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

Deliverable D6.3

Prototype Validation of ROBUST-6G Components

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:	30/06/2026	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:	D6.3
Document Title:	Prototype Validation of ROBUST-6G Components
Editor(s):	Cem Ayyıldız, Fatih Emre Yıldız
Authors:	Contributors and their organisations are listed below
Contractual Date of Delivery:	30/06/2026
Dissemination level:	PU[1]/SEN
Status:	Final
Version:	1.0
File Name:	ROBUST-6G D6.3_v1.0

Revision History

Revision	Date	Issued by	Description
0.1	23.02.2026	ROBUST-6G WP6	Initial Draft & ToC Created
0.2	21.04.2026	ROBUST-6G WP6	ToC Updated
0.3	25.05.2026	ROBUST-6G WP6	Chapter 1-2-3-4 Drafted.
0.4	18.06.2026	ROBUST-6G WP6	All Chapters Completed
0.5	22.06.2026	ROBUST-6G WP6	First Review Completed
0.6	26.06.2026	ROBUST-6G WP6	Second Review Completed
1.0	30.06.2026	ROBUST-6G WP6	Final Version

Abstract

This deliverable presents the final prototype validation of the ROBUST-6G platform. Building on the validation framework established in D6.1 and the intermediate results reported in D6.2, it covers the full scenario-level validation of five prototypes across three use cases. KPI attainment is reported per use case and scenario, and the project Global Objectives are verified at the integrated project level. Together, these elements constitute the definitive record of the ROBUST-6G consortium's integration and proof-of-concept activities.

Keywords

6G security, prototype validation, decentralised federated learning, trustworthy AI, physical layer security, zero-touch security management, Network-Security-as-a-Service, IoT threat detection, closed-loop security, KPI attainment, proof-of-concept, end-to-end validation, flow-based validation, security capabilities exposure, AI explainability, adversarial robustness

[1] SEN = Sensitive, only members of the consortium (including the Commission Services). Limited under the conditions of the Grant Agreement

PU = Public

List of Contributors

Participant	Short Name	Contributors
Ericsson Araştırma Geliştirme ve Bilişim Hizmetleri A.Ş	EBY	Betül Güvenç Paltun, Mustafa Riza Akdeniz
Telefónica Innovación Digital	TID	Riccardo Nicolicchia
Universidad de Murcia	UMU	Fernando Torres Vega, Alberto García Pérez, Enrique Tomás Martínez Beltrán, José María Jorquera Valero, Manuel Gil Pérez
Chalmers University of Technology	CHA	Tommy Svensson, Azadeh Tabeshnezhad, Mehdi Sattari, Masoom Rabbani
University College Dublin	UCD	Bartlomiej Siniarski, Madhusanka Liyanage, Chamara Sandeepa, Farah Abed Zadeh
University of Padova	UNIPD	Giovanni Perin, Michele Rossi, Stefano Tomasin
Nextworks	NXW	Marco Ruta, Pietro Giuseppe Giardina
ENSEA	ENSEA	Arsenia Chorti, Luan Chen, Sotiris Skaperas, Solomon Yese
Linkopings Universitet	LIU	Eunjeong Jeong, Nikolaos Pappas
EURECOM	EUR	Marios Kountouris, Ioannis Pitsiorlas
Thales Six Gts	THALES	Louis Cailliot
GOHM Elektronik ve Bilişim San. Tic. Ltd. Şti.	GOHM	Cem Ayyıldız, Fatih Emre Yıldız
Axon Logic	AXON	Wei Chuen Yau, Chih Yang Pee, Su Fong Chien, Charilaos Zarakovitis

List of Reviewers

Participant	Short Name	Contributors
Nextworks	NXW	Pietro Giuseppe Giardina
ENSEA	ENSEA	Laura Luzzi

Executive Summary

This deliverable presents the final validation of the ROBUST-6G platform, concluding the work carried out under Work Package 6. Building on the validation framework established in D6.1 and the intermediate results reported in D6.2, it covers the full scenario-level validation of five prototypes across three use cases: AI model trustworthiness evaluation in distributed 6G environments, automatic threat detection and mitigation in 6G-enabled IoT environments, and security capabilities exposure via Network-Security-as-a-Service.

The document presents the final state of the ROBUST-6G platform architecture, confirming the alignment of all active flows with the updated reference architecture and documenting the status of all components, flows, and validation scenarios across the partner testbed assets. The five prototypes that translate this architecture into demonstrable proof-of-concept systems are described in full detail, covering their objectives, position within the architecture, and composition. Prototype 1 implements a decentralised federated learning framework covering privacy-preserving distributed training, adversarial robustness, sustainability evaluation, and explainability artefact generation. Prototype 2 demonstrates automated zero-touch threat detection and closed-loop security management across IoT environments of increasing complexity, integrating programmable monitoring, AI-assisted incident response, and playbook-based enforcement. Prototype 3 exposes ROBUST-6G security capabilities to third-party applications through intent-based Representational State Transfer (REST) APIs (Application Programming Interfaces) aligned with the CAMARA Project, an open-source initiative under the GSMA Open Gateway framework that standardises telecom network API exposure. Prototype 4 addresses physical layer security through a closed-loop configuration covering jamming detection, physical layer authentication, and secret key generation. Prototype 5 serves as the Master Prototype, integrating the capabilities of all four prototypes into a unified demonstration spanning the full ROBUST-6G architecture.

The three use cases are described in their final form, covering the scenario objectives, functional flows, architectural positioning, and the prototype used for each scenario. Use Case 1 addresses AI model trustworthiness across two scenarios, covering decentralised federated learning for privacy-preserving model training, and physical- and sensing-layer trustworthiness and resilience. Use Case 2 covers automatic threat detection and mitigation across three scenarios of increasing complexity, demonstrating how the zero-touch security platform scales from a single closed-loop instance to multi-loop and multi-tenant configurations. Use Case 3 demonstrates how third-party applications can consume ROBUST-6G security capabilities through the NetSecaaS interface without requiring knowledge of the underlying orchestration.

Validation is conducted through the flow-based methodology established in D6.1. The consolidated results cover prototype-level functional validation, KPI attainment per use case and scenario, and Global Objective verification at the integrated project level. Full KPI and objective traceability is provided in the supporting appendices alongside the interface specifications and the dataset catalogue documenting the project-generated and partner-contributed datasets used throughout the validation campaign.

Table of Contents

1 Introduction 20
1.1 Objective of the Document 20
1.2 Structure of the Document 20
1.3 Terminology and Definitions 22
2 ROBUST-6G Platform Overview 23
2.1 System Architecture 23
2.2 Gap Analysis 26
2.3 Components 29
2.4 Flows 34
2.5 Scenarios 37
2.6 Prototypes 37
2.7 Unified Testbed Configuration 38
3 Prototypes 40
3.1 Prototype 1: Trustworthy AI 40
3.1.1 Overview and Demonstration Objectives 40
3.1.2 Position within the Architecture 41
3.1.3 Prototype Composition and Architecture 42
3.1.3.1 Architectural mapping and functional flows 42
3.1.3.2 Core Framework Components 43
3.1.3.3 Deep Dive: Global Model Repository (GMR) 44
3.1.3.4 Krum Integration for Byzantine Resilience 45
3.1.3.5 Integration and Modularity 45
3.2 Prototype 2: Multi-Layer Zero-Touch Defender 45
3.2.1 Overview and Demonstration Objectives 46
3.2.2 Position within the Architecture 47
3.2.3 Prototype Composition and Architecture 48
3.3 Prototype 3: NetSecaaS Gateway 49
3.3.1 Overview and Demonstration Objectives 49
3.3.2 Position within the Architecture 49
3.3.3 Prototype Composition and Architecture 51
3.4 Prototype 4: Physical and Sensing Layer Trustworthiness 54
3.4.1 Overview and Demonstration Objectives 54

3.4.2 Position within the Architecture 55
3.4.3 Prototype Composition and Architecture 56
3.5 Prototype 5: Master Prototype 58
3.5.1 Overview and Demonstration Objectives 58
3.5.2 Position within the Architecture 58
3.5.3 Prototype Composition and Architecture 59
3.5.4 Demonstration Storyline 60
4 Use Cases 62
4.1 AI model trustworthiness evaluation for 6G decentralised scenarios 62
4.1.1 Scenario 1: Decentralised federated learning for joint privacy-preserving ML/DL model training 63
4.1.1.1 Scenario Overview and Objectives 63
4.1.1.2 Position within the Architecture 64
4.1.1.3 Functional Flows Description 65
4.1.1.4 Prototype in Use 74
4.1.2 Scenario 2: Physical and Sensing Layer Trustworthiness and Resilience 74
4.1.2.1 Scenario Overview and Objectives 74
4.1.2.2 Position within the Architecture 75
4.1.2.3 Functional Flows Description 76
4.1.2.4 Prototype in Use 80
4.2 Use Case 2: Automatic threat detection and mitigation in 6G-enabled IoT environments 81
4.2.1 Scenario 1 – Device violation to cause an economic harm (a) 82
4.2.1.1 Scenario Overview and Objectives 82
4.2.1.2 Position within the Architecture 82
4.2.1.3 Functional Flows Description 82
4.2.1.4 Prototype in Use 85
4.2.2 Scenario 2 – Fraudulent usage of device resources 85
4.2.2.1 Scenario Overview and Objectives 85
4.2.2.2 Position within the Architecture 86
4.2.2.3 Functional Flows Description 86
4.2.2.4 Prototype in Use 87
4.2.3 Scenario 3 – Device violation to cause an economic harm (b) 87
4.2.3.1 Scenario Overview and Objectives 87

4.2.3.2 Position within the Architecture 88
4.2.3.3 Functional Flows Description 88
4.2.3.4 Prototype in Use 90
4.3 Use Case 3: Security Capabilities Exposure (NetSecaaS) 90
4.3.1 Scenario Overview and Objectives 90
4.3.2 Position within the Architecture 90
4.3.3 Functional Flows Description 91
4.3.4 Prototype in Use 93
5 Overall Validation Summary 95
5.1 Prototype Validation Assessment 95
5.1.1 Prototype 1: Trustworthy AI 95
5.1.1.1 Validation Setup 96
5.1.1.2 Validation Outcomes 98
5.1.2 Prototype 2: Multi-Layer Zero-Touch Defender 104
5.1.2.1 Validation Setup 104
5.1.2.2 Testbed Configuration 104
5.1.2.3 Validation Outcomes 106
5.1.3 Prototype 3: NetSecaaS Gateway 129
5.1.3.1 Validation Setup 130
5.1.3.2 Validation Outcomes 131
5.1.4 Prototype 4: Physical and Sensing Layer Trustworthiness 135
5.1.4.1 Validation Setup 136
5.1.4.2 Validation Outcomes 140
5.1.5 Prototype 5: Master Prototype 144
5.1.5.1 Testbed Configuration 144
5.1.5.2 Validation Outcomes 154
5.2 Use Case KPI Attainment 156
5.2.1 UC1 Scenario 1 KPI Attainment 157
5.2.1.1 Validation Setup 157
5.2.1.2 Validation Outcomes 158
5.2.2 UC1 Scenario 2 KPI Attainment 170
5.2.2.1 ENSEA: Physical Layer Security Closed Loop 170

5.2.2.2 GOHM: RF Fingerprinting based Rogue Transmitter Detection and Device Authentication 171
5.2.3 UC2 KPI Attainment 174
5.2.3.1 Validation Setup 175
5.2.3.2 Validation Methodology 175
5.2.3.3 Validation Outcomes 176
5.2.3.4 Consolidated Evaluation of WP4 Quantifiable Targets 182
5.2.4 UC3 KPI Attainment 183
5.2.4.1 Validation Setup 184
5.2.4.2 Validation Outcomes 186
5.3 Global Objective Verification 188
5.4 Overall Evaluation 189
5.4.1 Achievement against the DoA 189
5.4.2 What the Validation Demonstrates 189
5.4.3 Value and Reusability for Other Projects 190
6 Conclusions 191
6.1 Summary of Outcomes 191
6.2 Lessons Learned 191
6.3 Project Contributions 192

List of Tables

Table 2.1: Gap Analysis Status 27
Table 2.2 Status overview of ROBUST-6G components 29
Table 2.3: Functional Flows Status - Summary 34
Table 2.4: UC1.1 Flows Status 35
Table 2.5: UC1.2 Flows Status 36
Table 2.6: UC2.1 Flows Status 36
Table 2.7: UC2.2 Flows Status 36
Table 2.8: UC2.3 Flows Status 36
Table 2.9: UC3 Flows Status 36
Table 5.1: PHY Demonstrator capabilities and accessible methods 144
Table 5.2: Best three DFL TON-IoT threat-detection models retrieved from the GMR 150
Table 5.3 Use Case KPI attainment status 156
Table 5.4 Measured Metrics for UC1.1 TEST01 159
Table 5.5 Measured Metrics for UC1.1 TEST02 161
Table 5.6 Measured Metrics for UC1.1 TEST03.1 163
Table 5.7 Measured Metrics for UC1.1 TEST03.2 165
Table 5.8 Measured metrics for UC1.1 TEST04 167
Table 5.9 Measured metrics for UC1.1 TEST05 169
Table 5.10 Measured metrics for RFFI TEST01 - Rogue Transmitter Detection (KPI6) 173
Table 5.11 Measured metrics for RFFI TEST02 - Device Authentication Accuracy and Resilience Improvement (KPI7) 174
Table 5.12: UC2 KPIs and Target Values 175
Table 5.13: UC2 Validation Tests and KPI Mapping 175
Table 5.14: Numerical results for TEST-UC2-01 176
Table 5.15: SNORT Alert Generation Time Summary 177
Table 5.16: AI Based Threat Detection Summary 178
Table 5.17: Mitigation Accuracy and Velocity (S-CLs Executions) 179
Table 5.18: Service Instantiation and Teardown - Numerical Results 179
Table 5.19 Timing Statistics 180
Table 5.20 Per-run times 180
Table 5.21: Consolidated Evaluation of WP4 Quantifiable Targets 182

Table 5.22 Global Objective Verification 188
Table 5.23 Validation Status against DoA 189

List of Figures

Figure 1.1: D6.3 Chapter Flow and Reading Guide 21
Figure 2.1: ROBUST-6G Architecture 24
Figure 2.2 Unified ROBUST-6G testbed and Partner Testbed Assets 40
Figure 3.1 Architecture mapping of UC1_1 42
Figure 3.2 Architectural diagram of the components of Prototype 1 42
Figure 3.3: ROBUST-6G Zero-Touch Security Platform 46
Figure 3.4: Prototype 2 Positioning in the ROBUST-6G Architecture 47
Figure 3.5: Prototype 2 High Level Architecture 48
Figure 3.6 Generalised data retrieval flow Prototype 3 50
Figure 3.7: Generalised configuration and trigger flow Prototype 3 51
Figure 3.8:NetSecaaS Gateway — general architecture and integration with the ROBUST-6G platform 52
Figure 3.9:Generic data ingestion pipelines feeding the Data Fabric of the NetSecaaS Gateway 53
Figure 3.10 Architectural positioning of Prototype 4 within the Physical Layer Security Closed Loop 55
Figure 3.11 Prototype 4 Demonstrator architecture 56
Figure 3.12 Prototype 4 in operation mode 56
Figure 3.13 Prototype 5 ROBUST-6G architecture coverage 59
Figure 3.14 Prototype 5 Composition 60
Figure 3.15 Prototype 5 – GMR Models onboarding on the ZTSP 60
Figure 3.16 Prototype 5 – GMR Model Orchestration as Security Function 61
Figure 3.17 Prototype 5 – PHY SF and related OpenC2 Actuator Upload on the ZTSP 61
Figure 3.18: Prototype 5 - PHY S-CL with PHY OpenC2 Actuator 61
Figure 3.19: Prototype 5 - Exposure of Security Capability thought the NetSecaaS 62
Figure 4.1 Flow UC1_1_01 – Privacy and decentralization flow diagram, benign nodes 66
Figure 4.2 Flow UC1_1_02 – Sequence diagram for evaluating DFL system's robustness under attack 68
Figure 4.3 Flow UC1_1_03 – Sustainable and efficient evaluation of the model training lifecycle 69
Figure 4.4 Flow UC1_1_04 – Continuous monitoring of explainability (SHAP/t-SNE) in federated training 71
Figure 4.5 Flow UC1_1_05 – Privacy-enhanced collaborative model training flow diagram via secure aggregation 73
Figure 4.6 Flow UC1_2_01 – PHY layer trstworthiness evaluation 76
Figure 4.7 Flow UC1_2_02 – Mutual authentication 78
Figure 4.8 Flow UC1_2_03 – (Fast) Secret key generation 79

Figure 4.9 UC2.1 Proactive Security Enforcement Flow 83
Figure 4.10 UC 2.1 Threat Detection combining Network and IoT Data 84
Figure 4.11 UC2.1 Threat Mitigation via reactive plan execution 85
Figure 4.12 UC2.2 Investigative Loop 86
Figure 4.13 UC2.2 Resolutive Loop 87
Figure 4.14: UC2.3 Loops and Coordination deployment 89
Figure 4.15: UC2.3 Loops execution and conflict resolution 89
Figure 4.16:UC3_01 — Security Capabilities discovery data exposure 91
Figure 4.17: UC3_02 — XAI Analytics Data 92
Figure 4.18: UC3_03- Simplified SSLA enforcement 93
Figure 4.19 Deployment of Prototype 3 across the federated testbeds for UC3 94
Figure 5.1 DFL Framework Frontend and Attack configuration 99
Figure 5.2 DFL Framework Frontend Active Training Progress Metrics 100
Figure 5.3 DFL Framework Frontend Active Training Progress Metrics 100
Figure 5.4 DFL Framework Frontend Active Training Progress Metrics 101
Figure 5.5 SHATs Feature Importance Plot in the DFL Framework 102
Figure 5.6 DDoS Anomaly Explanation Participant 0 round 9 102
Figure 5.7 GMR API dashboard 103
Figure 5.8 GMR Postgress Database - Table scenarios 103
Figure 5.9 GMR Postgress Database - Table participants 104
Figure 5.10: UC2 Scenario 1 and 2 - testbed configuration 105
Figure 5.11: UC2 Scenario 3 - testbed configuration 105
Figure 5.12: ZTSO - TBOpenC2 Actuator Knowledge Graph 106
Figure 5.13: ZTSO - Catalogue of Security Functions 107
Figure 5.14: ZTSO - Infrastructure Environments 107
Figure 5.15: S-RO - Onboarded Environments 108
Figure 5.16: S-RO - Onboarded SFs and S-CL Functions Artefacts 108
Figure 5.17: S-CL Mgmt - Onboarded S-CLF Descriptors 108
Figure 5.18: S-CL Mgmt - Onboarded Rule-based S-CL Descriptor 109
Figure 5.19: ThingBoard User Dashboard 109
Figure 5.20: ThingBoard default Security Settings 110
Figure 5.21: ZTSO SCM Plan submission and context retrieval 110
Figure 5.22: GenAI Gateway IRP Generation 111

Figure 5.23: Use Case 2 Scenario 1 IRP 111
Figure 5.24: ZTSO SCM - Instantiated Security Service 111
Figure 5.25: Instantiated Security Service on the target Infrastructure 112
Figure 5.26: UC2 S1 - PMP Configuration 112
Figure 5.27: Scenario 1 - Attack from the Dashboard 113
Figure 5.28: UC2 Scenario 1 - S-CL Analysis execution 113
Figure 5.29: UC2 Scenario 1 - S-CL Decision execution 113
Figure 5.30: UC2 Scenario 1 - S-CL Execution execution 113
Figure 5.31: UC2 Scenario 1 - Device unassigned from the user 114
Figure 5.32: UC2 Scenario 1 - Improved Password Policy 114
Figure 5.33: UC2 Scenario 1 - User Account re-activation e-mail 115
Figure 5.34: UC2 Scenario 1 - New password policy enforcement 115
Figure 5.35: UC2 Scenario 2 - ThingBoard Dashboard 116
Figure 5.36: OTA Package on ThingBoard 116
Figure 5.37: Investigative S-CL Descriptor 116
Figure 5.38: Resolutive S-CL Descriptor 117
Figure 5.39: Investigative Service Instantiated 117
Figure 5.40: IoT Alert from ThingBoard Dashboard - power spike 118
Figure 5.41: Investigative S-CL Analysis - IoT Alert detected 118
Figure 5.42: Investigative S-CL Decision - Security Service Update Request 118
Figure 5.43: Resolutive Security Service Started 119
Figure 5.44: Resoultive Service SFs and S-CL Stages Deployed 119
Figure 5.45: Resolutive S-CL Analysis Logs - Flow retrieved and passed to AI Algorithm 119
Figure 5.46: Resolutive S-CL Decision Logs - Playbook selected and validated 120
Figure 5.47: UC2 Scenario 2 Cryptomining Playbook 120
Figure 5.48: Resolutive S-CL Execution Logs - Playbook executed 120
Figure 5.49: TB OpenC2 Consumer Logs - Playbook steps execution 121
Figure 5.50: Smart Farm Platform - S-RO View 121
Figure 5.51: SHORT Farm S-CL Descriptor 122
Figure 5.52: LONG Farm S-CL Descriptor 122
Figure 5.53: Smart Farms initial Infrastructure deployment 122
Figure 5.54: Instantiated SHORT Loops - No Coordination 123
Figure 5.55: SHORT Loop Monitoring and Analysis - No Attack 123

