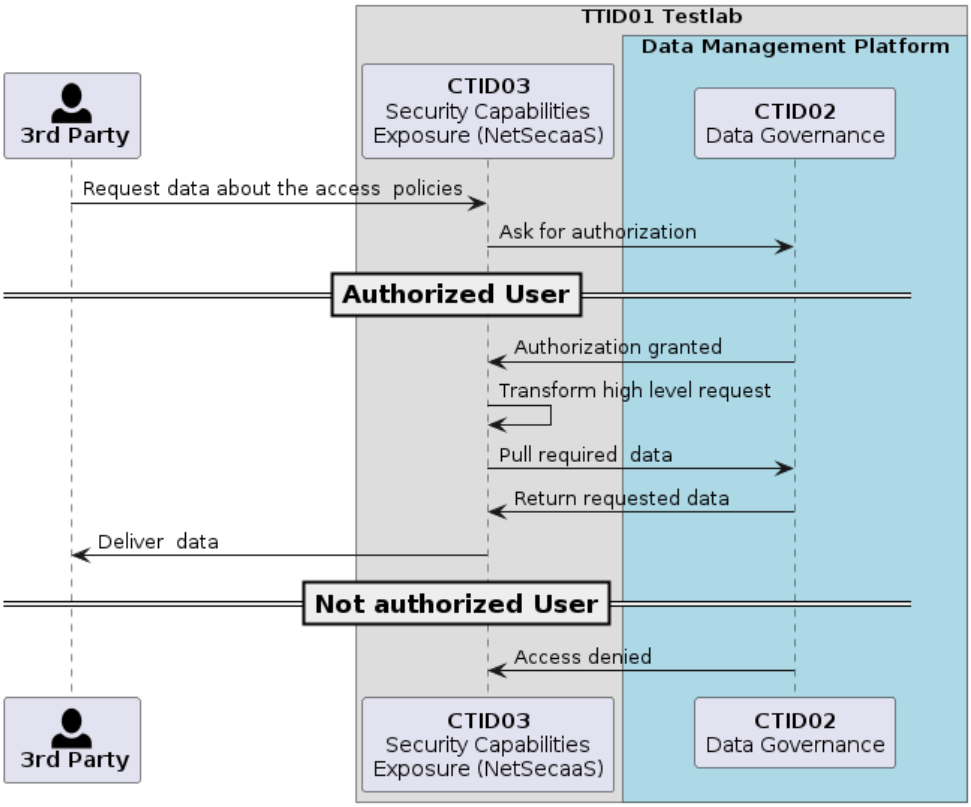

Figure 5-4: Access Governance data exposure flow

UC3_02 "Security Capabilities discovery data exposure"

In the workflow of Figure 5-5, a third-party entity interacts with the Security Capabilities Exposure module (CTID03) to discover available security capabilities. This initial request retrieves information about the security features or services that the platform can provide.

Upon receiving the request, the system first verifies the user's authorisation status, cross-checking with the Data Governance module (CTID02), which acts as the policy enforcement point for all access attempts. CTID02 assesses whether the requesting party meets the necessary criteria to proceed based on the security policies and access rules defined for the platform.

If authorisation is granted, the request is processed further by being translated into a more granular or system-specific query. The Data Fabric component (CTID01) then searches, compiles, and retrieves the required

dataset about available security capabilities from the platform's managed resources. The requested information is then compiled and passed back to the third party. The third-party user then receives detailed data about the platform's exposed security capabilities to support their operational or integration needs.

Conversely, if authorisation is denied at any stage, access is immediately denied, and the request is halted. No sensitive data is released, and the third party is simply informed that their access attempt has been denied.

This process ensures that exposure of critical security capability information is tightly governed and aligned with organisational access policies, providing comprehensive security capability discovery while maintaining robust control over who can access this sensitive information.