Figure 5.56: SHORT Loop Decision and Execution - No Attack 124
Figure 5.57: SHORT Loop Monitoring and Analysis - Attack 124
Figure 5.58: SHORT Loop Decision and Execution - Attack 125
Figure 5.59: Instantiated Short Loops and Long Loop 125
Figure 5.60: Farm 1 Decision - Attack Mitigated by Coordination 126
Figure 5.61: Master Loop Analysis - Inconsistency detection 126
Figure 5.62: Master Loop Decision - Local Farm decision ovveride 126
Figure 5.63: Farm data ontology 127
Figure 5.64: Data Fabric RML mapping storage snapshot 128
Figure 5.65: Data fabric Dagster data lifting pipelines 129
Figure 5.66: Farm data in the Data Fabric GraphDB 129
Figure 5.67:Validation setup for Prototype 3 130
Figure 5.68: NetSecaas NBI API UI 132
Figure 5.69:NetSecaaS API Data endpoints 133
Figure 5.70:NetSecaaS API Functional endpoints 133
Figure 5.71: NetSecaaS AuthN/AuthZ endpoint 134
Figure 5.72:NetSecaaS API Data Endpoint response 134
Figure 5.73:NetSecaaS API Functional endpoint response 135
Figure 5.74: Jamming detection and localisation output over the 24 × 24 spatial grid 141
Figure 5.75: Detection decision and estimated jammer position under a medium-SNR setting 141
Figure 5.76: AoA-based spoof detection 142
Figure 5.77: Secret key generation output 142
Figure 5.78: reconciliation error versus reconciliation code rate 143
Figure 5.79: PHY Demonstrator modelling in the KG 145
Figure 5.80: PHY Demonstrator Environment in the Catalogue 145
Figure 5.81 - PHY Demonstrator Security Functions in the Catalogue 146
Figure 5.82: PHY S-CL Descriptor 147
Figure 5.83: PHY Layer Security Service – PHY S-CL and PHY OpenC2 Acutator 148
Figure 5.84: SHY SSe Instantiated 148
Figure 5.85: SFs and S-CL Stages deployed 148
Figure 5.86: S-CL Monitoring Stage Logs 149
Figure 5.87: S-CL Analysis Stage Logs 149
Figure 5.88: S-CL Decision Stage Logs 149

Figure 5.89: S-CL Execution Stage Logs 149
Figure 5.90: OpenC2 Phyisical Layer consumer logs 149
Figure 5.91: GMR OpenAPI Specification 150
Figure 5.92: DFL AI Algorithms modelling in the KG 151
Figure 5.93: DFL Algorithm artefact in the S-RO 151
Figure 5.94: DFL AI Algorithms in the catalogue - Best 3 152
Figure 5.95: NetSecaaS endpoint for UC2.1 SSLA generation 154
Figure 5.96:NetSecaaS endpoint for UC2.1 SSLA generation response 154
Figure 5.97: F1-score convergence across 10 federated rounds with TON-IoT/CyberNet. 159
Figure 5.98: Robustness assessment by comparing poisoned FedAvg and Krum aggregation 161
Figure 5.99: Inference energy consumption for a traditional ANN when model quantization in different fashions is applied. 164
Figure 5.100: Inference energy comsumption: comparison between an SNN model and an equivalent standard ANN for the benchmark FedAvg and the two proposed aggregation methods (traditional hardware). 164
Figure 5.101 Global test accuracy with respect to cumulated energy consumption 166
Figure 5.102: Auditable evidence by ShaTS supporting CyberNet trustworthiness assessment. 167
Figure 5.103 Latency with respect to number of clients 169
Figure 5.104 Spoofing scenario as visualised through the RF fingerprinting monitoring dashboard 172
Figure 5.105 GOHM testbed configuration and high-level data flow from dataset generation to edge-based inference 173
Figure 5.106: SNORT Alert Generation Time BoxPlot 177
Figure 5.107: PMP Flow Generation BoxPlot 178
Figure 5.108: AI Inference Latency BoxPlot 178
Figure 5.109: Security Service Teardown and Deployment BoxPlot 179
Figure 5.110 CACAO playbook generation overall timing statistics 181
Figure 5.111 CACAO playbook generation per-run timing 181
Figure 5.112:Validation setup for UC3 184
Figure 5.113: UC3 latency validation results 187
Figure 5.114:Local CPU KPI report for UC3 187

Acronyms and abbreviations

Term	Description
3GPP	3rd Generation Partnership Project
ADMM	Alternating Direction Method of Multipliers
AES	Advanced Encryption Standard
AI	Artificial Intelligence
AKA	Authentication and Key Agreement
ANN	Artificial Neural Network
AoA	Angle-of-Arrival
API	Application Programming Interface
AuthN/AuthZ	Authentication / Authorization
CACAO	Collaborative Automated Course of Action Operations
CDL	Clustered Delay Line
CME	Conditional Min-Entropy
CNC	Computer Numerical Control
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSI	Channel State Information
CSV	Comma-Separated Values
DDoS	Distributed Denial-of-Service
DFL	Decentralized Federated Learning
DL	Deep Learning
DoA	Description of Action
DOI	Digital Object Identifier
DoS	Denial-of-Service
DRS	Douglas-Rachford Splitting
DX.Y	Deliverable X.Y
E2E	End-to-End
eBPF	extended Berkeley Packet Filter
F-BLEAU	Fast Blackbox Leakage Estimation Algorithm Using Machine Learning
FedAvg	Federated Averaging
FL	Federated Learning
GCM	Galois/Counter Mode for Symmetric Block Ciphers
GenAI	Generative Artificial Intelligence
GLRT	Generalized Likelihood Ratio Test
GMR	Global Model Repository
GPU	Graphics Processing Unit
GSMA	GSM Association
GUI	Graphical User Interface

HE	Homomorphic Encryption
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HVAC	Heating, Ventilation, and Air Conditioning
I/Q	In-phase/Quadrature
IDS	Intrusion Detection System
IID	Independent and Identically Distributed
IoT	Internet of Things
IP	Internet Protocol
IRP	Incident Response Playbook
ISAC	Integrated Sensing and Communication
JSON	JavaScript Object Notation
KG	Knowledge Graph
KPI	Key Performance Indicator
LCM	Lifecycle Management
LLM	Large Language Model
LLR	Log-Likelihood Ratio
LoS	Line-of-Sight
LRP	Layer-wise Relevance Propagation
LSTM	Long Short-Term Memory
MaMIMO	Massive Multiple-Input Multiple-Output
MIMO	Multiple-Input Multiple-Output
MITM	Man In the Middle
ML	Machine Learning
MLP	Multi-Layer Perceptron
MQTT	Message Queuing Telemetry Transport
MUSIC	Multiple Signal Classification
NBI	Northbound Interface
NetSecaaS	Network-Security-as-a-Service
NIST	National Institute of Standards and Technology
NLoS	Non-Line-of-Sight
NN	Neural Network
NOMA	Non Orthogonal Multiple Access
OAuth	Open Authorization
OFDM	Orthogonal Frequency Division Multiplexing
OpenC2	Open Command and Control
O-RAN	Open Radio Access Network
OTA	Over-The-Air
P2P	Peer-to-Peer
PHY	Physical (layer)
PID	Process ID

PLA	Physical Layer Authentication
PLCL	Physical Layer Security Closed Loop
PLS	Physical Layer Security
PMP	Programmable Monitoring Platform
PNG	Portable Network Graphics
PoC	Proof-of-Concept
PU	Public
QoS	Quality of Service
QT	Quantifiable Targets
RAM	Random-Access Memory
RAN	Radio Access Network
RDF	Resource Description Framework
REST	Representational State Transfer
RF	Radio Frequency
RFFI	Radio Frequency Fingerprinting Identification
RIS	Reconfigurable Intelligent Surface
RML	Relational Markup Language
RSA	Rivest-Shamir-Adleman Public-Key Cryptosystem
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
S-CL	Security Closed Loop
S-CLF	Security Closed Loops Functions
S-RO	Secure Resource Orchestrator
SCM	Security Context Manager
SDR	Software Defined Radio
SEN	Sensitive
SF	Security Function
SHA-256	Secure Hash Algorithm 256
SHAP	Shapley Additive exPlanations
SHATs	Shapley Values for Time Series Models
SINR	Signal-to-Interference-plus-Noise Ratio
SKG	Secret Key Generation
SLO	Service Level Objective
SNN	Spiking Neural Network
SNR	Signal-to-Noise Ratio
SNS JU	Smart Networks and Services Joint Undertaking
SoTA	State of The Art
SPARQL	SPARQL Protocol and RDF Query Language
SSe	Security Services
SSLA	Security Service Level Agreement
TLS	Transport Layer Security

ToC	Table of Contents
ToF	Time-of-Flight
TPR	True Positive Rate
TRL	Technology Readiness Level
t-SNE	t-Distributed Stochastic Neighbor Embedding
TX.Y	Task X.Y
UC	Use Case
UE	User Equipment
ULA	Uniform Linear Array
URL	Uniform Resource Locator
VNC	Virtual Network Computing
VPN	Virtual Private Network
VSB	Vertical Service Blueprint
WL-CUSUM	Windowed Limited Cumulative Sum
WP	Work Package
XAI	Explainable AI
ZTSO	Zero-Touch Security Orchestrator
ZTSP	Zero-Touch Security Platform

1 Introduction

This deliverable presents the final validation results of the ROBUST-6G project. Building on the validation framework established in D6.1 and the intermediate results reported in D6.2, D6.3 concludes this progression by executing and reporting on full scenario-level validation across the partner testbeds. It constitutes the definitive record of the consortium's integration and proof-of-concept activities under WP6, covering the validated use cases, the measured KPIs, and the prototypes developed across the project.

1.1 Objective of the Document

The purpose of this deliverable is to provide a final, comprehensive account of the technical validation of the ROBUST-6G platform. This includes the end-to-end execution of all use case scenarios, the demonstration of prototypes across the partner testbeds, and the measurement of Key Performance Indicators (KPIs) against the targets defined in the Description of Action (DoA).

D6.1 established the validation plan, introduced the flow-based pipeline, and defined the unified testbed strategy. D6.2 executed that plan, reporting on component and flow-level validation progress, performing the gap analysis against the D2.2 reference architecture, and identifying the open points to be resolved in the final phase.

D6.3 concludes this progression. It introduces the prototypes as the demonstrable units of ROBUST-6G, addresses each open point carried forward from D6.2, presents the validation results for all three use cases and their scenarios, and provides the consolidated KPI attainment evidence required to assess compliance with DoA commitments.

The scope of this deliverable covers the end-to-end execution of all use case scenarios across the partner testbeds; the demonstration of the prototypes, the resolution of open points from D6.2; and the final documentation of key inter-component interface specifications.

The validation approach used in this deliverable is reported as follows: each prototype is assessed individually, KPI attainment is reported per use case and scenario, and the project Global Objectives defined in the DoA are verified at the integrated project level. The consolidated view of all results is provided in Chapter 5, with full traceability available in the supporting appendices.

1.2 Structure of the Document

The remainder of this document is organised as follows.

Chapter 2 presents the final state of the ROBUST-6G platform, covering the system architecture, the gap analysis, the status of all components, flows, and scenarios, an overview of the prototypes, and the unified testbed configuration.

Chapter 3 describes each of the five ROBUST-6G prototypes, presenting their objectives, position within the architecture, and composition. Prototype 5 additionally includes a dedicated demonstration storyline section, reflecting its scope spanning multiple use cases and scenarios.

Chapter 4 presents the technical description of each of the three use cases, covering the scenario objectives, the position within the architecture, the functional flows with their diagrams, and the prototype used for each scenario.

Chapter 5 consolidates all validation results into a single project-level view, presenting the prototype validation assessment, the KPI attainment results per use case, the Global Objective verification and the overall evaluation.

Chapter 6 provides the conclusions, lessons learned, and a summary of the project's scientific and technical contributions.

Five appendices support the main body, covering the component and flow validation summaries, the final interface specifications, the demo materials index, the KPI and Objective Traceability Matrix, dataset catalogue and the Technology Readiness Level (TRL) assessment for each prototype.

Figure 1.1 provides a visual overview of the document structure and the reading flow between chapters.

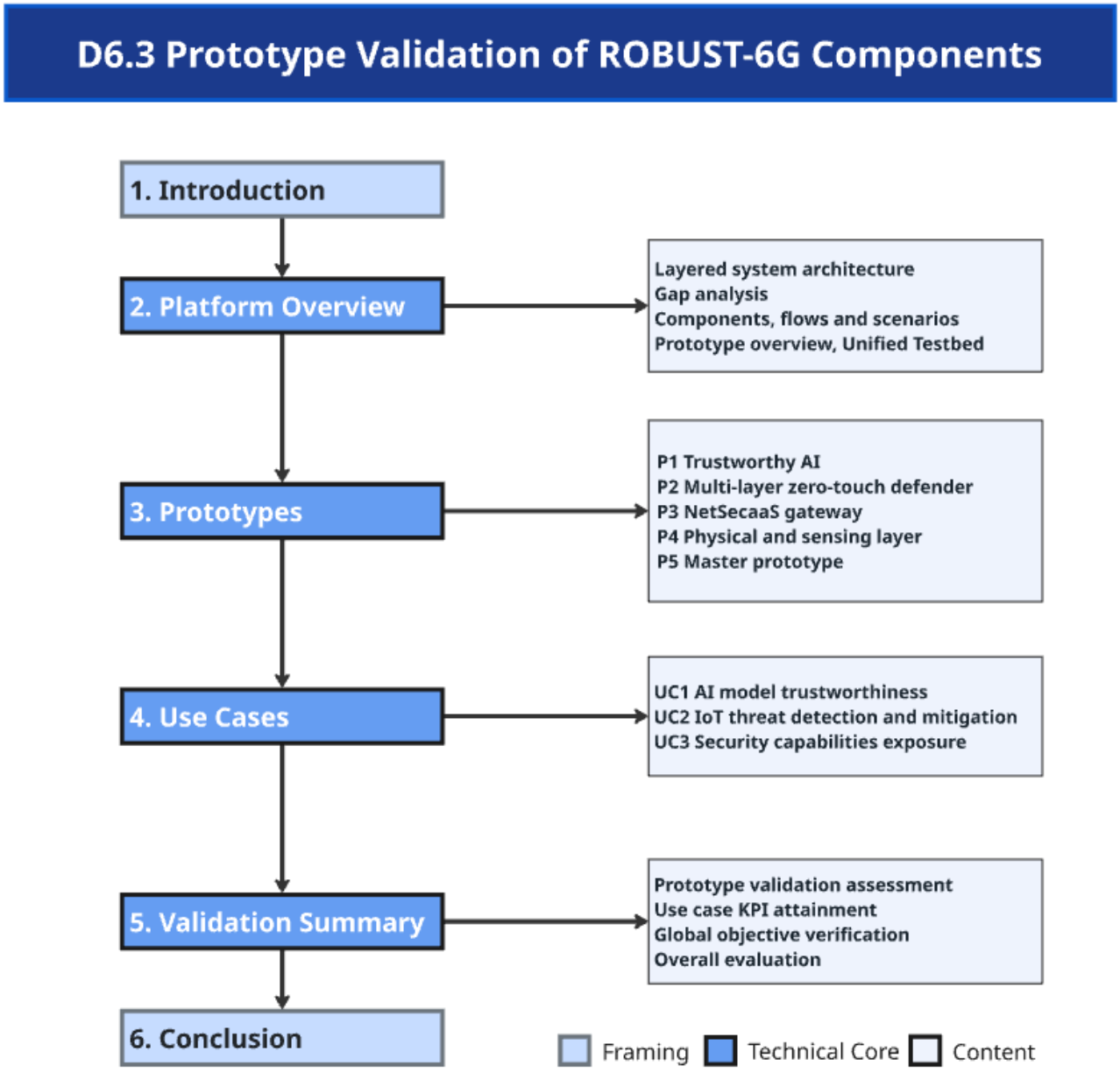

Figure 1.1: D6.3 Chapter Flow and Reading Guide

1.3 Terminology and Definitions

The following terms are used consistently throughout this deliverable to describe the ROBUST-6G validation approach and its key elements.

Component: An individual building block developed by a project partner that provides a specific capability within the ROBUST-6G architecture, such as a software module, Artificial Intelligence (AI) model, algorithm, security function, API, or framework.

Component-Based Validation: The process of validating an individual component in isolation to verify its functionality, performance, robustness, security, and compliance with its defined specifications before integration with other project elements.

Flow: A structured unit of integration and validation that represents a sequence of interconnected components, each fulfilling a specific functional role within a use case scenario. A flow defines clear input triggers, internal logic, and expected outputs; maps directly to the ROBUST-6G reference architecture; and is deployed and validated on one or more Partner Testbed Assets. It serves as the intermediate building block between individual component validation and full scenario-level testing.
Flow-Based Validation: The process of assessing a sequence of interconnected components against their defined KPIs and acceptance criteria to confirm correct interaction, data exchange, interface compatibility, and collective achievement of the intended functionality and performance targets.

Partner Testbed Asset (PTA): An infrastructure resource contributed by a project partner to support component, flow, and scenario validation, providing the computing, networking, radio, cloud, edge, or software resources required for testing and experimentation.

Unified Testbed: A federated validation environment created by interconnecting multiple Partner Testbed Assets, enabling end-to-end testing of distributed ROBUST-6G solutions across partners, domains, and technology environments.

Use Case: A real-world challenge, operational objective, or business problem that ROBUST-6G aims to address, defining the validation context and expected outcomes without prescribing a specific technical implementation.
Use Case-Based Validation: The process of validating an end-to-end use case by integrating multiple components and flows within realistic scenarios to demonstrate that the intended operational objectives and performance targets are achieved.

Scenario: A specific operational situation within a use case that combines multiple flows to validate a particular aspect of the ROBUST-6G solution under realistic conditions.

Prototype: A concrete implementation of one or more ROBUST-6G innovations, serving as a working demonstrator used to validate the feasibility and effectiveness of a specific technical solution.
Prototype Validation: The process of validating a specific ROBUST-6G innovation or demonstrator to prove its technical feasibility, effectiveness, maturity, and readiness for integration, exploitation, or further development.

Use Case KPI: A technical performance indicator defined in the Description of Action (DoA) and used to validate that ROBUST-6G components, flows, scenarios, and prototypes achieve their intended functionality and performance targets within a specific use case.

Project Objective KPI: A project-level performance indicator defined in the Description of Action (DoA) and linked directly to the project's scientific, technical, and operational objectives,

demonstrating the overall impact and success of ROBUST-6G through evidence collected from one or more Use Case KPIs.

Validation Hierarchy: The structured, three-step integration and validation pipeline followed by ROBUST-6G, in which individual components are first validated in controlled environments, then integrated into flows and validated collectively, and finally assembled into scenarios and validated end-to-end on the Unified Testbed. Evidence collected through this pipeline is used to assess Use Case KPIs and Project Objective KPIs defined in the DoA. This term is used in D6.3 as a summary label for the three-step integration pipeline established in D6.1. Prototype Validation runs alongside this hierarchy as a separate track, assessing each ROBUST-6G prototype independently of use case scenario KPI attainment.

2 ROBUST-6G Platform Overview

This section consolidates the architectural and integration foundations that underpin the final validation of the ROBUST-6G platform. The layered design of the end-to-end system architecture is presented. The gap analysis, a structured process that maps each flow to the reference architecture and flags any missing components, overlaps, or unclear interfaces, was initiated in D6.1 and refined in D6.2, is reported for the last time, confirming the alignment of all 18 active flows with the final reference architecture updated in D2.3. The status of the components along with the flows that integrate them into platform-level capabilities are revisited. Validation scenarios that exercise these flows across various use cases are provided. The five prototypes that translate the architecture into concrete demonstrators are also introduced in this section. These prototypes range from layer-specific implementations of trustworthy AI, zero-touch defence, NetSecaaS exposure, and physical-layer resilience, up to the integrated Master Prototype that showcases their joint operation as a unified 6G security platform.

2.1 System Architecture

The ROBUST-6G architecture presents a comprehensive, end-to-end security framework designed for future 6G networks, structured around a set of tightly integrated architectural layers. These layers collectively enable autonomous, intelligent, and scalable protection across distributed environments, spanning from the physical radio interface to application-facing services. They operate as a coordinated system driven by continuous data flows, AI-powered analytics, and closed-loop automation. Figure 2.1 shows the main elements of the architecture.

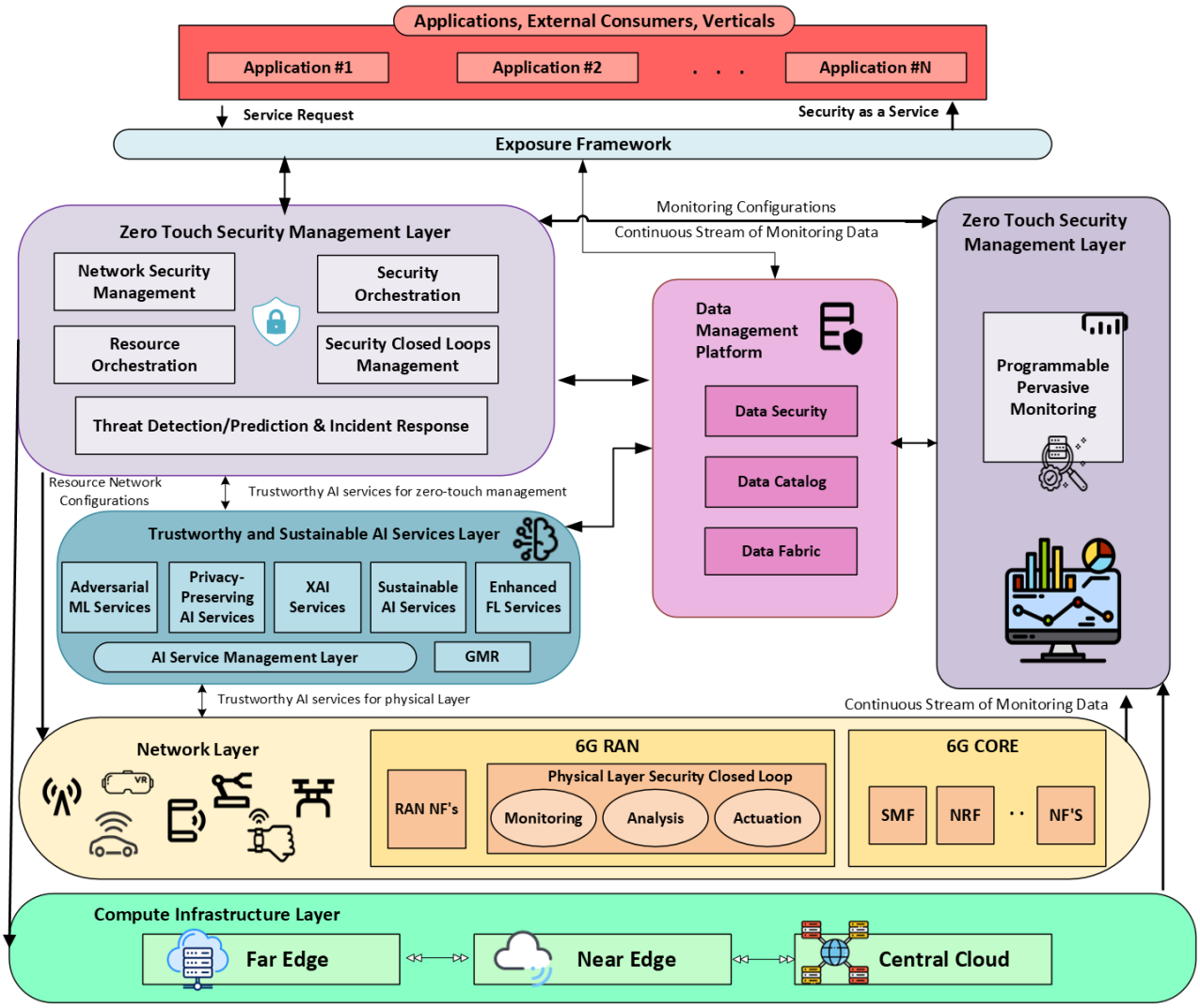

Figure 2.1: ROBUST-6G Architecture

At the outermost boundary of the architecture lies the Exposure Framework, which serves as the interface between the internal security ecosystem and external consumers such as applications, vertical industries, and service providers. This layer abstracts the complexity of the underlying system into a set of secure and standardised APIs, allowing external entities to access advanced security capabilities in a programmable way. Through this interface, consumers can request threat intelligence, invoke mitigation actions, or define high-level security requirements such as service-level agreements. The importance of this layer lies in transforming security from an internal operational concern into an accessible service, effectively enabling "security-as-a-service" in 6G systems. It ensures that the sophisticated intelligence generated within the architecture can be consumed in a flexible and interoperable manner without exposing implementation complexity.

Beneath the Exposure Framework, the architecture is grounded in the Programmable Pervasive Monitoring layer, which provides continuous visibility into the state of the network. This layer functions as the sensing mechanism of the system, collecting telemetry, performance metrics, alarms, and anomaly indicators from a wide range of sources across the edge–cloud continuum. Its programmability enables it to adapt dynamically to new monitoring requirements, emerging threats, or evolving performance indicators. The data collected at this stage forms the raw input for all higher-level intelligence and decision-making processes. By maintaining persistent awareness of network

behaviour across heterogeneous environments, this layer ensures that the system can detect early signs of degradation or attack, thereby supporting proactive rather than reactive security strategies.

The data gathered through monitoring is then handled by the Data Management Platform, which constitutes the central data backbone of the architecture. Its role extends beyond simple storage; it integrates, secures, and governs data flows across the system. By utilising a combination of data fabric [R6G26-D23], a semantic knowledge graph, and governance mechanisms, it enables interoperable and privacy-preserving data sharing across multiple domains. This platform ensures that data from various sources—ranging from physical-layer measurements to application-level metrics—can be normalised, discovered, and accessed under strict policy control. The presence of a knowledge graph introduces semantic consistency, allowing heterogeneous datasets to be linked and interpreted in a unified manner. In addition, identity management, authentication, and policy-based authorisation mechanisms ensure that data access is controlled and auditable. As a result, the platform acts as the "nervous system" of the architecture, providing reliable and trustworthy data to all other layers that depend on it.

Building on this data foundation, the Trustworthy and Sustainable AI Services layer introduces intelligence into the system. This layer is responsible for developing, managing, and deploying AI models that support security operations across the architecture. A defining aspect of this layer is its emphasis on sustainability and trustworthiness, which encompasses robustness against adversarial manipulation, privacy preservation, and explainability. It supports distributed learning paradigms such as federated learning, allowing models to be trained across decentralised environments without exposing raw data. At the same time, it provides explainability mechanisms and confidence metrics that make AI-driven decisions transparent and auditable. Sustainability is also treated as a core concern, with strategies aimed at minimising the computational and energy footprint of AI processes. The outputs of this layer take the form of predictions, classifications, and anomaly detections, which are enriched with contextual and trust-related information and passed on to the orchestration layer for action.

The Zero-Touch Security Management layer acts as the central decision and control entity in the architecture. It leverages the insights produced by the AI layer along with data from the monitoring and data management layers to orchestrate security operations across the network. Its distinguishing feature is its ability to operate autonomously through closed-loop mechanisms. These loops continuously monitor system behaviour, analyse potential threats, decide on appropriate responses, and execute mitigation actions without requiring human intervention. The layer translates high-level security intents into concrete actions by decomposing them into deployable configurations and coordinating multiple domain-specific orchestrators. This enables real-time detection and mitigation of threats while maintaining consistency across distributed environments. By automating the full lifecycle of security management—from policy definition to enforcement and validation—it significantly reduces operational complexity and improves response times, embodying the zero-touch vision of future 6G systems.

Complementing this system-wide orchestration and the security closed loom management in the management layer, the Physical Layer Security closed-loop layer extends security mechanisms down to the radio interface, addressing threats that originate at the lowest level of the communication stack. This layer operates through its own monitoring–analysis–actuation cycle, focusing specifically on signal-level characteristics such as channel behaviour, spatial properties, and electromagnetic patterns. By applying AI-driven analysis to these parameters, it can detect attacks such as jamming, spoofing, or unauthorised signal manipulation. The actuation component then enforces rapid countermeasures directly at the physical layer, including adjustments to transmission power,

beamforming configurations, or authentication mechanisms. The proximity of this layer to the radio environment enables extremely low-latency responses, which are essential for mitigating fast-evolving threats. In this way, the architecture ensures that security is enforced not only at the software and network levels but also at the physical foundation of communication.

Taken together, these layers form a deeply interconnected architecture in which data flows upward from monitoring to intelligence and decision-making layers, while control actions propagate downward through orchestration and actuation mechanisms. The Exposure Framework bridges this internal ecosystem with external actors, enabling the deployment of programmable security services. The overall design reflects a shift from static, reactive security models to dynamic, AI-driven, and fully automated protection strategies. By tightly integrating monitoring, data governance, AI intelligence, orchestration, and physical-layer enforcement into a unified system, the ROBUST-6G architecture provides a scalable and adaptive solution capable of addressing the complexity and threat landscape of future 6G networks.

2.2 Gap Analysis

Building on the gap analysis methodology established in D6.1 and the intermediate findings reported in D6.2 (Section 2.4, "Gap Analysis and Alignment with the Architecture"), this section presents the final closure of all open points identified during the project lifecycle. In D6.2, 19 flows across three use cases were examined against the reference architecture defined in D2.2, and a set of open points were identified. In the final validation cycle, the active flow count has been updated to 18.

It should also be noted that between the intermediate and final validation phases, the reference architecture was updated within the scope of WP2. This update introduced structural clarifications that are directly relevant to the open points identified in D6.2, and are explicitly referenced in the per-use-case findings below.

Use Case 1.1 - Decentralised Federated Learning: D6.2 reported one open point: the interactions between the Global Model Repository and potential external consumers had not been formally defined. This point has been resolved: as clarified during inter-partner technical discussions, any authorized external element accesses the GMR through the Trustworthy AI Service Exposure interface, without requiring a dedicated flow-level coupling.

Use Case 1.2 - Physical and Sensing Layer Trustworthiness: D6.2 reported no open points for this use case. Since all components involved in these flows operate within the Physical Layer Security Closed Loop, architectural alignment is inherently maintained. No further action was required.

Use Case 2 - Automatic Threat Detection and Mitigation: The first concern was the visibility of closed-loop data interactions within the functional architecture. As clarified during inter-partner technical discussions, closed-loop instances are not expected to appear as discrete entities in the functional reference architecture; they are a deployment-level construct. Their governance is represented by the Security Closed Loops Management block within the Zero-Touch Security Management Layer, which is explicitly present in the final reference architecture. This point is formally closed.

The second open point concerned the harmonization of monitoring responsibilities across Internet of Things (IoT) and Radio Access Network (RAN) layers. At the time of D6.2, this point was flagged as requiring clarification due to the early integration stage of the UC2 flows. As components were further developed and integrated during the final validation cycle, it became evident that the Programmable Pervasive Monitoring platform naturally serves as the unified collection point for both

IoT and RAN telemetry. This was confirmed through the practical execution of the UC2 flows, which demonstrated that no additional architectural mechanism was required to harmonize monitoring responsibilities across the two domains. This point is formally closed.

Use Case 3 - Security Capabilities Exposure (NetSecaaS): D6.2 reported one open point: the identification of all modules that can be exposed through the NetSecaaS interface had not been completed. This point is resolved in D6.3: the flows are re-defined; three functional flows each mapping onto a set of exposed capabilities and components. The exposed module set is now fully specified through these flow definitions.

The Flow alignment Table 2.1 shows the gap status of each use case flows and architecture.

Table 2.1: Gap Analysis Status

UC	ID	Name	Architectural Layers	Gap Status
1.1	1	Privacy and Decentralization	Trustworthy and Sustainable AI Services Layer	Aligned
1.1	2	Evaluation of Model Robustness	Trustworthy and Sustainable AI Services Layer	Aligned
1.1	3	Sustainability Evaluation	Trustworthy and Sustainable AI Services Layer	Aligned
1.1	4	Explainability of the Models Obtained	Trustworthy and Sustainable AI Services Layer	Aligned
1.1	5	Privacy-Enhanced DFL	Trustworthy and Sustainable AI Services Layer	Aligned
1.2	1	PHY layer trustworthiness evaluation	Physical Layer Security Closed Loop, Zero-touch Security Management Layer	Aligned
1.2	2	Mutual authentication	Physical Layer Security Closed Loop, Zero-touch Security Management Layer	Aligned
1.2	3	(Fast) Secret key generation	Physical Layer Security Closed Loop,, Zero-touch Security Management Layer	Aligned
2.1	1	Proactive Security Enforcement	Zero-touch Security Management Layer, Trustworthy and Sustainable AI Services Layer	Aligned
2.1	2	Threat Detection combining Network and IoT Data	Zero-touch Security Management Layer	Aligned
2.1	3	Threat Mitigation via reactive plan execution	Zero-touch Security Management Layer	Aligned
2.2	1	Investigative Loop	Zero-touch Security Management Layer, Trustworthy and Sustainable AI Services Layer	Aligned
2.2	2	Resolutive Loop	Zero-touch Security Management Layer, Trustworthy and Sustainable AI Services Layer	Aligned
2.3	1	Loops and Coordination Deployment	Zero-Touch Security Management Layer; Data Management Platform	Aligned
2.3	2	Loops Execution and Conflict Resolution	Zero-Touch Security Management Layer; Data Management Platform	Aligned
3	1	Security Capabilities discovery data exposure	Data Management Platform, Exposure Framework	Aligned

3	2	XAI Analytics Data	Data Management Platform, Exposure Framework, Trustworthy and Sustainable AI Services Layer	Aligned
3	3	Simplified SSLA enforcement	Data Management Platform, Exposure Framework, Zero-touch Security Management Layer,	Aligned

2.3 Components

This section summarises the status of all components developed across WP3, WP4 and WP5 and integrated into the ROBUST-6G platform. As established in D6.1 and tracked through D6.2, components are the elementary building blocks that are composed into the flows, scenarios and prototypes validated in this deliverable. The objective here is to confirm that the full component set has been completed and to indicate, for each component, the deliverable in which its detailed results are reported and the use case or prototype in which it is exercised. All active components defined in D6.1 have been completed and validated. One component (CNXW04) was added during the project as result of the separation of the CL Management from CNXW01. The consolidated reference table, with the development task, reporting deliverable, concrete outputs and use-case / prototype mapping for every component, is provided below.

Table 2.2 Status overview of ROBUST-6G components

ID	Component	Partner	Reported in	Used in (UC / Prototype)	Status
CEBY01	Enhanced AI/ML robustness against adversarial attacks	EBY	D3.2	Ad hoc PoC	Completed
CEBY02	Privacy-preserving and security-enhanced DFL	EBY	D3.4	UC1.1 / P1	Completed
CEBY03	XAI-based detection and mitigation for adversarial attacks	EBY	D3.4	UC3 / P3	Completed
CEBY04	Signal/attack identification of electromagnetic signals	EBY	D5.2	Ad hoc PoC	Completed
CEBY05	Identification/authentication of legitimate devices (RIS/non-RIS, anti-spoofing)	EBY	D5.2	Ad hoc PoC	Completed
CTID01	Data Fabric	TID	D2.3	UC3 / P3, P5	Completed
CTID02	Data Governance	TID	D2.3	UC3 / P3, P5	Completed
CTID03	Security Capabilities Exposure (NetSecaaS)	TID	D2.3	UC3 / P3, P5	Completed
CUMU01	Programmable Monitoring Platform (PMP)	UMU	D4.4	UC2 / P2	Completed
CUMU02	DFL Framework	UMU	D3.4	UC1.1 / P1	Completed
CUMU03	Reputation-Based Trust Management System	UMU	D3.4	UC1.1 / Ad hoc PoC	Completed
CUMU04	Enhanced AI/ML Model Robustness (Krum)	UMU	D3.4	UC1.1 / P1	Completed
CUMU05	XAI Integration for Model Explainability	UMU	D3.4	UC1.1 / P1	Completed
CCHA01	Physical Layer Security in NOMA-MIMO systems	CHA	WP5	—	Completed
CCHA02	Datasets generation and fingerprinting for PLS	CHA	WP5	—	Completed
CUCD01	Distributed FL Poisoning Attack & Defense	UCD	WP3	—	Completed
CUCD02	Evasion Attack Detection	UCD	WP3	—	Completed

CUCD03	XAI-IDS	UCD	WP3	UC3 (XAI)	**Completed**
CUPD01	Decentralised FL with ADMM	UNIPD	D3.4	UC1.1 / P1	**Completed**
CUPD02	Spiking Neural Network Simulator	UNIPD	D3.4	UC1.1 / P1	**Completed**
CUPD03	Jamming Detection	UNIPD	WP5	—	**Completed**
CUPD04	PHY-layerenhanced Authentication &Key Agreement	UNIPD	WP5	—	**Completed**
CUPD05	Cross-Layer Holistic Security Anomaly Detection	UNIPD	WP5	—	**Completed**
CNXW01	Zero-Touch Security Orchestrator	NXW	D4.4	UC2 / P2	**Completed**
CNXW02	Network CNN-based IDS	NXW	D4.4	Not applicable	**Completed**
CNXW03	Resource Orchestrator	NXW	D4.4	UC2 / P2	**Completed**
CNXW04	Closed-Loop Management (S-CL Manager)	NXW	D4.4	UC2.1, UC2.2, UC2.3 / P2	**Completed**
CENS01	PHY monitoring (SNR, LoS/NLoS)	ENSEA / CYU	WP5	UC1.2 / P4	**Completed**
CENS02	Secrecy and information leakage	ENSEA / CYU	WP5	Not applicable	**Completed**
CENS03	Trustworthy Sensing and Localization	ENSEA / CYU	WP5	UC1.2 / P4	**Completed**
CENS04	AoA-based Physical Layer Authentication	ENSEA / CYU	D5.3	UC1.2 / P4	**Completed**
CENS05	Fast SKG using LSTM networks for privacy amplification	ENSEA / CYU	D5.3	UC1.2 / P4	**Completed**
CLIU01	Semantics-aware task scheduling in Federated Learning	LIU	WP3	UC1.1 / P1	**Completed**
CLIU02	Remote estimation under heterogeneous semantic significance	LIU	WP3	—	**Completed**
CEUR01	XAI AI/ML algorithms	EUR	WP3	—	**Completed**
CEUR02	Risk-averse Resource Management Framework	EUR	D4.4	—	**Completed**
CTHA01	Security Orchestrator	THALES	D4.4	UC2 / P2, P3, P5	**Completed**
CTHA02	Monitoring and Closed-Loop Remediation System	THALES	D4.4	UC2 / P2	**Completed**
CGHM01	RF Fingerprinting Migration	GOHM	D5.2	PoC (KPI6/KPI7)	**Completed**
CGHM02	RF-PREDICT	GOHM	D5.2	PoC (KPI6/KPI7)	**Completed**
CAXN01	Proactive Threat Prediction and Mitigation for 6G Security Orchestrators	AXON	D4.4	UC2 / P2	**Completed**

CAXN02	CryptoToN-IoT – cryptomining-augmented IoT network-flow dataset for 6G security AI	AXON	D4.4	UC2 / P2	**Completed**

Component Descriptions

Each component developed across WP2, WP3, WP4 and WP5 is summarised below. The corresponding validation and reporting notes - stating the development task, the deliverables in which results are reported, any concrete outputs (publications, datasets, software releases), and how the component was validated (use case, prototype, KPI, or proof-of-concept) - are provided in the component table in Appendix A.

EBY (Ericsson)

- **CEBY01** — Enhancement of AI/ML model robustness against adversarial evasion and poisoning attacks; validated as a dedicated proof-of-concept (D3.2).
- **CEBY02** — Privacy-enhancing and security techniques that protect the decentralised federated-learning process from data leakage and poisoning; integrated in Prototype 1 / UC1 Scenario 1 and reported in D3.4.
- **CEBY03** — Explainable AI (XAI) based detection and mitigation of adversarial threats; its explainability outputs are exposed through the NetSecaaS interface in UC3 / Prototype 3 (D3.4).
- **CEBY04** — AI/ML solution to classify different types of electromagnetic signals with high accuracy; validated as a proof-of-concept (D5.2).
- **CEBY05** — Identification and authentication of legitimate devices, including Reconfigurable Intelligent Surface (RIS) and non-RIS spoofing scenarios; mapped into the Physical-Layer Security loop (D5.2).

TID (Telefónica)

- **CTID01** — Data Fabric responsible for collecting, processing, storing and semantically integrating security data; central to UC3 and reused by Prototypes 3 and 5 (D2.3).
- **CTID02** — Data Governance plane providing authentication, authorisation and policy-based access control across the exposed capabilities (D2.3).
- **CTID03** — Security Capabilities Exposure (NetSecaaS) gateway exposing security capabilities to third parties via CAMARA-style REST APIs; it is designed to be the only externally visible component for 3rd parties of the ROBUST-6G platform as seen in the context of UC3 (D2.3).