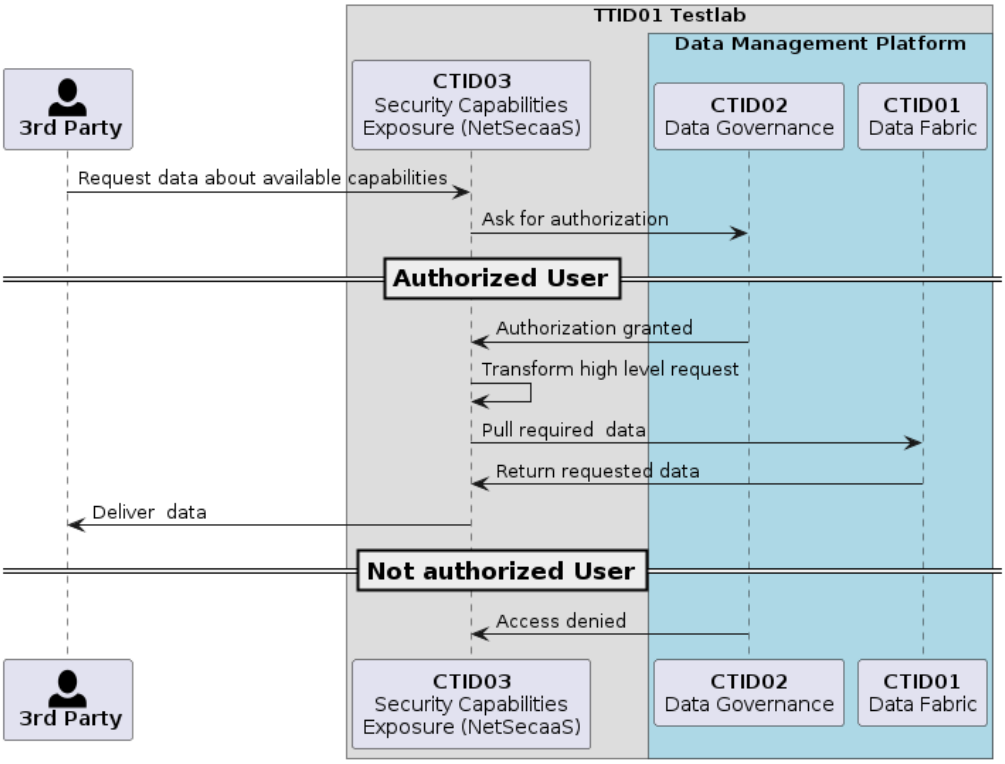

Figure 5-5: Security Capabilities discovery data exposure flow

UC3_03 "XAI Analytics Data"

As illustrated in Figure 5-6 the preliminary offline phase involves the use of AI analytics and Explainable AI (XAI) services, such as components like CUCD03 and CEBY03, to process available security data. These AI-based modules analyse raw security information to identify patterns, detect threats, and highlight potential areas of interest, while also generating explainability artefacts that clarify the rationale behind each decision or detection outcome. The results of these analyses, including both the processed insights and their associated explainability data, are securely stored in the Data Fabric component to ensure they are ready for use in subsequent interactions.

Following this preparatory phase, a third-party user then triggers the subsequent step by submitting a data request. This request typically specifies the type of attack of interest or delineates a particular time window for which security information is sought. Such requests are processed through the Exposure Framework, referenced as CTID03, which serves as a gatekeeper to the platform's sensitive data resources. Prior to the dissemination of any data, the Exposure Framework enforces a set of security rules and organisational policies, leveraging guidance from the Data Governance module, CTID02. This ensures that only authorised requests which comply with access policies are processed further.

Following the successful completion of the security and governance checks, the Data Fabric function (CTID01) is responsible for orchestrating the retrieval process. It manages and integrates the stored security data, drawing

from the pool of pre-analysed and explainable information generated by the AI/XAI modules. The relevant security dataset is then assembled and returned to the requesting third party via the same Exposure Framework channel, ensuring that both the insights and their explanations are readily available to inform the recipient's decisions or defence mechanisms. In the event of a failure in any of the authorisation steps, access will be denied, and no data will be transmitted, thereby maintaining the platform's stringent security posture throughout the process.

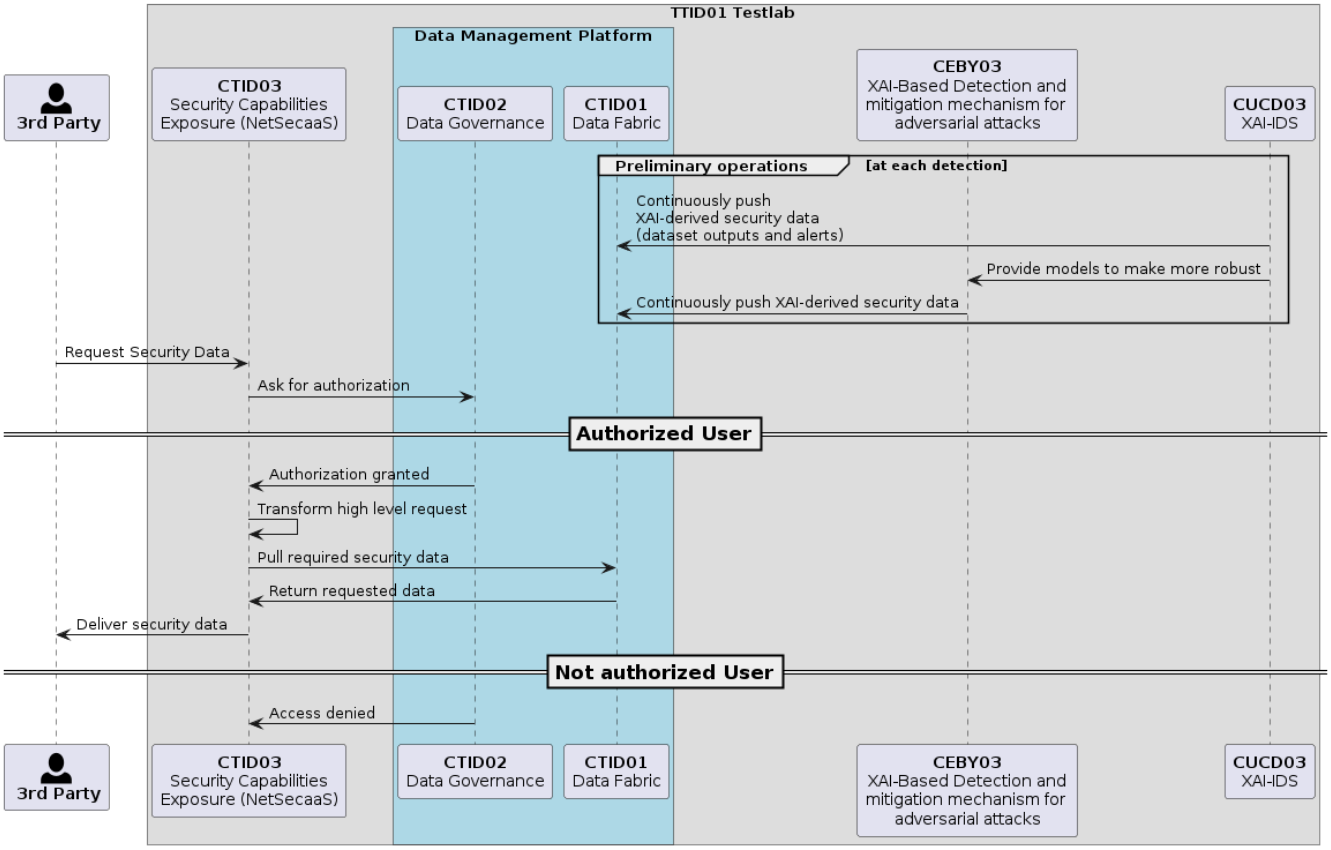

Figure 5-6: XAI Analytics Data flow

UC3_04 "Simplified SSLA enforcement"

As illustrated in Figure 5-7, the orchestration of security capabilities begins when a third-party entity submits a high-level request to interact with the ROBUST-6G platform. This request is first received by the Security Capabilities Exposure module, referenced as CTID03. It acts as the initial interface for external orchestration demands. Upon receipt, CTID03 forwards the request to the Data Governance component, CTID02, which is responsible for enforcing access control policies and verifying the requester's authorization status. If the requester is authorized, CTID02 grants access, unlock in the CTID03 the transformation of the high-level request into a structured Service Security Level Agreement (SSLA), encapsulating the specific parameters and constraints for the requested operation.