UMU (Universidad de Murcia)

- **CUMU01** — Programmable Monitoring Platform (PMP) for closed-loop, virtualised monitoring, anomaly detection and data aggregation; the unified collection point for IoT and RAN telemetry in UC2 / Prototype 2 (D4.4).
- **CUMU02** — Decentralised Federated Learning Framework providing the core orchestration engine for privacy-preserving model training; the heart of Prototype 1 / UC1 Scenario 1 (D3.4).
- **CUMU03** — Reputation-Based Trust Management System implementing a behaviour-based trust element to weight nodes when sharing model updates; defined in D3.2, with validations in [MMG+26].
- **CUMU04** — Enhanced AI/ML Model Robustness implementing Krum Byzantine-robust aggregation to filter poisoned updates; validated in UC1 Scenario 1 / Prototype 1 (D3.4).
- **CUMU05** — XAI Integration for Model Explainability generating SHAP/t-SNE explainability artefacts per training round; validated in UC1 Scenario 1 / Prototype 1 (D3.4).

CHA (Chalmers)

- **CCHA01** — Physical Layer Security mechanisms for Non-Orthogonal Multiple Access (NOMA)-Multiple-Input Multiple-Output (MIMO) systems addressing eavesdropping mitigation at the physical layer (WP5) / D5.2 and T5.1.
- **Validation & reporting**: Developed under WP5 to address physical-layer eavesdropping mitigation in uplink NOMA-MIMO/Integrated Sensing and Communication (ISAC) systems, with the corresponding methodology and performance evaluation contributing to D5.2. A paper presenting the 'Robust Beamforming Design for Secure Uplink NOMA-ISAC' has been submitted to IEEE Transactions on Wireless Communications.
- **CCHA02** — Generation of Radio Frequency (RF) digital-twin datasets and fingerprinting material supporting physical-layer security research (WP5) / D5.3 and T5.2.
- **Validation & reporting**: Generated a wireless Channel State Information (CSI) dataset using the 3rd Generation Partnership Project (3GPP)-compliant Clustered Delay Line (CDL) model, covering propagation scenarios CDL-A to CDL-E and mobility conditions ranging from 3 to 120 km/h (WP5).

UCD (University College Dublin)

- **CUCD01** — Layer-wise Relevance Propagation (LRP)-based study of poisoning and inference attacks together with robust defences for federated-learning systems (WP3).
- **CUCD02** — Evasion-attack detection model targeting beamforming prediction (WP3).
- **CUCD03** — XAI-Intrusion Detection System (IDS) using SHAP explanations to improve detection performance, interpretability and efficiency of AI/ML intrusion detection; its explainability outputs feed the UC3 exposure flow (WP3).

UNIPD (University of Padova)

- **CUPD01** — Secure and decentralised federated-learning aggregation method based on ADMM, optimising training efficiency and scalability; part of the Sustainable AI service in UC1 Scenario 1 / Prototype 1 (D3.4).
- **CUPD02** — Spiking Neural Network simulator providing sparse, event-driven models that reduce the edge compute footprint; part of the Sustainable AI service in UC1 Scenario 1 (D3.4).
- **CUPD03** — Machine-learning jamming detection from In-phase/Quadrature (I/Q) samples (WP5).
- **CUPD04** — Novel PHY-layer Authentication and Key Agreement protocols for low-latency, low-complexity scenarios, including false-base-station authentication (WP5).
- **CUPD05** — Cross-layer holistic security anomaly-detection system for early anomaly identification (WP5).

NXW (Nextworks)

- **CNXW01** — Zero-Touch Security Orchestrator components for semantic reasoning and cataloguing of Security Functions and Target Environments; core of UC2 / Prototype 2 (D4.4).
- **CNXW02** — Network IDS converting raw traffic into images for using Convolutional Neural Network for intrusion detection.
- **CNXW03** — Resource Orchestrator managing compute, network and storage resources in the cloud edge continuum. Responsible of Security Functions and Security Closed Loops stages deployment. (D4.4).
- **CNXW04** — Closed-Loop Management (S-CL Manager) handling governance and coordination of monitoring/analysis/decision/execution stages composing cloud-native closed loops; drives the multi-loop scenarios UC2.2 and UC2.3 (D4.4). Introduced after D6.1.

ENSEA / CYU

- **CENS01** — PHY monitoring of the physical context (SNR, LoS/NLoS); supports jamming detection and localisation in UC1 Scenario 2 / Prototype 4 (WP5).

- **CENS02** — Estimation of available secrecy rate with wiretap-coding and beamforming configuration for a target information-leakage level (WP5).
- **CENS03** — Trustworthy sensing and localisation, including Sybil-attack detection and sensing-accuracy trustworthiness; UC1 Scenario 2 / Prototype 4 (WP5).
- **CENS04** — AoA-based Physical Layer Authentication preventing spoofing/impersonation; validated on real CSI in UC1 Scenario 2 / Prototype 4 (D5.3).
- **CENS05** — Fast secret-key generation using LSTM-based privacy amplification; achieved a 100% reconciliation rate in UC1 Scenario 2 / Prototype 4 (D5.3).

LIU (Linköping University)

- **CLIU01** — Semantics-aware, user-oriented task-scheduling algorithm for federated learning; part of the Sustainable AI service in UC1 Scenario 1 (WP3).
- **CLIU02** — Remote state-estimation framework that weights data by semantic significance leveraging system history (WP4).

EUR (EURECOM)

- **CEUR01** — Set of techniques for ensuring and enhancing the trustworthiness of AI/ML algorithms (WP3).
- **CEUR02** — Resource control and optimisation framework incorporating risk aversion and subjective performance assessment (WP3).

THALES

- **CTHA01** — Security Orchestrator implementing security policies across network, IT and application services on edge and cloud infrastructure; The security policy implementation and enforcement is based on SSLAs (Security Service Level Agreement), which are managed by a policy manager component inside the ZTSO (D4.4). This component also implements interfaces and verification processes to coordinate the actions taken by the others ZTSO's components (NetSecaaS Gateway, Generative Artificial Intelligence (GenAI) Gateway, Ontology Manager, Context Manager) to maintain their compliance with the enforced SSLAs.
- **CTHA02** — Monitoring and Closed-Loop Remediation System using extended Berkeley Packet Filter (eBPF)-based observation and closed-loop security remediation (D4.4). This component implements a GenAI Gateway inside the ZTSO (D4.4) to leverage external generative AI services by contextualizing security policies enforcement plans, or security alerts raised by the eBPF monitoring system. From SSLA sent by the policy manager (CTHA01) of the ZTSO, this component's output generates remediation workflows that aim to maintain the SSLAs on the target system against security threats. In a second mode and from security alerts sent by the alert manager of the ZTSO (D4.4), this component's output also generates remediation workflows that aim to mitigate security incidents detected in the target system.

GOHM

- **CGHM01** — RF Fingerprinting Migration model enabling domain-invariant RF fingerprinting; informed a standalone PoC providing KPI6/KPI7 evidence (D5.2).

  Validation & reporting: Developed under Task T5.1. Initial results are reported in D5.2. A paper presenting this work was accepted and published at EuCNC & 6G Summit 2025 [AYS26]. A real-hardware dataset was collected and publicly released on Zenodo [AYA+25].

- **CGHM02** — RF-PREDICT, a predictive model anticipating RF-fingerprint changes for low-power sensors to support trustworthy sensing (D5.2).

  Validation & reporting: Developed under Task T5.3. Initial results are reported in D5.2. A longitudinal dataset was collected and shared on Zenodo under restricted access, pending publication [AYY+26].

Cross-component note: the expertise accumulated while developing CGHM01 and CGHM02 informed a standalone RF-fingerprinting proof-of-concept, developed and validated separately, which provides KPI-level evidence for KPI6 and KPI7 (reported in Section 5.2.2.2).

AXON

- **CAXN01 –** Proactive Threat Prediction and Mitigation for 6G Security Orchestrators. Three containerised AI functions for the zero-touch security loop: network-attack detection (XGBoost, 98.15%), automated mitigation over the M1–M16 action matrix (Binary-Relevance Random Forest, 98.89%), and 5-minute-ahead attack prediction (Binary-Relevance Random Forest over TSFresh features, 95.36%), enabling pre-emptive Collaborative Automated Course of Action Operations (CACAO)/Open Command and Control (OpenC2) remediation in UC2 / Prototype 2 (recall deliverable D4.4 at paragraph §2.3). Trained on CAXN02 [AXPTP].
- **CAXN02 –** CryptoToN-IoT, a cryptomining-augmented IoT network-flow dataset for 6G security AI (named CICToN-IoT in deliverable D4.4 at paragraph §2.3). AXON's extension of UNSW's ToN-IoT dataset with Coinhive/Madominer/Xmrstack traffic, re-processed via CICFlowMeter – 16,422,866 flows, 12 attack types, temporal 60:20:20 split. CryptoToN-IoT has been used as the training and benchmark dataset for CAXN01 [AXCCA].

2.4 Flows

As described in D6.1 and D6.2, the different components of ROBUST-6G, reported in Section 2.3, are integrated into flows, representing a logical chain that delivers a particular function of the ROBUST-6G Platform. These flows serve as the critical intermediate step between individual component validation and full scenario validation. To track the progressive integration of the platform, the evolution and execution status of these workflows have been closely monitored. In total, 19 distinct flows were initially examined and mapped to the reference architecture in D6.2. Table 2.3 reports the updated status overview, detailing the total number of current flows, whether they remain consistent with the D6.2 baseline, and if any modifications were introduced during this validation phase.

Table 2.3: Functional Flows Status - Summary

Scenario Identifier	Scenario Name	Initial Number of Flows	Final Number of Flows	Updates
UC1.1	Decentralised federated learning for joint privacy-preserving ML/DL model training	5	5	UC1.1_05 is treated as a separate demo, not integrated in the DFL Framework.
UC1.2	Physical and sensing layer trustworthiness and resilience	3	3	Validation of UC1.2 moves from NYUSim-simulated CSI (D6.2) to a real indoor maMIMO dataset, initially described in D5.1. The 3 flows defined in D6.2 remain the same, but the components integrated in each flow are re-structured around the real-data validation under Prototype 4.
UC2.1	Device violation to cause economic harm (a)	3	3	UC2.1_01 now includes an optional human-validation step

				for the generated Security Service and IRP; the rule-based detection/decision in UC2.2_02 was re-architected around the PMP, CACAO playbooks, and OpenC2 actuation.
UC2.2	Fraudulent usage of device resources	2	2	Now models the escalation between loops via a service_update request to the orchestrator (rather than a static second-loop trigger); AI Threat Detection module and ThingsBoard OpenC2 actuator are deployed on demand.
UC2.3	Device violation to cause economic harm (b)	2	2	Reflects the move to a five-farm, two-zone hierarchical architecture coordinated by a centralised Master Loop via the Data Fabric, replacing the original two-field internal/external loop design.
UC3	Security Capabilities Exposure (NetSecaaS)	4	3	UC3_01 and UC3_02 are considered as a single flow UC3_01. The data encoded in the token already allows the retrieval of the accessible capabilities, therefore there is no need to split the information into two separate APIs or processes. As consequence UC3_02 corresponds to old UC3_03, and UC3_03 to UC3_04

In the case of UC 1.1, five different flows were defined, spanning from foundational privacy and decentralisation baselines to advanced privacy-enhanced collaborative learning. Table 2.4 reports an overall view of these flows, mapping their current status and any modifications with respect to what was initially declared in D6.2. Section 4.1.1 provides a detailed report on the implementation and adjustments of these flows.

Table 2.4: UC1.1 Flows Status

Flow ID	Flow Name	Status
UC1_1_01	Privacy and decentralization	Validated
UC1_1_02	Evaluation of model robustness	Validated
UC1_1_03	Sustainability evaluation	Validated
UC1_1_04	Explainability of the obtained model	Validated
UC1_1_05	Privacy-Enhanced DFL	Validated

In the case of UC 1.2, three different flows were defined, spanning from initial physical layer trustworthiness evaluations to fast secret key agreement configurations. Table 2.5 reports an overall view of these flows, mapping their consistency and structural evolution during this validation cycle. Section 4.1.1 provides a detailed report on the implementation and adjustments of these flows.

Table 2.5: UC1.2 Flows Status

Flow ID	Flow Name	Status
UC1_2_01	PHY layer trustworthiness evaluation	Validated
UC1_2_02	Mutual Authentication	Validated
UC1_2_03	(fast) Secret Key Generation	Validated

In the case of UC 2.1, three different flows were defined, spanning from proactive security automation deployment to reactive threat decision and execution actions. Table 2.6 reports an overall view of these flows, mapping their consistency and structural evolution during this validation cycle. Section 4.2.1 provides a detailed report on the implementation and adjustments of these flows.

Table 2.6: UC2.1 Flows Status

Flow ID	Flow Name	Status
UC2_1_01	Proactive Security Enforcement	Validated
UC2_1_02	Threat Detection combining Network and IoT Data	Validated
UC2_1_03	Threat Mitigation via reactive plan execution	Validated

In the case of UC 2.2, two different flows were defined: initial explorative investigative loops to second-stage resolutive remediation. Table 2.7 reports an overall view of these flows, mapping their consistency and structural evolution during this validation cycle. Section 4.2.2 provides a detailed report on the implementation and adjustments of these flows.

Table 2.7: UC2.2 Flows Status

Flow ID	Flow Name	Status
UC2_2_01	Investigative Loop	Validated
UC2_2_02	Resolutive Loop	Validated

In the case of UC 2.3, two different flows were defined: the deployment of the hierarchical multi-loop and coordination infrastructure across five smart farms, followed by the runtime execution and conflict-resolution logic between local and master loops. Table 2.8 reports an overall view of these flows, mapping their consistency and structural evolution during this validation cycle. Section 4.2.3 provides a detailed report on the implementation and adjustments of these flows.

Table 2.8: UC2.3 Flows Status

Flow ID	Flow Name	Status
UC2_3_01	Loops and Coordination Deployment	Validated
UC2_3_02	Loops Execution and Conflict Resolution	Validated

In the case of UC3, four different flows were defined: spanning from secure access governance data exposure APIs to simplified intent-based SSLA enforcement mechanisms. Section 4.3 provides a detailed report on the implementation and adjustments of these flows.

Table 2.9: UC3 Flows Status

Flow ID	Flow Name	Status
UC3*01*-old	Access Governance data exposure	Removed
UC3_01	Security Capabilities discovery data exposure	Validated
UC3_02	XAI Analytics Data	Validated
UC3_03	Simplified SSLA enforcement	Validated

2.5 Scenarios

The platform is validated through a set of meaningful scenarios defined by the different UCs covered by the project. The Use Cases are discussed in Section 4. This section provides a summary of those scenarios divided into three main groups:

- **Trustworthiness and Privacy Preserving.** Two scenarios thataddress the topic of privacy and trustworthiness from two different angles. The first scenario focuses on demonstrating Decentralised Federated Learning techniques for preserving the AI model privacy. In contrast, classical Federated Learning techniques follow a centralised approach that does not fit with distributed scenarios characterising6G. The second scenario focuses on the trustworthiness of the physical and sensing layer in mobile networks. The idea is to incorporate information coming from sensors, signals, and radio equipment (e.g., RF fingerprints between base stations) into the trust analysis. These two scenarios are covered by the Use Case 1.
- **Cyberphysical Anomaly.** Use Case 2 covers the scenarios in this group. In particular, the use case proposes three different scenarios of increasing complexity, all of them based on a common idea: not all the anomalies can be detected solely by analysing the network traffic. In this regard, the scenarios include the monitoring of parameters from IoT devices that can be sensors, or even smart devices (e.g., smart lamps), combined with a detailed network traffic analysis. In the first scenario, the compromise of the HVAC system in a small office is explored. The second scenario considers the compromission of smart devices exploited for crypto mining (cryptojacking attack). The third scenario is focused on a set of smart farms, where the IoT platform for smart agriculture is compromised in one farm: the challenge is to identify this farm and remediate the attack. All these scenarios consider automated remediations based on security closed-loops.
- **Security Capability Exposure and Consumption**. This group considers scenarios where external users lacknetwork security skills but nevertheless need to configure security for their applications. The scenarios aim at demonstrating the capability of the ROBUST-6G system to implement and expose an interface for the Network-Security-as-a-Service (NetSecaaS), characterised by a high-level of abstraction that prevents consumers from needing to know the technical details of the underlying system.

The scenarios summarised so far are used in demos and PoC of the different prototypes developed by the project and described in Section 3.

2.6 Prototypes

This section introduces the prototypes that form the core of the final validation activities within the ROBUST-6G platform. Each prototype addresses a specific set of technological objectives and demonstrates key platform capabilities related to trustworthy AI, zero-touch security automation, capability exposure, and physical-layer resilience. A total of five prototypes is presented in this report:

- **Prototype 1:** Trustworthy AI
- **Prototype 2:** Multi-Layer Zero-Touch Defender
- **Prototype 3:** NetSecaaS Gateway
- **Prototype 4:** Physical and Sensing Layer Trustworthiness and Resilience
- **Prototype 5:** Master Prototype

Prototype 1 delivers an end-to-end AI production pipeline designed to generate, validate, and store trustworthy AI models within the ROBUST-6G platform. The prototype integrates decentralised model training, robustness verification, model explainability, and model reuse capabilities through a shared repository. Its objective is to ensure that AI models are reliable, transparent, reusable, and deployment-ready for operational 6G environments.

Prototype 2 implements an autonomous security closed-loop framework capable of detecting and mitigating cyber threats across IoT and network infrastructures through zero-touch orchestration. The prototype combines monitoring, analysis, automated decision-making, and execution functions to provide adaptive and resilient cybersecurity responses with minimal human intervention. It demonstrates scalable and automated security operations integrated into the ROBUST-6G platform.

Prototype 3 implements the NetSecaaS Gateway, a service exposure framework enabling controlled access to security capabilities, analytics, and platform functionalities through standardised APIs. The prototype allows external consumers to discover, access, and retrieve security-related insights generated by the ROBUST-6G platform while enforcing permission-based exposure policies. It supports interoperability and secure external service consumption while exposing explainable AI outputs and orchestrated platform functionalities.

Prototype 4 demonstrates a physical-layer closed-loop security framework for 6G, where physical observations are continuously monitored through sensing, analysed, and fed into a trust evaluation engine that adapts RAN resource allocation decisions in real time. This closed loop transforms the physical layer from a passive transmission medium into an active trust anchor, enabling continuous monitoring, analysis, and actuation against adversarial behaviour. The prototype leverages sensing capabilities of 6G networks to extract spatial (Angle-of-Arrival) and channel-state features (Channel State Information and Signal-to-Interference Ratio), which serve as intrinsic indicators for: i) PHY anomaly detection, ii) location-based node authentication, and iii) symmetric key agreement at remote locations. By leveraging real-time sensing information, the framework incorporates jamming detection, AoA-based authentication, and symmetric key generation while enabling continuous trustworthiness evaluation of the physical layer. Physical-layer measurements are mapped into dynamic trust metrics that quantify link reliability, signal consistency, and resilience under interference or spoofing attempts. These metrics can be propagated to higher-layer orchestration functions, supporting adaptive access control, resource management, and RAN resource allocation to ensure explicit security guarantees. Through this sensing-driven and trust-aware closed loop, the prototype illustrates how security in 6G can be embedded natively into the physical and sensing layers.

Prototype 5 demonstrates the integrated ROBUST-6G security platform by combining the functionalities developed across the other four prototypes into a unified system. Unlike the previous prototypes, which focus on specific layers of the ROBUST-6G architecture, Prototype 5 showcases how different security components collaborate to provide comprehensive 6G security services. The prototype integrates security capability exposure and service requests through the NetSecaaS Gateway (Prototype 3), security orchestration and automation via the Multi-Layer Zero-Touch Defender (Prototype 2), AI-based security functions from Prototype 1 via the Global Model Repository, and PHY layer security services from the Physical and Sensing Layer Trustworthiness framework (Prototype 4). Furthermore, the prototype demonstrates the interoperability and composability of these functionalities through a common Security Ontology and Exposure Framework, enabling coordinated orchestration of security services within a coherent platform.

2.7 Unified Testbed Configuration

The Unified ROBUST-6G Testbed is the federated validation environment on which the platform is exercised end to end. Rather than building a single, monolithic laboratory, the consortium interconnects the Partner Testbed Assets (as described in D6.1) that already exist on partner premises, with each partner contributing the hardware, data, or software environment best suited to the

components it hosts. The federation is realised through the three mechanisms established in D6.1 and D6.2: secure inter-partner connectivity, containerised and orchestrated deployment, and the exposure of each component through a well-defined interface. Components hosted at different partners therefore interact as if they were co-located, while each site retains local control over its own infrastructure and data. The final configuration is shown in Figure . Its upper half is the federated core that is operated as one system by the Master Prototype (Prototype 5); its lower half is the wider set of partner testbed assets that support standalone, component-level validation. The two parts are complementary: the federated core proves that heterogeneous capabilities interoperate end to end, while the standalone assets validate individual components where the required hardware or data resides.

Federated configuration - in the federated configuration, four of the five prototypes are deployed on four different partner testbeds and bound together into a single security platform, with a fifth partner providing an external service consumed by the orchestrator. This cross-site binding is what makes scenario-level validation possible. A model trained and published at UMU is consumed by the orchestrator at NXW, exposed by northbound interfaces, and used to drive a mitigation that is actuated at ENSEA, all within one continuous control loop. The validation of a scenario is therefore not confined to any single site: it follows the request and the data across partner boundaries, which is the property the unified testbed is designed to demonstrate.