The generated SSLA is then forwarded from CTID03, to the Security Orchestration engine, CTHL01, for execution. CTHL01 interprets the SSLA and carries out the corresponding security actions within the platform. Once the orchestration is complete, the results are propagated back through CTID03 to the original requester, presented in a simplified and consumable format. In contrast, if the requester fails the authorization checks enforced by CTID02, the process is halted immediately, and access is denied. This structured flow ensures that all orchestration activities are tightly governed, traceable, and compliant with the platform's security and policy frameworks.

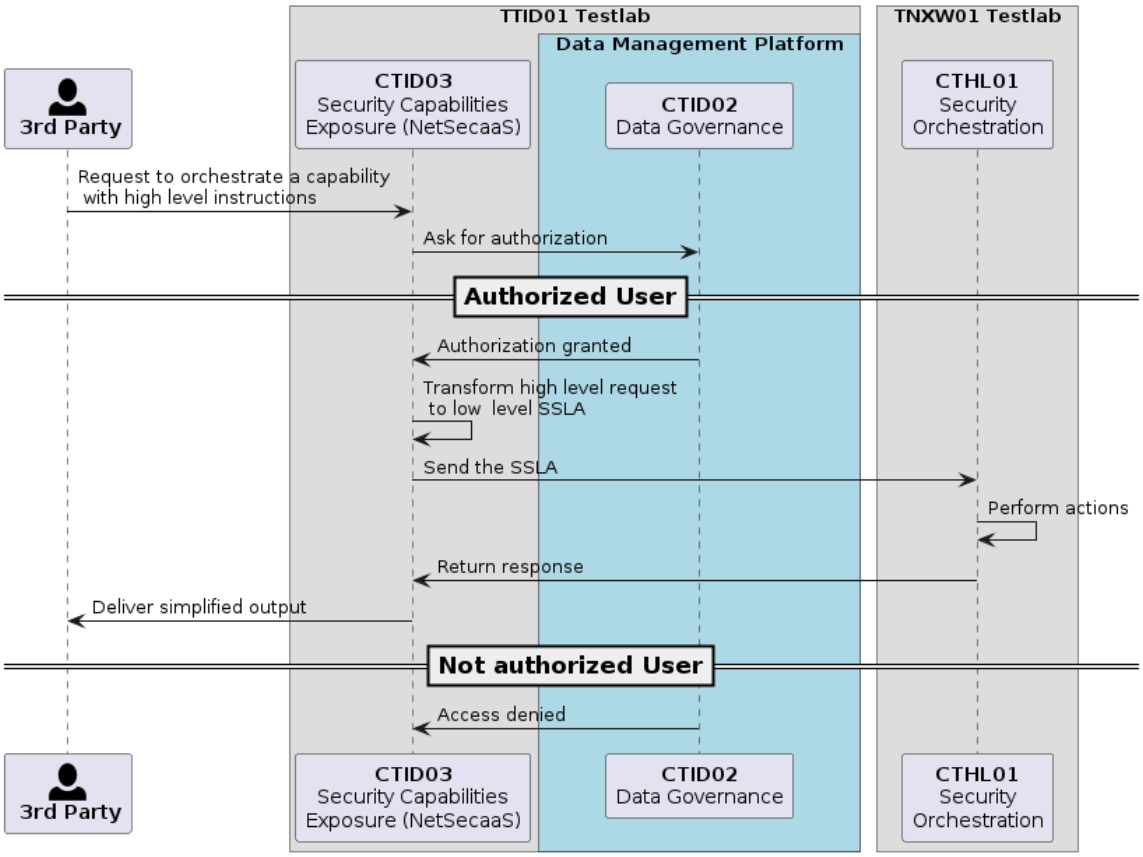

Figure 5-7: SSLA simplification process flow

5.1.2 Testbed Requirements and Deployment

The experimental validation of this scenario will be conducted mostly in TTID01, i.e., the TID testbed, as described in Figure 5-8, but with the involvement of TNXW01 testbed. Moreover, all dedicated modules are deployed within the TTID01 TID testbed. The Exposure Framework module (CTID03) handles API requests from external applications and facilitates access to the TID premises from outside. Authorized queries are then routed to the Data Fabric (CTID01) and Data Governance (CTID02) modules, as well as to the XAI Services modules (CUCD03 and CEBY03), which are also deployed in the TID testbed but are not directly accessible from outside the premises. The connection with TNXW01 will be established to communicate with security orchestrator (CTHL01).