Other partner testbed assets - beyond the federated core, the consortium maintains the wider set of PTAs catalogued in D6.1, shown in the lower half of Figure . These assets support standalone, component/flow-level validation on the partner site where the relevant hardware or data resides, and are not part of the integrated configuration. Each validates a specific component and its associated KPIs locally, and the resulting measurements feed the use-case KPI attainment reported in Chapter 5 directly, without requiring the full federated stack to be exercised.

This division follows the federation principle adopted in D6.1, which maps each component to the testbed asset that satisfies its requirements rather than replicating every component across every site.

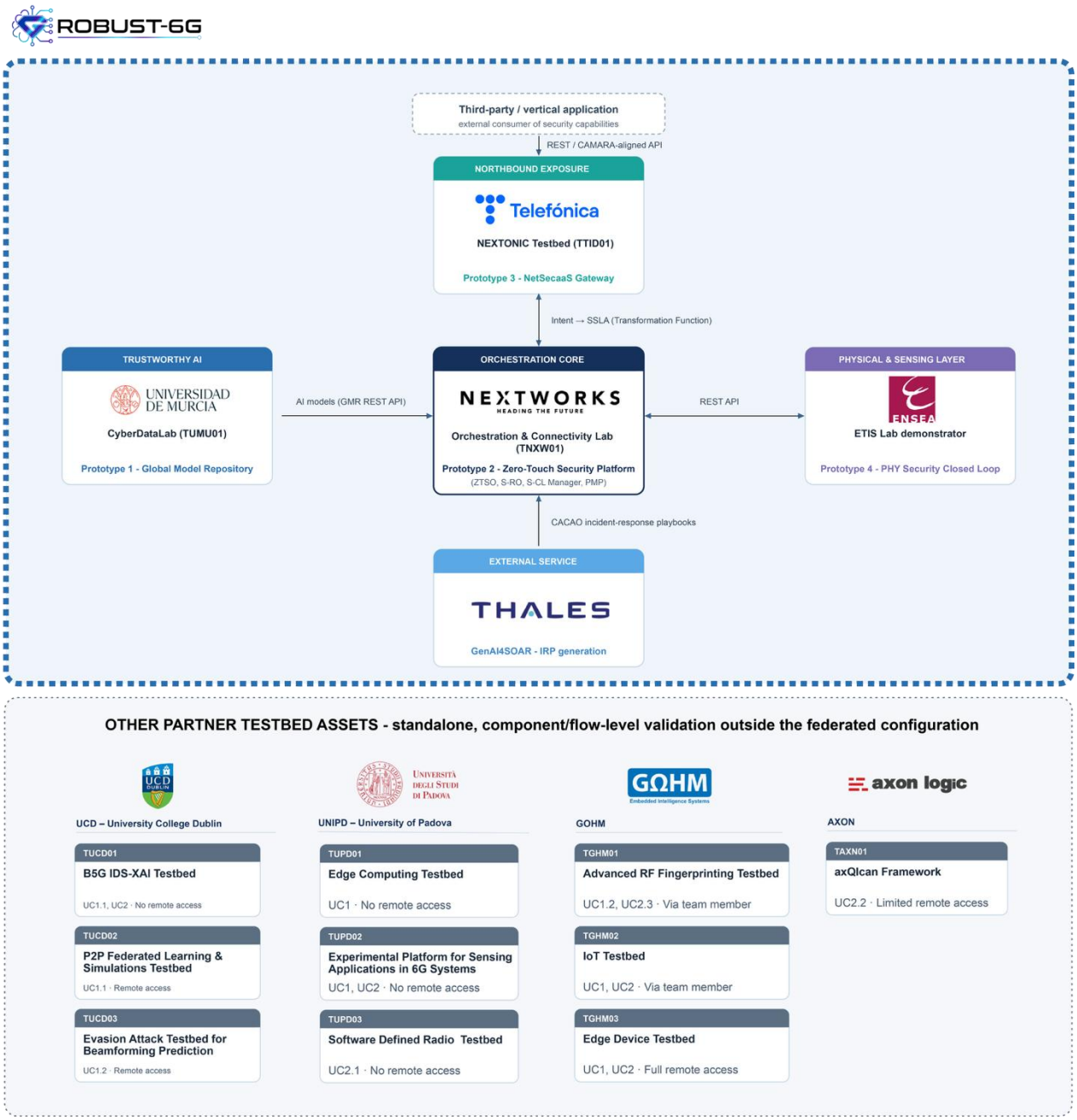

Figure 2.2 Unified ROBUST-6G testbed and Partner Testbed Assets

3 Prototypes

This chapter presents detailed descriptions of the five prototypes demonstrated in ROBUST-6G. For each prototype, an overview and the corresponding demonstration objectives are first introduced, followed by a discussion of its role within the overall ROBUST-6G architecture. The composition and architectural design of each prototype are then discussed in detail.

3.1 Prototype 1: Trustworthy AI

3.1.1 Overview and Demonstration Objectives

The ROBUST-6G Decentralised Federated Learning (DFL) Framework serves as a comprehensive tool for privacy-preserving Machine Learning (ML) and Deep Learning (DL) model training. By

adhering to the principle of "moving the compute to the data," the DFL Framework enables collaborative model training across highly decentralised edge networks without ever exposing or exchanging raw local data telemetry. This fully federated approach ensures strict compliance with the data privacy and security requirements established in previous architectural deliverables.

However, deploying federated systems in untrusted edge environments introduces severe vulnerabilities. Malicious actors or compromised edge nodes can execute Model Poisoning attacks, injecting maliciously crafted gradients to degrade the global model's performance, introduce targeted backdoors, or cause catastrophic forgetting. Standard aggregation techniques like Federated Averaging (FedAvg) are highly susceptible to these attacks, as even a single extreme malicious update can corrupt the entire global model.

To maintain the operational integrity of the learning process, the framework abandons vulnerable averaging methods and employs advanced Byzantine-robust defence mechanisms. Chief among these is the deep integration of the Krum aggregation algorithm [BEG+17]. Krum acts as the primary computational shield against adversarial attacks. By rigorously evaluating the high-dimensional spatial distribution of incoming model updates, Krum effectively isolates and filters out poisoned gradients, ensuring that the aggregated global model remains robust, accurate, and untainted by Sybil nodes or compromised participants.

Core objective of Prototype 1: The demonstration of this prototype aims to showcase a complete, end-to-end decentralised training lifecycle within a hostile environment. It highlights the system's ability to not only train complex AI models collaboratively but also to actively and autonomously defend against targeted adversarial attacks during the training phase, successfully yielding a certified trustworthy final model.

Furthermore, the Global Model Repository (GMR) serves as the central anchor of trust and the single source of truth for this prototype. It functions as a secure, immutable registry responsible for storing, versioning, and distributing these certified trustworthy models. Ultimately, the GMR enables critical model reuse capabilities, ensuring that the resulting AI models are reliable, transparent, reusable, and deployment-ready for operational 6G environments.

3.1.2 Position within the Architecture

The Decentralised Federated Learning (DFL) Framework and the Global Model Repository (GMR) are positioned within the Trustworthy and Sustainable AI Services Layer of the ROBUST-6G architecture. As defined in Section 3.1.1 of D6.2, we show here in Figure 3.1 the same architectural picture shown in that deliverable, in order to highlight that these components operate across specific architectural elements (marked A, B, C, D, E, F in the architectural mapping of Figure 3.1) to provide end-to-end trustworthy intelligence:

- **Robustness Service (A):** Provides tools for evaluating and enhancing model resilience against adversarial threats.
- **DFL Framework and Privacy-enhanced Services (B, E):** Act as the core orchestration engine, managing the federation lifecycle and enforcing privacy-preserving training protocols.
- **XAI Services (C):** Deliver advanced explainability analysis, generating trustworthiness artefacts like feature attribution stability and latent space visualisations.
- **Sustainable AI Service (D):** Responsible for tracking and optimizing the energy consumption and resource footprint of the AI lifecycle.
- **Global Model Repository (F):** Functions as the logical registry for the secure storage of versioned models, metrics, and resource logs.

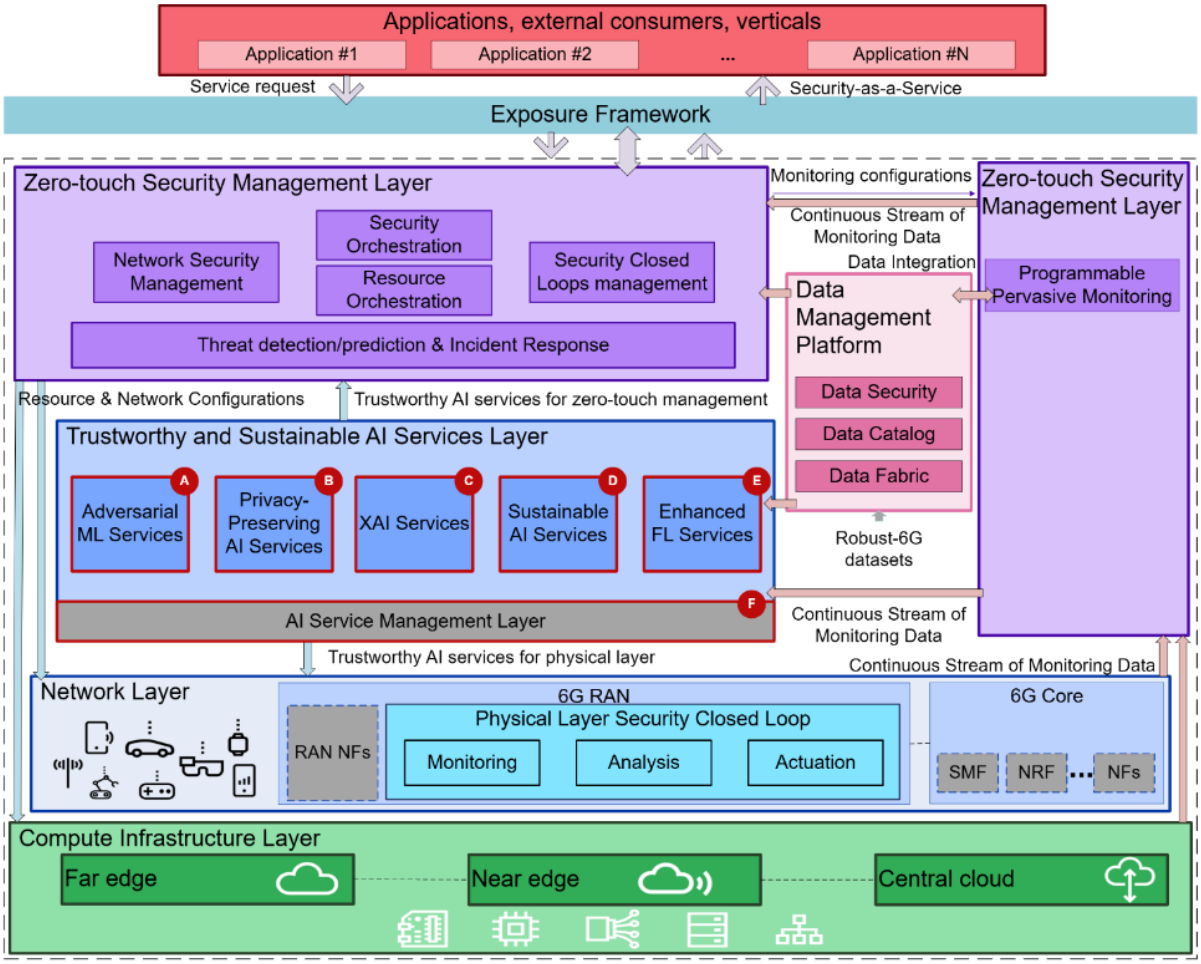

Figure 3.1 Architecture mapping of UC1_1

3.1.3 Prototype Composition and Architecture

The DFL Framework is structured as a containerised system that integrates several functional services to enable decentralised model training and evaluation. These components are organised and connected to achieve the prototype's objectives of privacy, robustness, and sustainability, as illustrated in the architectural mapping in Figure 3.1.

3.1.3.1 Architectural mapping and functional flows

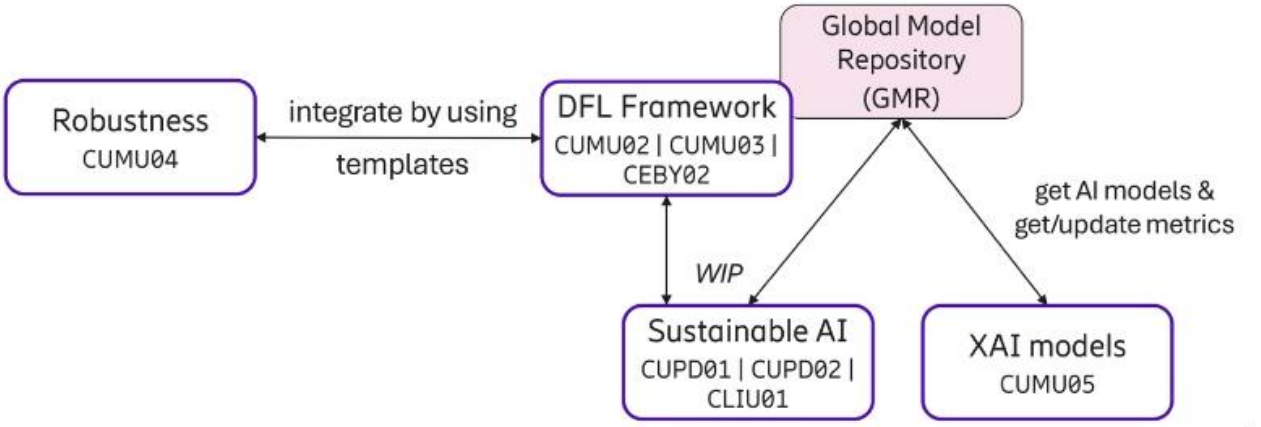

Figure 3.2 Architectural diagram of the components of Prototype 1

- The interactions between the Trustworthy AI prototype and the rest of the ROBUST-6G ecosystem are driven by a highly integrated, on-demand workflow exposed via standardised RESTful API interfaces.
- **Model Request & Trigger:** The lifecycle begins when Model Consumers request a specific threat detection or network optimisation model via the GMR API.
- **Decentralised Training:** If the requested model is not already available in the repository, a "Training Trigger" is fired. This initiates a new, collaborative training round within the **Decentralised Federated Learning Framework (CUMU02)**.
- **Module Execution & Cross-Prototype Synergy:** During training, the **DFL Framework (CUMU02)** leverages its internal modules to ensure the resulting model is secure, transparent, and robust. This process deeply integrates with other ROBUST-6G prototypes:
    - It utilises the Krum algorithm (associated with the **Robustness Service CUMU04**) as the primary trust-aware aggregation mechanism to detect and mitigate adversarial threats by filtering out malicious model updates during the training rounds.
    - It exchanges real-time performance metrics (CPU/GPU usage, network traffic) with the **Sustainable AI Service (CUPD01, CUPD02, CLIU01)** to track and optimise the energy consumption and resource footprint of the DFL process.
    - It generates feature attribution and explainability artefacts (XAI images) at the conclusion of each training round via the **XAI Services (CUMU05)** to ensure full transparency of the model's learning trajectory.
    - It utilises **Privacy-enhanced Services (CEBY02)** to safeguard the distributed learning process from data leakage and poisoning attacks.
- **Registry & Distribution:** Once the DFL-trained models are aggregated and finalized, the DFL Framework populates the Global Model Repository (GMR) with the versioned models, XAI images, metrics, and resource logs. The GMR then serves as the central distributor, delivering the final model artefacts and their associated explainability reports back to the original Model Consumers. This creates a powerful closed-loop system in which the Trustworthy AI layer trains and secures the intelligence, while other partners seamlessly operationalise it.

3.1.3.2 Core Framework Components

DFL Framework (CUMU02): The core orchestration engine and federation node management system. It provides a unified environment for Decentralised Federated Learning, handling the federation lifecycle, network topology enforcement, and the execution of distributed model aggregation algorithms. It is the primary entry point for external training requests and manages the interaction between local training and the Global Model Repository.

Robustness Service (CUMU04): Responsible for enhancing the resilience of the AI models against adversarial threats. In this prototype, the **Krum algorithm** is implemented in the aggregation logic to filter out malicious model updates during training and ensure global model integrity.

Sustainable AI Service (CUPD01, CUPD02, CLIU01): Focuses on the efficiency and resource footprint of the AI lifecycle. It utilises **ADMM (CUPD01)** to optimise model training efficiency and **Spiking Neural Networks (CUPD02)** for energy-conscious learning. It tracks and analyses real-time metrics (CPU/GPU usage, network traffic) from the federation nodes to provide insights into the sustainability of the DFL process.

XAI Services (CUMU05): Provide advanced explainability analysis. It generates feature attribution artefacts **at the conclusion of each training round**. These artefacts ensure transparency by documenting the model's evolutionary history before the results are deposited into the GMR.

Privacy-enhanced Service (CEBY02): Integrates privacy-preserving technologies to safeguard the decentralised learning process from data leakage. It ensures that only model parameter updates, never raw data, are exchanged among participants, maintaining strict data sovereignty.

Global Model Repository (GMR) (CUCWP03): Acts as the centralised registry and secure storage for all finalised artefacts. It stores versioned ML/DL models (packaged via BentoML) [x], performance metrics, and the associated XAI images. It facilitates model reuse by serving as a passive receiver for completed models and a distributor for authorized Model Consumers.

Frontend Dashboard: A user interface built with **Dash/Plotly** for real-time monitoring. It parses local Comma-Seperated Values (CSV) telemetry (CPU, RAM, Bytes sent) and visualizes the training progress and XAI plots generated by the nodes.

P2P Communications Protocol: A robust peer-to-peer protocol based on **epidemic gossip algorithms**. It eliminates single points of failure by facilitating the secure, asynchronous exchange of model weights between decentralised Docker nodes.

3.1.3.3 Deep Dive: Global Model Repository (GMR)

The Global Model Repository acts as a vital structural bridge between the edge and the core network. While the DFL Framework intentionally decentralises the training and aggregation processes to enhance privacy and eliminate single points of failure, the resulting models must ultimately be accessible to core network orchestrators. The GMR solves this operational gap by serving as a secure, centralised registry (F). It links the fully autonomous, distributed training nodes, which process localised data, with the centralised security management components that need to consume and deploy the final AI models.

It functions as the single source of truth for the entire ecosystem. Rather than having orchestration layers attempt to pull models directly from the federation nodes, the distributed nodes themselves push their final models into the GMR. The GMR securely stores these models alongside the hyperparameter configurations, explainability data, and performance metrics generated during the decentralised training phase. By acting as a passive receiver, it creates a highly organised, queryable repository of trustworthy intelligence.

The GMR's internal architecture, primarily driven by robust_gmr/server.py [CDL-DFL, exposes a comprehensive suite of secure REST API endpoints (e.g., POST /api/robust/upload/model, POST /api/robust/upload/metrics). It serves as the primary data source for XAI Services and enables post-auditing of the entire training history.

Models are automatically packaged with BentoML for standardised serving, while underlying PostgreSQL operations are handled via AsyncPG [AsyncPG] to ensure model provenance. The database scheme is designed to track:

- **training_sessions & rounds:** Linking specific training epochs to hyperparameter configurations and federation topologies.
- **node_registry:** Storing the identities, health status, and historical behaviour of all contributing edge nodes.
- **performance_metrics:** Historical logs of model accuracy, resilience drops under simulated attacks, and data distribution parameters.
- **model_artefacts:** Pointers to binary weight files, cryptographic hashes (SHA-256), and the associated XAI images generated during round finalization.

3.1.3.4 Krum Integration for Byzantine Resilience

To ensure Byzantine resilience during model updates, the Krum algorithm is integrated directly into the core P2P aggregation logic. Traditional Federated Averaging is highly susceptible to model poisoning, where a single malicious update can corrupt the global state.

Krum mitigates this by applying a spatial distance filter. For each received update, it computes the squared Euclidean distance to all other updates. It then calculates a score based on the distances to the closest neighbours and selects the update with the lowest score as the representative benign update for the round. This effectively discards anomalous or poisoned gradients, allowing the federation to converge even in the presence of malicious participants.

3.1.3.5 Integration and Modularity

PyTorch and Lightning Integration: The framework leverages PyTorch and PyTorch Lightning as its foundational ML engines. This architecture abstracts the complexity of neural network models (such as Spiking Neural Network (SNN), CyberNet, and Multilayer Perceptron), providing optimised forward/backward propagation and high-performance tensor manipulation that automatically utilises local hardware acceleration.