Deployment Architecture

The components of this use case will be deployed using a container-based architecture to ensure portability, scalability, and ease of management. The Data Fabric (CTID01) and Data Governance (CTID02) modules will be deployed as services within a Kubernetes cluster running on TTID01. Meanwhile, the Exposure Framework (CTID03) and XAI Services (CUCD03 and CEBY03) modules will be instantiated as standalone Docker containers, TTID01 testbed as well. The deployment of Security orchestrator (CTHL01) was already discussed in section 4.1.2.

Current Status and Future Work

The initial implementation of the Exposure Framework (CTID03), Data Fabric (CTID01), and Data Governance (CTID02) modules has been completed. The next steps in the operational setup depend on the outputs provided by the other components to be exposed. These outputs will enable the creation of the data models essential for data storage, followed by policy definition and assignment, and ultimately the mapping of APIs to the correct endpoints of the exposed capabilities or Data Fabric data endpoints.

The first flow (UC3_01) has been successfully deployed and tested within the TID testbed using dummy policy configurations but still is pending of KPIs validations. The second flow (UC3_02) is still under development, as it requires the complete list of capabilities in order to expose a fully featured capabilities discovery API. Regarding the fourth flow (UC3_04), it is still in the design phase because the parsing and mapping of the high-level requirements to the underlying SSLA need to be developed using the final versions of the SSLA configurations. In addition, as mentioned in Section 5.1.2, this capability abstracts the exposition of more granular capabilities that depend on different versions of the SSLA.

Current development efforts are focused on UC3_03, specifically on implementing the mapping between the data exposed by the XAI Services (CUCD03 and CEBY03) and the Data Fabric (CTID01). Upcoming milestones include the development of API mappings to integrate CTID03 with the other capabilities listed in Table 5-1, namely CTHL01.

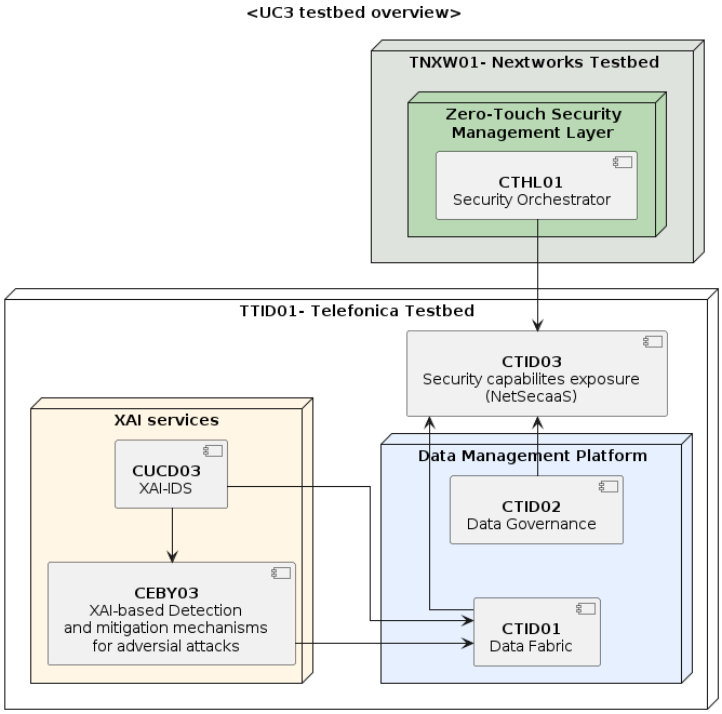

Figure 5-8:UC3 testbed planification

5.1.3 KPIs and Validation Criteria

The success of **Use Case 3** is measured using the following KPIs:

- **API Latency**:
  Average API response latency: ≤ 300ms.
  Maximum API response latency: ≤ 1 second for external applications.
- **API Resource Efficiency**:
  API CPU usage: ≤ 30%, ensuring efficient handling of API calls.
- **Security Capabilities Exposure**:
  At least 50% of ROBUST-6G's security capabilities are exposed through standard CAMARA APIs.