Extensible Design Paradigm: The system architecture is guided by a modular design pattern that separates concerns among training orchestration, adversarial defence, and model execution. This modularity is implemented through standardised internal interfaces and factory patterns, enabling the framework to serve as a flexible research testbed for ROBUST-6G. Researchers can independently inject new adversarial attack vectors, evaluate diverse aggregation algorithms, and incorporate specialized neural architectures without modifying the core peer-to-peer synchronization logic. This design ensures that the prototype is future-proof and adaptable to the evolving trustworthiness and sustainability requirements of next-generation 6G environments.

3.2 Prototype 2: Multi-Layer Zero-Touch Defender

The Multi-Layer Zero-Touch Defender operationalises the autonomous decision-making, orchestration, and automated remediation capabilities of ROBUST-6G into an integrated Zero-Touch Security Platform (ZTSP). Depicted in Figure 3.3, the evolution of the ZTSP has been extensively detailed in the WP4 Deliverables [R6G24-D41] [R6G25-D43] [R6G26-D44].

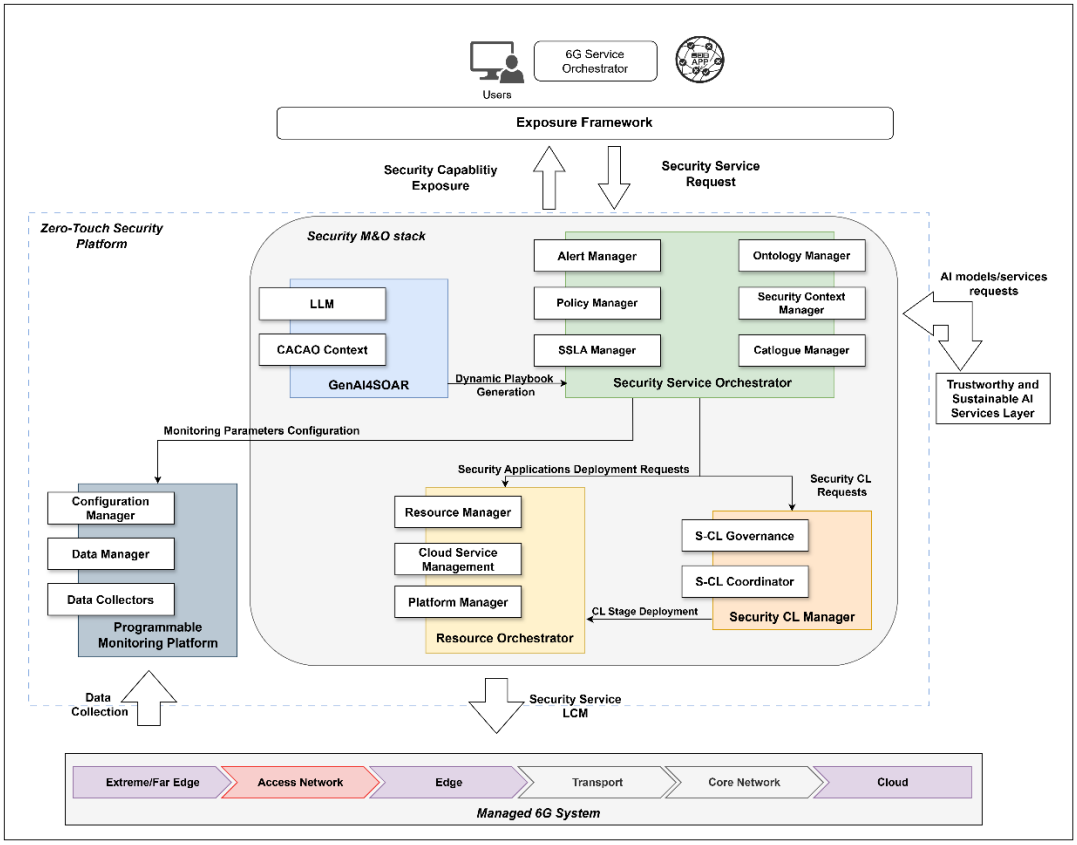

Figure 3.3: ROBUST-6G Zero-Touch Security Platform

Designed to address complex vulnerabilities across distributed next-generation architectures, its primary purpose is to compose fully automated Security Services capable of establishing and maintaining a Security posture with zero human intervention. In the ZTSP, the concept of Security Service is an umbrella that encapsulates two other baseline concepts:

- **Security Function** (SF): any software, tool, application, configuration, actuator that is able to perform one or more Security Activities defined in the Zero-Touch Security Orchestrator (ZTSO) Ontology.
- **Security Closed Loop** (S-CL): a multi-stage automation that is capable of remediating to Security Threats autonomously by using one or more Security Functions.

Demonstrating the ZTSP in action, this prototype shifts security engineering away from static, human-operated mitigation towards an AI-driven, multi-layer defence engine, where semantic reasoning and GenAI assist in composing each Security Service from one or more SFs and S-CL, as detailed in D4.4 [R6G26-D44].

3.2.1 Overview and Demonstration Objectives

The demonstration objectives for this prototype can be summarized as follows. These objectives are validated in Section 5.1.2 in the context of Use Case 2 Scenario 1, defined in Section 4.2.1.

- SSLA Ingestion, Validation, and Parsing: demonstrate that the ZTSP (in particular, the Policy Manager Module of the ZTSO) is capable of ingesting, validating, and parsing SSLAs in accordance with the activities that are available in the ZTSO Ontology.
- Semantic-driven Security Functions selection: demonstrate that the ZTSP (in particular, the Ontology Manager, Catalogue Manager, and Security Context Manager of the ZTSO) is capable of driving Security Function selection based on the available Security Functions, Target Environments, and the required Security activities.

- Generation of Incident Response Playbooks (IRPs) through GenAI: demonstrate that the ZTSP (in particular, the GenAI4SOAR component and the GenAI Gateway component of the ZTSO) is able to produce CACAO v2 [OAS23a] and OpenC2 [OAS19a] compliant IRPs starting from the target infrastructure and available Security Functions context.
- Dynamically reconfigurable and pervasive Monitoring: demonstrate that the ZTSP (in particular, the Programmable Monitoring Platform component) can be dynamically re-configured to monitor multiple sources and provide insightful traces and alerts.
- Management and Orchestration of Security Services: demonstrate that the ZTSP (in particular, the Secure Resource Orchestrator, the Security Closed Loop Management, and the Security Context Manager of the ZTSO) are able to orchestrate and manage complex Security Services composed of SFs and S-CLs.

3.2.2 Position within the Architecture

As described in previous sections, the components of Prototype 2 are the ones of the ROBUST-6G Zero Touch Management Layer implemented by the ZTSP. Figure 3.4 depicts the mapping of prototype 2 with the functionalities of the ROBUST-6G architecture, in particular:

- The Network Security Management, Security Orchestration, Resource Orchestration, Security Closed Loops Management functionalities are provided by the different components of the ZTSP, specifically the ZTSO, Secure Resource Orchestrator (S-RO), S-CL Manager and the GenAI4SOAR.
- The Threat Detection/Prediction & Incident response functionalities are provided by the different Security Functions developed in WP4 [R6G26-D44].. In particular, AI Models for threat detection and prediction represent the analytics part while the OpenC2 actuators and CACAO Playbook management components represent the Incident Response part.
- The Programmable Pervasive Monitoring functionalities are provided by the ZTSP, in particular by the Programmable Monitoring Platform.

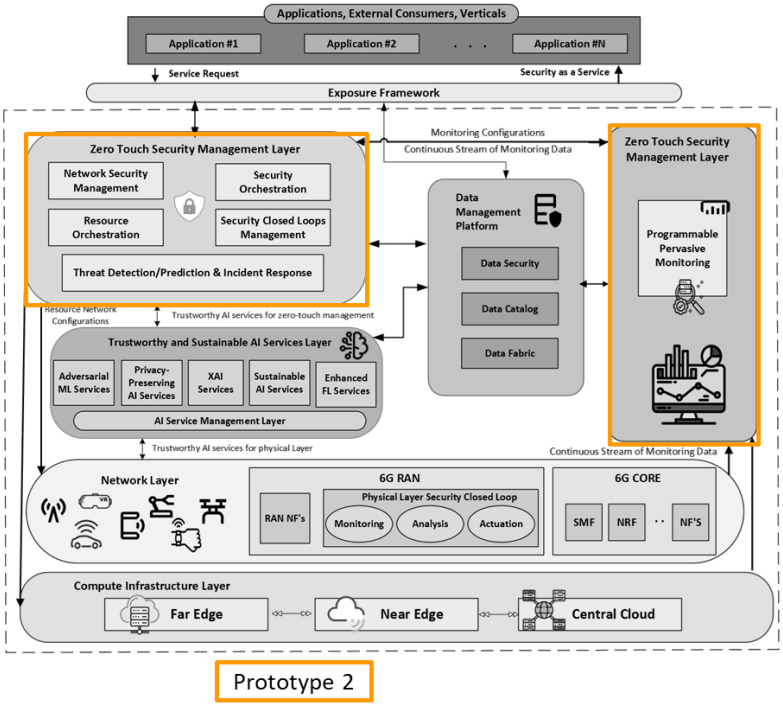

Figure 3.4: Prototype 2 Positioning in the ROBUST-6G Architecture

3.2.3 Prototype Composition and Architecture

Figure 3.5 provides a high-level architecture of Prototype 2, detailing the Security Service composition and orchestration capabilities of the ZTSP. The platform ingests a Security Service Request via an SSLA and dynamically composes a Security Service using:

- **Security Functions (SFs):** Including the Programmable Monitoring Platform (PMP), which acts as an SF to provide advanced monitoring and analysis capabilities.
- **Security Closed Loop (S-CL):** Which interacts with the SFs to provide zero-touch automation for security posture management.

As shown in the figure, the Security Service follows a strict lifecycle:

1. **Monitoring:** the Managed Entity (the target infrastructure) is monitored by the PMP.
2. **Analysis:** the monitoring data is processed through analytics algorithms (e.g., Snort3, AI-based IDSs).
3. **Decision and Execution:** automated responses are driven by CACAO IRP playbooks and actuated by dedicated OpenC2 actuators.

As detailed in D4.4 [R6G26-D44], these Security Services are highly composable. The S-CL can be dynamically adapted to include one to five stages. For example, the PMP can provide raw network traces for external AI analysis, or it can output direct alerts via its internal SNORT module. In the latter case, the S-CL is streamlined to utilise only the decision and execution stages.

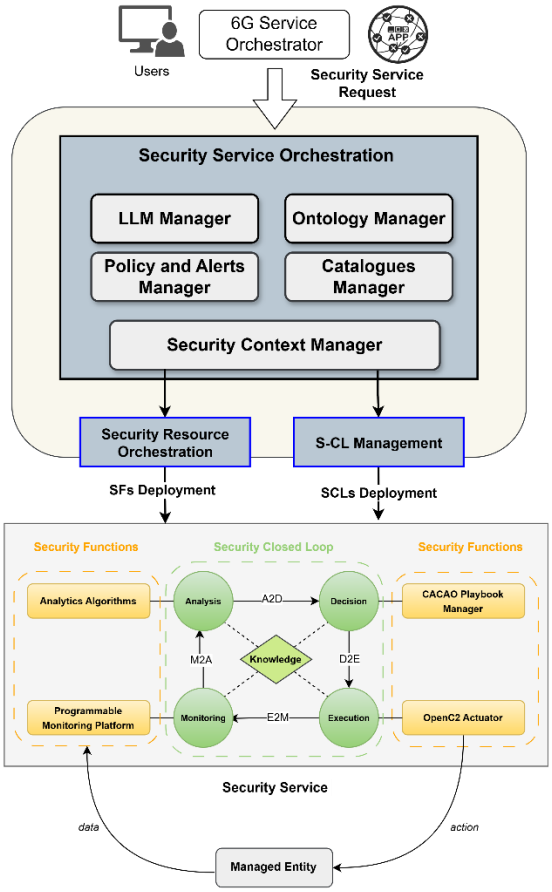

Figure 3.5: Prototype 2 High Level Architecture

3.3 Prototype 3: NetSecaaS Gateway

3.3.1 Overview and Demonstration Objectives

The NetSecaaS Gateway operationalises the Exposure Framework of the ROBUST-6G architecture as a demonstrable system. Its purpose is to make 6G security capabilities, such as trustworthy AI services, zero-touch orchestration, data governance and physical-layer security, consumable by external stakeholders through a coherent set of intent-oriented, developer-friendly APIs. By doing so, it reflects the fundamental principle that 6G is moving towards, namely that security should be offered as a service rather than imposed as an opaque operational requirement. This allows application developers, enterprises and vertical providers to implement security policies and utilise security insights without needing to understand the underlying network or AI internals.

Design-wise, NetSecaaS takes inspiration from the GSM Association (GSMA) Open Gateway initiative and the CAMARA project, both of which define a results-oriented, Open Authorization-protected (OAuth-protected) approach to API exposition over telco assets. Prototype 3 adopts this approach and builds upon it to encompass 6G-specific security semantics that are not yet addressed by these standardisation bodies, specifically explainable AI outputs and Security Service Level Agreements (SSLAs). Consequently, the prototype is positioned as both a consumer of established exposure patterns and a potential contributor to future work items on security exposure by CAMARA, GSMA Open Gateway and 3GPP.

The objectives of the prototype demonstration can be summarised as follows:

- Demonstrate a working end-to-end exposure path from a high-level request issued by a third party to the invocation of internal ROBUST-6G capabilities, with all interactions mediated by intent-based REST APIs in the CAMARA style.
- Validate two complementary interaction patterns: retrieval of security-relevant data (e.g. analytics, explainability artefacts, capability catalogues) and configuration or triggering of security actions (e.g. closed-loop reactions, SSLA enforcement, policy provisioning).
- Show that the NetSecaaS Gateway transparently provides access to several ROBUST-6G capabilities, such as trustworthy AI services from Prototype 1, the multi-layer zero-touch defender from Prototype 2 and the physical-layer trustworthiness mechanisms from Prototype 4, via a single, uniform northbound interface.
- Demonstrate the abstraction provided by the Transformation Function, which translates concise REST calls into heterogeneous internal SSLA queries that map to orchestration requests expected by the underlying components, thus abstracting their technical complexity from consumers.
- Ensure that every exposed interaction is governed end-to-end by authentication, authorisation and policy-based access control mechanisms, which are enforced by the Data Governance plane. This ensures auditability and compliance with the data owner permissions defined in the ROBUST-6G Data Management Platform.

3.3.2 Position within the Architecture

Within the ROBUST-6G reference architecture introduced in Figure 2.1, the NetSecaaS Gateway occupies the outside boundary of the system, as it is the concrete instantiation of the Exposure Framework layer. From this position, it facilitates every interaction between external consumers — such as vertical applications, enterprise tenants and third-party service providers — and the internal ROBUST-6G ecosystem, which comprises the Data Management Platform, the Zero-Touch Security Management Layer and, consequently, the Trustworthy and Sustainable AI Services Layer and the Physical Layer Security closed loop.

The Prototype 3 implements two complementary interaction directions it supports. The first is the retrieval direction, illustrated in Figure 3.6, in which a third-party requests information about or evidence of internal security capabilities. Examples include discovering available security capabilities, consulting access-governance metadata and retrieving explainability reports for AI/ML-driven security events. In this pattern, the third party invokes the NetSecaaS endpoint (CTID03), which delegates authorisation to the Data Management Platform — specifically to its Data Governance plane (CTID02) — and once authorisation is granted, the requested artefacts are retrieved from the Data Fabric (CTID01). The outputs of the underlying ROBUST-6G security capabilities (denoted CRG00) have previously been ingested and semantically integrated here.

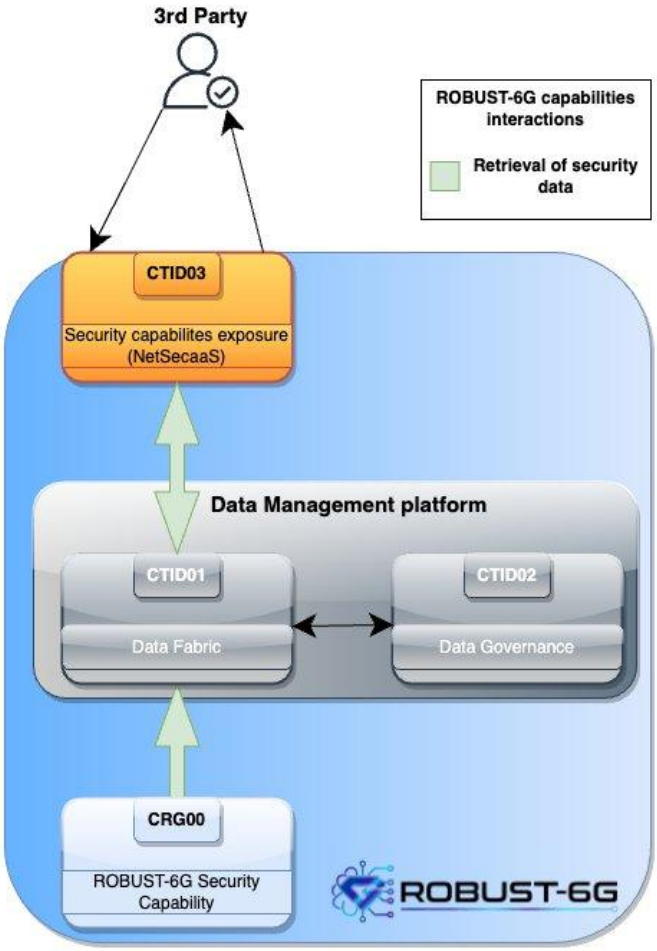

Figure 3.6 Generalised data retrieval flow Prototype 3

The second direction involves configuring and triggering security actions, as shown in Figure 3.7. In this case, the third party expresses a high-level security intent, such as activating a specific protection mechanism, enforcing a simplified SSLA that can trigger a closed-loop remediation. The NetSecaaS Gateway then authenticates and authorises the request, transforming the intent into a structured SSLA or policy artefact before forwarding it to the Security Orchestrator (CTHL01) within the Zero-Touch Security Management Layer. This component interacts directly with the Zero-Touch Security Orchestration components (CNXW01). The orchestrator then executes the corresponding actions on the relevant ROBUST-6G security capability and propagates the enforcement status back through the gateway to the requester.

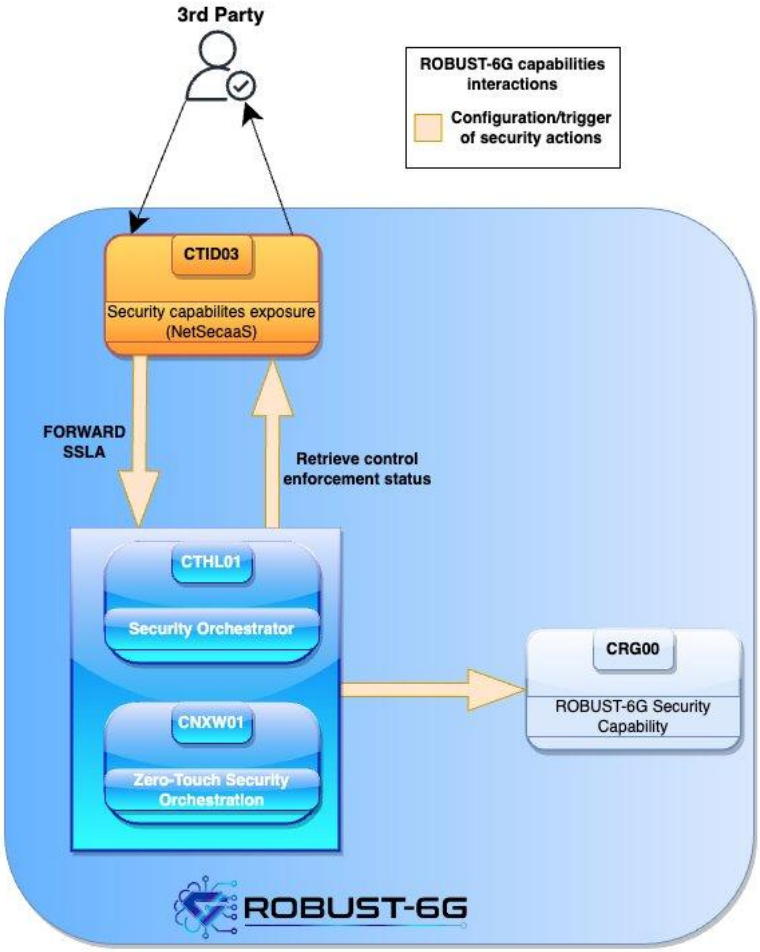

Figure 3.7: Generalised configuration and trigger flow Prototype 3

Together, these two patterns establish the NetSecaaS Gateway as the structural bridge between the external environment and the internal ROBUST-6G stack. In relation to the other prototypes, this positioning has direct operational consequences. Firstly, some Trustworthy AI artefacts produced by Prototype 1, such as explainability reports, can be accessed via the retrieval endpoints provided by NetSecaaS. The Multi-Layer Zero-Touch Defender of Prototype 2 can be accessed from outside via SSLA-style configuration endpoints, while Prototype 3 acts as the front door to the Security Orchestrator. The physical-layer security services delivered by Prototype 4 can also be advertised in the capability catalogue and triggered via CAMARA-style APIs. In the Master Prototype (Prototype 5), the NetSecaaS Gateway serves as the entry point for validating the integrated ROBUST-6G system as a whole.

3.3.3 Prototype Composition and Architecture

The internal composition of Prototype 3 follows a layered design that separates the externally visible interface from the components that turn each call into concrete actions on the ROBUST-6G infrastructure. Figure 3.8 provides an overview of this composition and of the way the prototype is wired to the rest of the platform.

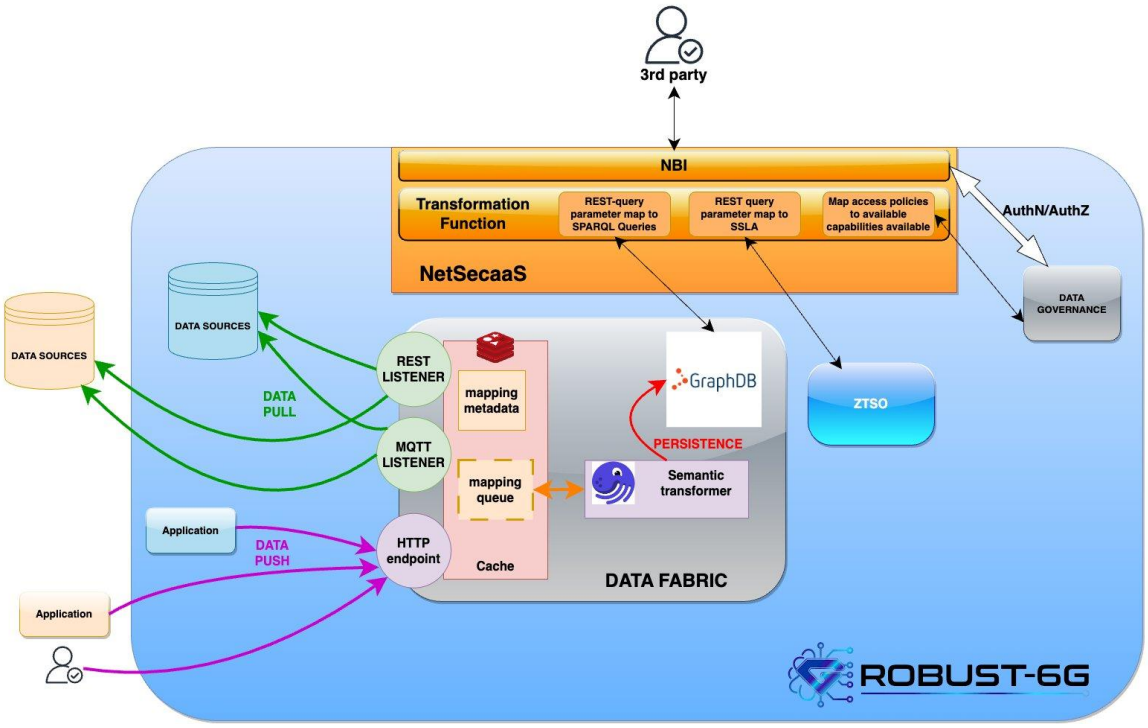

Figure 3.8:NetSecaaS Gateway — general architecture and integration with the ROBUST-6G platform

The Northbound Interface (NBI) is the interface exposed to third parties. Designed in accordance with CAMARA principles, it is implemented as a set of REST endpoints, each named after the outcome the consumer is trying to achieve. Authentication is OAuth-based. The Transformation Function sits behind the NBI and is responsible for translating the abstract intent expressed in a REST call into the concrete operations that the rest of the platform can execute. As shown in Figure 3.8, the Transformation Function performs three complementary types of translation. It maps REST query parameters into SPARQL Protocol and RDF Query Language (SPARQL) queries against the Data Fabric Knowledge Graphs for data retrieval requests. It maps REST request payloads into structured SSLA artefacts for the configuration of security service requests. It also maps the requester's credentials and the requested resource against the access policies provisioned in the Data Governance plane. This ensures that only the capabilities to which the consumer is entitled are made available.

This separation between interface and transformation is what makes the prototype extensible: new ROBUST-6G capabilities can be exposed by adding new endpoints and the corresponding transformation rules, without modifying the consuming applications. It also enables the Gateway to enforce a uniform security model — authentication, authorisation, and policy enforcement — across heterogeneous internal components that would otherwise present widely different interfaces.

The NetSecaaS Gateway does not retrieve raw data directly from each ROBUST-6G capability. Instead, it relies on the Data Management Platform, which acts as the consolidated, governed substrate of all data exchanged between internal capabilities and external consumers. Two cooperating planes compose this substrate. The Data Fabric ingests, normalises, semantically integrates, and exposes data through a Knowledge Graph backed by a triple store (GraphDB); it is the source against which the Transformation Function's SPARQL queries are issued. The Data Governance plane stores the access policies, authenticates the requester through the platform's Identity Provider, and evaluates each request against the policies defined by data owners; it is the authoritative source for the authentication and authorisation (AuthN/AuthZ) decisions performed by the Gateway.

This design has two important consequences for Prototype 3. First, every exposed capability shares a common, auditable access-control mechanism, so that the security guarantees offered to third parties are uniform regardless of which internal component ultimately serves the request. Second, the semantic integration performed by the Data Fabric isolates the Gateway from changes in the data formats produced by individual capabilities: as long as a capability registers its outputs in the Knowledge Graph under the agreed ontology, no modification of the exposure layer is required to make those outputs consumable through the NBI.

A central element of Prototype 3 is the set of data pipelines that feed the Data Fabric with the outputs of the ROBUST-6G security capabilities. These pipelines have been designed to be generic and reusable across heterogeneous data sources, supporting both push and pull integration patterns. Figure 3.9 summarises their structure.

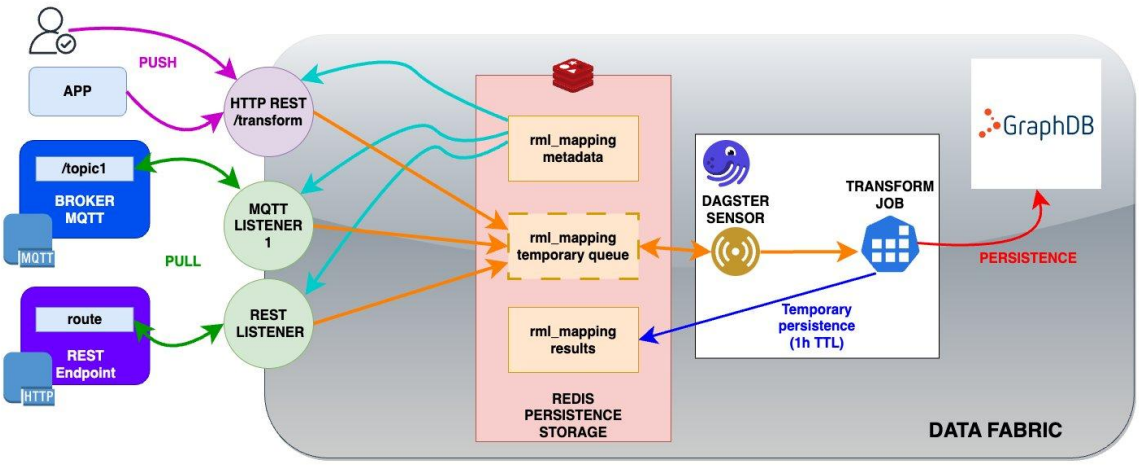

Figure 3.9:Generic data ingestion pipelines feeding the Data Fabric of the NetSecaaS Gateway

Three entry points are offered to data producers. The first is an HTTP REST endpoint that accepts push requests issued explicitly by an application or by a user, typically when the producer wants to deliver a self-contained payload on demand. The second is an Message Queuing Telemetry Transport (MQTT) listener that subscribes to one or more broker topics and is therefore suitable for streaming telemetry and continuous event flows. The third is a REST listener that pulls data from existing REST endpoints exposed by the producers, supporting integration with components that already provide their own HTTP interfaces. The choice between push and pull is deliberately offered at the design stage so that the Gateway can absorb the diverse styles of interaction characteristic of the heterogeneous components present in the ROBUST-6G ecosystem.

Once received, all incoming payloads converge on a Redis[2]-backed persistence layer that decouples the ingestion endpoints from the downstream transformation logic. Redis maintains two cooperating structures: a metadata store that holds the Relational Markup Language (RML) mapping definitions associated with each data source, and a temporary queue that holds the payloads awaiting transformation. A Dagster[3] sensor monitors this queue and, upon the arrival of a new payload, schedules a transformation job that applies the corresponding RML mapping to lift the raw data into Resource Description Framework (RDF) triples aligned with the platform's ontology. The resulting

[2] https://redis.io

[3] https://dagster.io

graph fragments are then persisted into the GraphDB triple store, where they become reachable through the same SPARQL endpoint the Transformation Function uses to serve retrieval requests.

This pipeline design carries two properties that are essential to the prototype's role within the broader platform. On the one hand, it is declarative: integrating a new data source requires defining a new RML mapping rather than writing custom code, which significantly reduces the integration effort associated with the diverse outputs produced by other ROBUST-6G capabilities. On the other hand, it is uniform: irrespective of whether the source pushes JSON over HTTP, publishes messages on an MQTT topic, or exposes a REST API, the data ultimately lands in the same Knowledge Graph and is governed by the same access policies, which is precisely the property that enables the NetSecaaS Gateway to offer a coherent, single-pane-of-glass exposure of the ROBUST-6G security capabilities.

While the data pipelines just described serve the retrieval interaction pattern, the configuration pattern relies on a direct integration between the Transformation Function and the Zero-Touch Security Orchestrator (ZTSO). When a third-party request expresses a high-level security intent, the Transformation Function maps the request parameters into a structured SSLA, and the resulting artefact is forwarded to the ZTSO for enforcement. The orchestrator interprets the SSLA, decomposes it into the concrete actions to be applied to the underlying capabilities, and reports the enforcement status back to the Gateway, which in turn returns a simplified, consumable response to the requester. This integration is what allows Prototype 3 to expose not only the data produced by ROBUST-6G but also the security behaviours that the platform is able to enact.

3.4 Prototype 4: Physical and Sensing Layer Trustworthiness

3.4.1 Overview and Demonstration Objectives

Prototype 4 demonstrates a physical-layer closed-loop security framework for 6G, with the primary objective of transforming the physical layer from a passive transmission medium into an active trust anchor. To this end, the prototype continuously monitors physical observations through sensing, analyses them in real time, and feeds the resulting indicators into a trust evaluation engine that adapts radio access network (RAN) resource allocation decisions accordingly. It leverages the sensing capabilities of 6G to extract spatial (angle-of-arrival, AoA) and channel-state features (channel state information, CSI, and signal-to-interference-plus-noise ratio, SINR), which serve as intrinsic indicators for: i) physical-layer (PHY) anomaly detection, ii) location-based node authentication, and iii) symmetric secret key agreement at remote locations.

Within the broader ROBUST-6G platform, the prototype is designed to demonstrate three interlinked capabilities. First, it showcases the integration of advanced physical-layer security mechanisms developed in Work Package 5 (WP5), namely jamming detection, AoA-based authentication, and symmetric key generation, into a single coherent demonstrator. Secondly, it shows how these mechanisms can be tested against realistic adversarial behaviour, namely jamming, spoofing, and eavesdropping attacks, in a 6G context. Thirdly, it illustrates how the resulting PHY measurements can be mapped into dynamic trust metrics that quantify link reliability, signal consistency, and resilience under interference or spoofing attempts, with a view to propagating these metrics to higher-layer orchestration functions to support adaptive access control, resource management, and RAN resource allocation with explicit security guarantees.

3.4.2 Position within the Architecture

Through this sensing-driven and trust-aware closed loop, Prototype 4 illustrates how security in 6G can be embedded natively into the physical and sensing layers, rather than being treated as an add-on at higher layers. Thus, it provides a concrete and reproducible vehicle for showcasing how the WP5 contributions to ROBUST-6G can be combined and exposed to a user through an interactive demonstrator that allows attack scenarios to be configured, executed, and observed in real time.

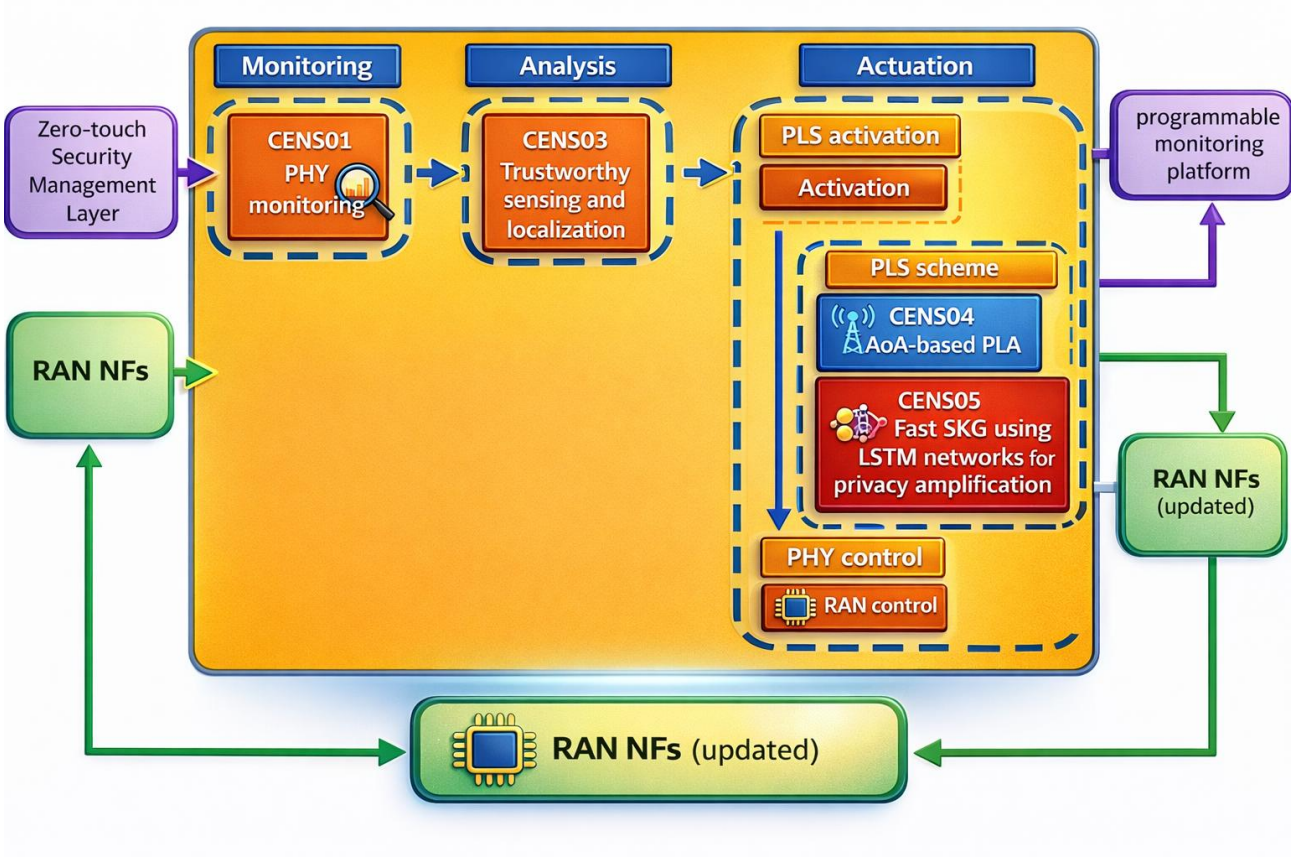

Figure 3.10 Architectural positioning of Prototype 4 within the Physical Layer Security Closed Loop

Prototype 4 is positioned within the Physical Layer Security Closed Loop (PLCL) of the 6G RAN, which belongs to the Network Layer of the ROBUST-6G reference architecture presented in Section 2.1. Within this architectural position, the prototype provides the physical-layer security and trustworthiness capabilities required to monitor radio conditions, analyse security-relevant PHY-layer information and activate suitable Physical Layer Security (PLS) schemes or PHY control responses. Figure 3.10 provides a detailed view of the architectural blocks activated by Prototype 4 within the PLCL. The prototype uses **CENS01 – PHY Monitoring** in the monitoring stage and **CENS03 – Trustworthy Sensing and Localization** in the analysis stage. The actuation stage includes the activation of the selected PLS schemes through **CENS04 – AoA-based PLA** and **CENS05 – Fast SKG using LSTM Networks for Privacy Amplification**, together with **RAN control** for PHY-level adaptation. In this way, the prototype maps the monitoring–analysis–actuation logic of the PLCL into a concrete physical-layer security demonstrator.

Additionally, the prototype is designed to interact with the **RAN Network Functions**, from which PHY-layer observations are obtained and to which updated PHY configurations may be provided following actuation. As also depicted in Figure 3.13, potential interfaces with the **Zero-Touch Security Management Layer** support the exchange of upper-layer security context and PHY-layer

security reports. These broader interactions are further discussed and demonstrated through **Prototype 5 – Master Prototype**.

3.4.3 Prototype Composition and Architecture

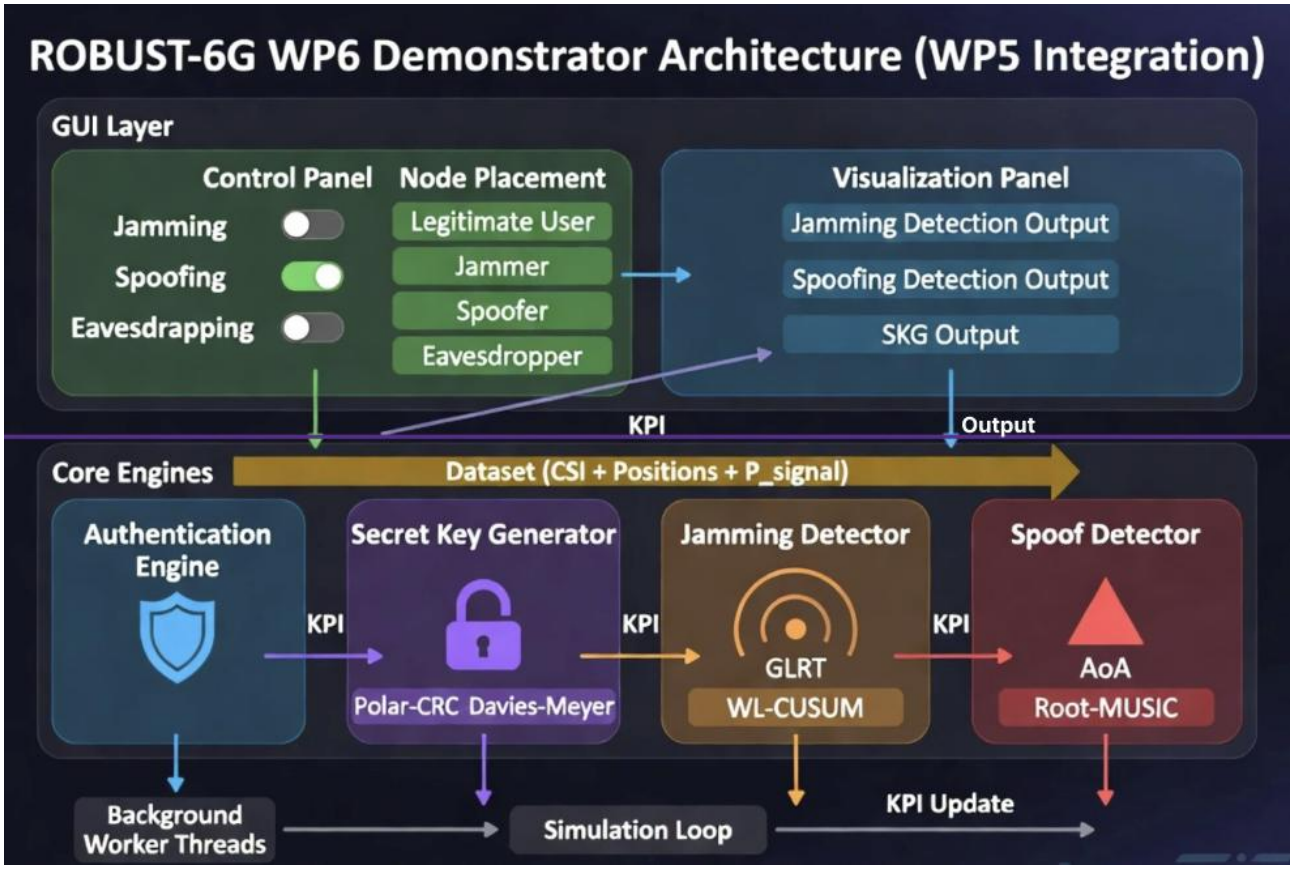

Figure 3.11 Prototype 4 Demonstrator architecture

Figure 3.12 Prototype 4 in operation mode

To realize the above objectives in a concrete and reproducible manner, the prototype, whose architecture is depicted in Figure 3.11, is implemented as a modular, graphical user interface (GUI)-based interactive demonstrator that integrates the WP5 physical-layer security mechanisms within a single tool. It leverages real-world measured CSI from a 24×24 grid dataset, enabling users to select attack scenarios, place virtual nodes on the grid, activate the selected attacks, and observe the system's detection and mitigation capabilities in real time. The prototype's architecture follows a layered design that cleanly separates the GUI, the data foundation, the core security engines, and the simulation orchestration, thus ensuring responsiveness and maintainability while prominently integrating the WP5 computational modules.