At this stage, the precise numerical values associated with the first two KPIs and their corresponding evaluation criteria have not yet been determined, as discussions are still ongoing regarding their concrete technical implementation, methodological approach, and validation procedures. Nevertheless, the definition of such

quantitative measurements – or, if required, the preparation of an updated and refined version of the current assessment framework – remains a planned activity for the near term.

With regard to the third KPI, it is currently being monitored and has shown the feasibility to address the exposition of 50% of the capabilities of ROBUST-6G. However, it should be noted that this statistic is subject to variation, since the list of underlying capabilities, which is dependent on the development and integration of additional system components, is still in a state of evolution. Consequently, the final value may differ once all dependencies are fully implemented and stabilized.

It is anticipated that this series of developments will undergo a progressive advancement in the forthcoming months, with successive refinements being introduced as the technical details reach maturity. The consolidated results, incorporating updated KPI values and any requisite methodological clarifications, will be formally reported in the subsequent project deliverable.

Finally, the UC3 challenges reported in D6.1 are being monitored and addressed through various approaches, considering the current status. For Challenge 3 (Scalability of Data Governance), CTID02 in the exposition shows that it is feasible to mitigate this challenge, although no validation is planned at this stage. Regarding Challenge 4 (Explainability of Security Mechanisms), the UC3_03 flow illustrates the efforts made to address this issue and shows that a solution is feasible. For Challenge 5 (Seamless Integration of Security Capabilities), work exposing Table 5-1 capabilities indicates that a solution can be delivered in future phases. All APIs under development follow the CAMARA style and are designed to be compatible with Open Gateway standards for third-party use. Challenges 6 (compliance with security and performance KPIs) and 1 (high latency in API response) are closely linked to numerical validation, which is not yet fully available. However, current results provide positive indications for addressing these challenges. Lastly, Challenge 2 (Security Policy Mapping Complexity) was found to be more related to security than to the transformation function's mapping, making it a lower priority. Therefore, no specific approach is planned at this time.

5.1.4 Flow Progress Tracking

Table 5-2 shows the intermediate results for the four data flows relating to capability exposure and NetSecaaS. Integration efforts have commenced in the TTID01 testbed, with notable progress in governance data exposure and capabilities discovery. Core components are now integrated to approximately 70–90% completion. The XAI analytics data flow is at an earlier stage, with several dependencies still unresolved, including partner feedback and the definition of output structures and ontologies. Current discussions are focused on aligning component inputs and ensuring accurate mapping to access policies. Overall, the data flows are advancing steadily, but full validation will require further coordination and completion of the remaining integration tasks.

Table 5-2 : Use Case 3 Flow Progress Tracking Table

ID	Included Components		Integration %	PTA	Status
UC3_1	CTID03	CTID02	90%	TTID01	Refinements in progress
UC3_2	CTID03	CTID01	70%	TTID01	Integration is ongoing
	CTID03	CTID02	70%	TTID01	Integration is ongoing
UC3_3	CTID03	CTID01	75%	TTID01	Integration is ongoing
	CTID03	CTID02	50%	TTID01	Integration is ongoing
	CTID01	CEBY03	10%	TTID01	Interfaces under configuration
	CTID01	CUCD03	33%	TTID01	Partial integration ongoing

	CEBY03	CUCD03	25%	TTID01	Interfaces under configuration
UC3_4	CTID03	CTHL01	10%	TTID01	Interfaces under development
	CTID03	CTID02	70%	TTID01	Integration is ongoing

6 Conclusions and Next Steps

This deliverable has reported the intermediate validation progress results of the ROBUST-6G project, documenting the first operational application of the validation plan set out in D6.1. The flow-based methodology has been applied in practice across the three use cases, moving from isolated component checks towards the first stages of integrated validation and preparing the ground for scenario-level execution.

Achievements at this stage can be summarised as follows:

- **Validation framework in practice:** The multi-level strategy (from component to flow and from flow to scenario) has been applied consistently. Adopting flows as the central validation unit has strengthened traceability between components, architectural functions, and testbed assets.
- **Component and flow progress:** Several key components have been validated in isolation, confirming functional readiness. Selected flows have been instantiated and exercised, demonstrating initial interoperability and revealing integration dependencies that inform subsequent iterations.
- **Testbed Integration Initiated:** Work has started to interconnect Partner Testbed Assets (PTAs). Containerisation and secure networking practices have enabled the first integration steps, providing the technical basis for a federated environment to support cross-partner execution.

Taken together, these results indicate that the project has moved beyond planning into practical validation. The flows provide a clear mechanism to validate the capabilities; the initial interconnections across partner testbeds show that multi-domain integration is feasible; and the KPI framework establishes how results will be collected and assessed. This progress creates a traceable and auditable path to scenario-level runs without overstating maturity at this intermediate point.

There are some limitations observed during this stage, these are:

- **Scenario-level validation:** It is not yet completed; preparation activities are ongoing across the three use cases; full end-to-end executions are still pending.
- **Limited quantitative KPI results:** While definitions and measurement procedures are available, numerical evidence remains sparse because several flows are mid-integration.
- **Uneven integration maturity:** The degree of completeness varies by use case and flow, with some elements still under preparation or dependent on components and interconnections that are not yet available.

In summary, the methodology has proven sound and the technical groundwork for final validation is in place. The achievements recorded here demonstrate concrete progress towards integrated validation while the limitations identify the remaining work to reach scenario-level execution and comprehensive KPI assessment.

References

[36.888] 3GPP TR 36.888, "Study on provision of low-cost Machine-Type Communications (MTC) User Equipments (UEs) based on LTE (Release 12)", June 2013.

[FU98] G. D. Forney and G. Ungerboeck, "Modulation and coding for linear Gaussian channels", IEEE Transactions on Information Theory, vol. 44, no. 6, pp. 2384-2415, October 1998.

[Maz75] J. E. Mazo, "Faster-than-Nyquist signaling", Bell System Technical Journal, vol. 54, no. 8, pp. 1451-1462, October 1975.

[TAZ+13] A. Tzanakaki, M. P. Anastasopoulos, G. S. Zervas, B. R. Rofoee, R. Nejabati and D. Simeonidou, "Virtualization of heterogeneous wireless-optical network and IT infrastructures in support of cloud and mobile cloud services". IEEE Communications Magazine, vol. 51, no. 8, pp. 155-161, August 2013.

[ROB25-DFL] ROBUST-6G, "Demo – Distributed Federated Learning Framework", EUCNC 2025, displayed at the ROBUST-6G booth [Online]. Available: https://youtu.be/m60IAzbjYpQ

[ROB24-D22] ROBUST-6G, "Use Cases, Requirements, ROBUST-6G Initial Architecture and Initial ROBUST-6G Dataspace", ROBUST-6G, Project Deliverable D2.2, December 2024. [Online]. Available: https://robust-6g.eu

[ROB24-D41] ROBUST-6G, "Security Automation for 6G", ROBUST-6G, Project Deliverable D4.1, December 2024. [Online]. Available: https://robust-6g.eu

[ROB25-D61] ROBUST-6G, "Use Case Validation Plan and Testbed Design", ROBUST-6G, Project Deliverable D6.1, June 2025. [Online]. Available: https://robust-6g.eu

[ROB25-D51] ROBUST-6G, "Library of known PHYs Attacks and PLS Dataset", ROBUST-6G, Project Deliverable D5.1, December 2024. [Online]. Available: https://robust-6g.eu

[NYU] H. Poddar, S. Ju, D. Shakya, T. S. Rappaport, "A tutorial on NYUSIM: sub-terahertz and millimeter-wave channel simulator for 5G, 6G and Beyond", IEEE Communications Surveys & Tutorials, vol. 26, no. 2, pp. 824-857, 2024.

[OPG25] GSMA Open Gateway API Descriptions booth [Online]. Available: https://www.gsma.com/solutions-and-impact/gsma-open-gateway/gsma-open-gateway-api-descriptions/

[CAM25] CAMARA API overview [Online]. Available: https://camaraproject.org/api-overview/

[CSCN-2025] A. P. Mayya, Y. Richhariya, A. K. Boroujeni, S. Vorberg, M. Matthe, R. Vinz, L. Senigagliesi, K. Klamka and A. Chorti, "Context-aware secret key generation demonstrator based on physical layer security", IEEE Conference on Standards for Communications and Networking (CSCN), 15-17 September, Bologna Italy