The top layer consists of the GUI, which comprises a Control Panel and a Visualisation Panel as in Figure 3.12. The Control Panel allows users to toggle the three attack types (jamming, spoofing, and eavesdropping), select and place nodes with specific roles (legitimate user, jammer, spoofer, eavesdropper), adjust the signal-to-noise ratio (SNR), and monitor key performance indicators (KPIs). The Visualisation Panel, on the other hand, displays the grid with the placed nodes and provides dedicated output sections for jamming detection results, spoofing detection maps, and secret key generation metrics, thus offering immediate visual feedback on system behaviour. Underpinning the prototype is the Data Layer, which relies on the dataset containing measured CSI and user equipment (UE) positions, supplemented by antenna array information. To ensure that every simulation utilises authentic measured channels, all node placements are automatically snapped to the nearest dataset locations. Moreover, a derived 24×24 signal power grid is computed once and reused by the jamming detection module to minimise redundant computation.

The Core Engines, primarily integrated from WP5, form the heart of the security functionality and constitute the computational realisation of the closed-loop trust framework. These include: i) the Authentication Engine for AoA-based user verification; ii) the Secret Key Generator (SKG Engine), which performs linear quantization, Polar-cyclic redundancy check (CRC) information reconciliation, and Davies–Meyer/AES-128 key derivation; iii) the generalized likelihood ratio test (GLRT)-based Jamming Detector, which combines a spatial GLRT with windowed limited cumulative sum (WL-CUSUM) for temporal confirmation; and iv) the AoA-based Spoof Detector, which employs Root-MUSIC estimation in conjunction with calibration and mitigation techniques.

Orchestration is handled by the main simulation loop in conjunction with background worker threads. When the user starts a simulation, the loop coordinates multiple steps: it routes the appropriate CSI data to the active engine(s) according to the selected attack scenarios and placed nodes, aggregates authentication success rates, key generation performance, and detection rates into KPIs, and finally updates the visualisation. To prevent the interface from freezing, computationally intensive tasks such as spoofing AoA map generation and SKG reconciliation are offloaded to background threads with progress queues, while on-disk caching in the Spoof Detector further improves responsiveness for repeated configurations.

Taken together, this layered organisation enables the prototype to effectively realise its objectives by allowing interactive exploration of physical-layer security techniques within a sensing-driven and trust-aware closed loop. Users can dynamically test attack scenarios and immediately observe how the WP5-integrated detectors identify threats, how mitigation strategies (such as jammer subtraction in the spoofing path) improve AoA estimation, and how secret key generation performs under different conditions. The modular, dataset-driven design adopted ensures reproducibility, clarity, and a solid foundation for demonstrating robust 6G security concepts.

3.5 Prototype 5: Master Prototype

3.5.1 Overview and Demonstration Objectives

The prototypes presented in the sections above are mainly focused on single layers of the ROBUST-6G architecture (see Section 2.1). This implies that the single prototype is only able to demonstrate a specific set of functionalities proposed by the project, limited to a certain architecture part, and not able to showcase how the different parts of the system should work together to provide security in 6G. The aim of Prototype 5 is to demonstrate the ROBUST-6G system as a whole, by integrating the functionalities implemented by the other four prototypes into a coherent security platform. In particular, the prototype considers the following aspects:

- **Access to security capabilities and request of security services:** this part is in charge of NetSecaaS GW demonstrated in Prototype 3.
- **Security service orchestration and automation**: the Prototype 2, Multi-Layer Zero-Touch Defender covers these functionalities with the Security Service Orchestrator, Security CL-based automation, and pervasive monitoring via the Programmable Monitoring Platform.
- **Exploitation of AI-based security functions**: these are collected via the Global Model Repository (GMR, Prototype 1)
- **PHY Layer security services**: provided by Prototype 4, Physical and Sensing Layer Trustworthiness.

It is important to highlight that the Prototype 5 aims to demonstrate that it is not merely a collection of separate software components, but that it is capable of integrating the functionalities of the various prototypes into a single system. The PHY security services and AI-based security functions are modelled with the Security Ontology and Knowledge graph developed in WP4, visible in the internal catalogues of the Security Orchestrator, and composable to build security services. Furthermore, they are exposed by the Exposure Framework, through which their orchestration can be directly requested. The interactions between the different parts of the prototype are detailed in Section 3.5.4, where the demonstration storyline is presented.

3.5.2 Position within the Architecture

Figure 3.13 shows the Prototype 5 coverage of ROBUST-6G architecture, given by the union of the focus domains from the other four prototypes. The infrastructure layer is not covered, since it is out of the ROBUST-6G scope.

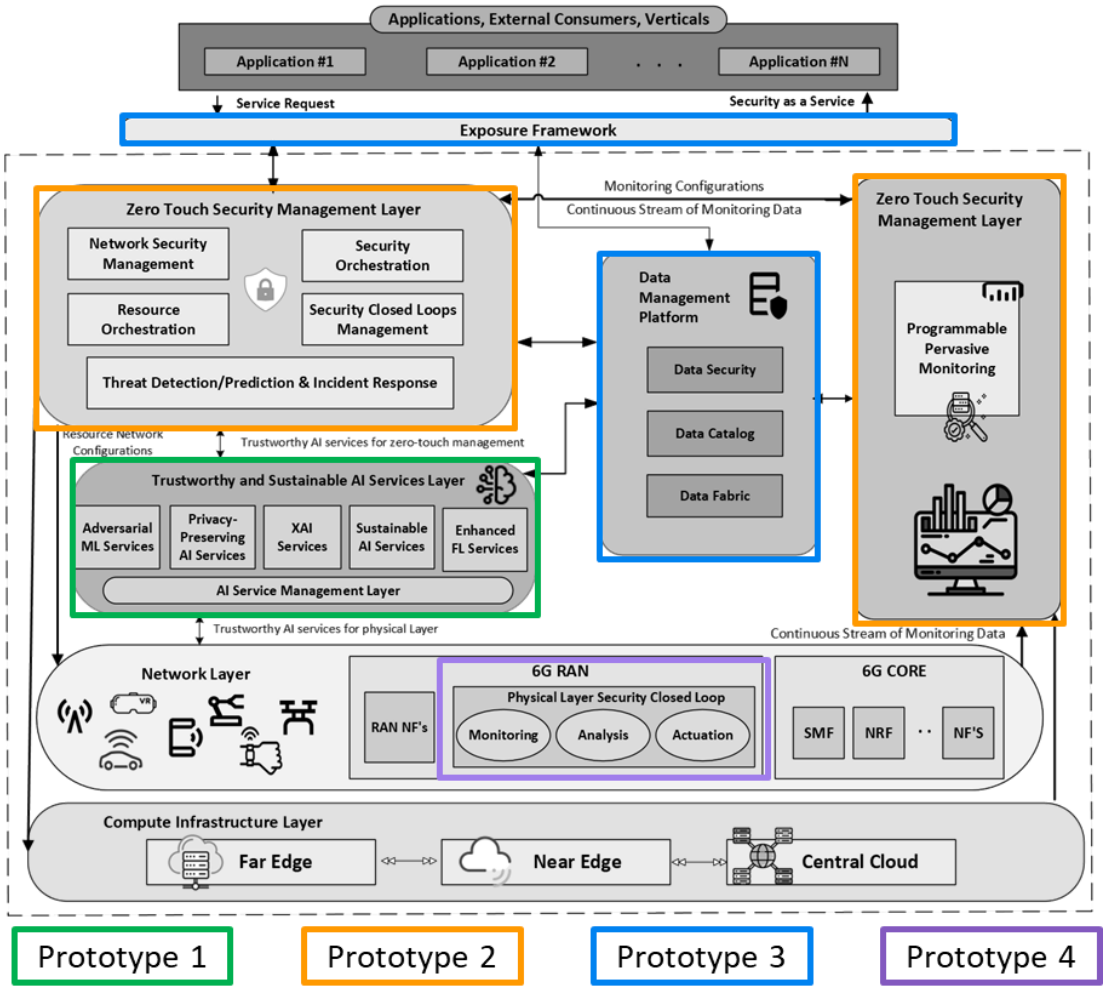

Figure 3.13 Prototype 5 ROBUST-6G architecture coverage

3.5.3 Prototype Composition and Architecture

The previous section described how the prototype is mapped on the ROBUST-6G architecture, presented in Section 2.1 and detailed in the various deliverables of WP2. In this section, the objective is to explicitly identify the components building the prototype, as shown in Figure 3.14.

- **Exposure Framework (CTID03).** Represents the entry point for the security service consumers (verticals, applications, or even third-party orchestrators).
- **Zero-Touch Security Platform (CTHA01, CNXW01, CNXW03, CNXW04, CUMU01).** Provides security orchestration and advanced monitoring capabilities through multiple components.
- **Global Model Repository (GMR).** The centralised registry providing trustworthy AI/ML models that the orchestrator dynamically selects to serve as the anomaly detection or prediction engines within orchestrated security loops
- **Physical Layer Security Loop (CEBY05).** The localised physical-layer execution environment. Within the Master Prototype, these mechanisms are logically mapped into the ZTSO's Security Catalogue as targetable functions that can be triggered using dedicated OpenC2 actuators

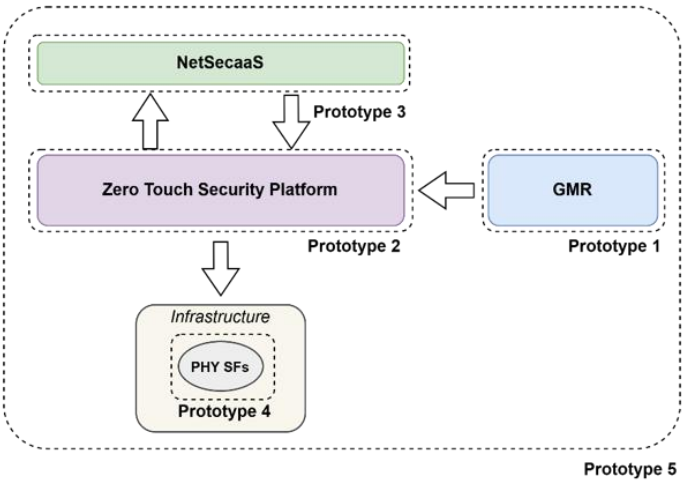

Figure 3.14 Prototype 5 Composition

3.5.4 Demonstration Storyline

Prototype 5 is not directly linked to any project's use cases, so there are no pre-established reference scenarios. Since the aim is to give a demonstration of the ROBUST-6G platform as a coherent security entity, the prototype considers the deployment of at least two security services, to showcase the capability to manage multiple and heterogenous security services. The first service considered is a variation of Use Case 2 (see Section 4.2) that introduces the Global Model Repository beyond the ZTSP. Figure 3.15 shows the high-level integration between the different components involved:

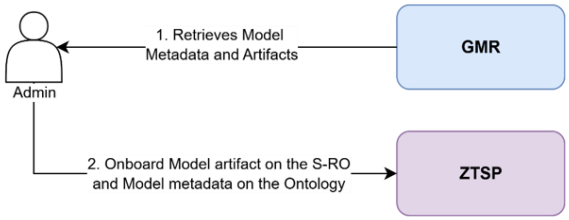

Figure 3.15 Prototype 5 – GMR Models onboarding on the ZTSP

1. An Administrator checks the existing models listed in the GMR together with their metadata (e.g. model metrics) and downloads the relevant model artefacts (e.g. BentoML, Docker Image).
2. The Administrator then onboards the Model artefacts in the Secure Resource Orchestrator (S-RO) as orchestrable artefacts that can be managed by the ZTSP and onboards the Model details in the ZTSO Security Functions catalog respecting the ZTSO Ontology.

From this moment on, as explained in D4.4 [R6G26-D44], the Security Function is registered and considered by the ZTSO as possible element of a Security Service, based on the description in terms of Functional Capabilities provided by the Ontology. The second phase of this process, depicted in Figure 3.16, follows the standard Prototype 2 workflow:

1. An SSLA is submitted by a Security Service consumer (e.g. Vertical Consumer).
2. The ZTSP, in particular the ZTSO, generates a Security Service using the AI Model as Security function (e.g. for Network Analysis).
3. The Security Service is deployed by the ZTSP, including the AI Model that will serve as Security Function.

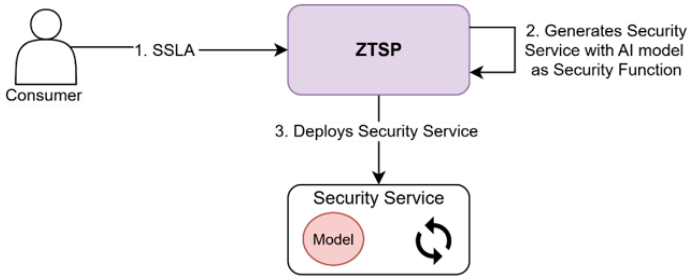

Figure 3.16 Prototype 5 – GMR Model Orchestration as Security Function

The second service makes use of the security capability of the PHY layer. In this case the idea is to orchestrate a security closed loop by exploiting the PHY CL Framework from prototype 4. Due to the architecture of PHY layer security services (lab prototypes with limited programmability) a complete lifecycle management is not feasible. The orchestrator, in this case, requests the activation of the PHY security service, while the PMP (part of the ZTSP) will collect notifications concerning security events. To enable the usage of PHY resources, the ZTSP needs to model PHY SF in the ZTSO catalogue following the ZTSO ontology and use dedicated OpenC2 actuators that can interact with the PHY SF. The choice of using OpenC2, as described in D4.4 [R6G26-D44], makes the Security Services S-CLs capable of interacting with heterogeneous targets. Figure 3.17 depicts the first part of this process, where the admin onboards the PHY SFs metadata in the ZTSO Catalogue together with the needed OpenC2 actuator artefacts in the S-RO.

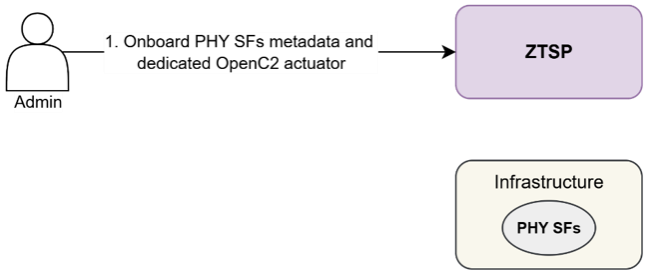

Figure 3.17 Prototype 5 – PHY SF and related OpenC2 Actuator Upload on the ZTSP

Now that the PHY SFs is modelled in the ZTSO and the needed artefacts (i.e. the OpenC2 actuator), the ZTSP is capable of generating and deploying a Security Service capable of interacting with the PHY SF. Figure 3.18 reports the process composed of the following steps:

1. An SSLA targeting the Physical Layer is submitted to the ZTSP.
2. A Security Service composed of a S-CL and the needed PHY OpenC2 actuator is generated.
3. The ZTSP deploys the Actuator and the S-CL in the target infrastructure.

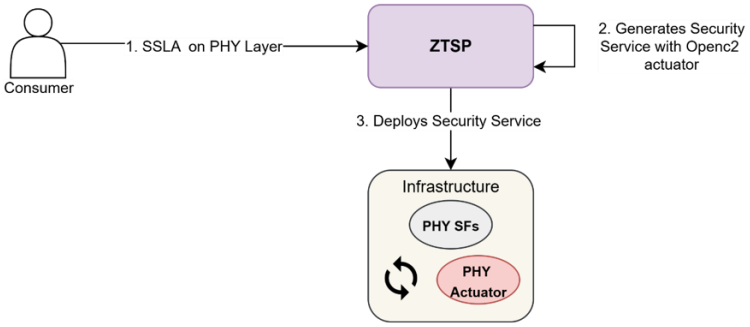

Figure 3.18: Prototype 5 - PHY S-CL with PHY OpenC2 Actuator

While the previous workflows showcase the interaction between the ZTSP, the GMR, and the PHY Layer, the final scenario of prototype 5 is the demonstration of the interaction between the ZTSP and the NetSecaaS module to simplify the submission of SSLAs. Figure 3.19 depicts the interaction between these components with the Consumer that is able to request a Security Service through simplified APIs and the NetSecaaS component that is able to translate them into a complete and ready to be ingested SSLA.

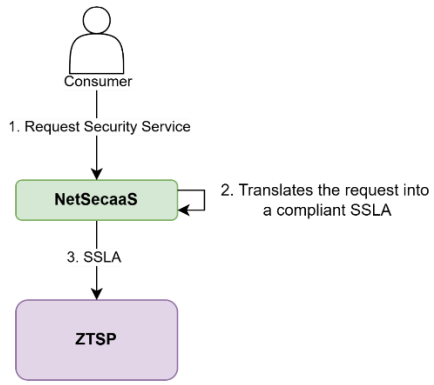

Figure 3.19: Prototype 5 - Exposure of Security Capability thought the NetSecaaS

The overall purpose of this demonstration storyline is to showcase that the ROBUST-6G platform components developed in the different prototypes are able to interact and **the full ROBUST-6G solution can be considered as a comprehensive solution able to integrate Zero-Touch Security Orchestration, AI/ML Models, PHY Security, and simplified exposure of Security capabilities to the end user.**

4 Use Cases

This section presents the three ROBUST-6G Use Cases, which are tailored to demonstrate the functionalities of parts of the ROBUST-6G Platform in real-world scenarios. In the following subsections, the use cases scenarios in their final form with the components/flows/integration information are presented. Crucially, these use cases are intended to validate the ROBUST-6G Platform prototypes; a given use case may cover one or multiple prototypes, as well as the intersections between them. The results of this process are divided into two distinct evaluations later in the document. Section 5.1 details the functional validation, where the use cases serve as the means to validate the prototype functionalities. Subsequently, Section 5.2 focuses on quantitative validation, assessing the use cases in terms of Key Performance Indicator (KPI) attainment based on the targets defined during the initial phases of the project.

4.1 AI model trustworthiness evaluation for 6G decentralised scenarios

Use Case 1 addresses the evaluation of AI model trustworthiness in 6G distributed environments. The use case leverages Decentralised Federated Learning (DFL) to enable collaborative model training across multiple administrative domains without centralising sensitive data. Trustworthiness is assessed across four pillars: robustness against adversarial threats, sustainability in terms of computational efficiency, explainability for transparent decision-making, and fairness to prevent algorithmic bias. UC1 is structured across two complementary scenarios.

Use case 1 scenario 1 is represented by the DFL Framework and its interaction with the Global Model Repository (GMR). The DFL Framework provides the decentralised training environment, where participants perform local learning and exchange model updates with neighbouring nodes through peer-to-peer communication. The GMR complements this process by acting as a governance and evidence layer, storing model versions, configuration metadata, metrics, logs and explainability artefacts.

This scenario introduces the AI-oriented part of Use Case 1 and provides the basis for validating several trustworthiness dimensions, including privacy preservation, robustness against malicious or unreliable participants, sustainability of distributed AI execution and explainability of trained models. The scenario is instantiated through cybersecurity-oriented collaborative learning, using the TON-IoT dataset and the CyberNet model as the main reference workflow for decentralised intrusion-detection model training.

Use Case 1 Scenario 2 addresses the trustworthiness and resilience of the physical and sensing layers through the Physical Layer Closed Loop, the monitoring, analysis and actuation mechanism that integrates the physical layer security solutions (mainly developed in WP5) into the 6G RAN. The channel is continuously monitored and analysed to evaluate PHY-layer trustworthiness by detecting and localising jamming, performing mutual authentication from the transmitter's angular signature, and generating secret keys from channel reciprocity; the analysis outcomes then drive the actuation stage, namely transmission-power adaptation fed back to the PHY layer through RAN control.

In this context, this scenario demonstrates the capabilities of the Physical Layer Closed Loop, which does not merely detect an attack but acts on it to preserve link quality. It exercises three complementary properties aligned with its core objectives: resilience to jamming (PHY layer trustworthiness evaluation), authentication against spoofing and impersonation (mutual authentication), and confidentiality against eavesdropping (fast secret key generation). All three capabilities are validated on real channel measurements from a 64-antenna indoor massive-MIMO testbed, using the Ultra-Dense Indoor MaMIMO CSI dataset.

4.1.1 Scenario 1: Decentralised federated learning for joint privacy-preserving ML/DL model training

This subsection covers the ROBUST-6G scenario dealing with decentralised federated learning framework as foundation to ensure the privacy of data being part of AI models that will be used by certain ROBUST-6G architecture components.

4.1.1.1 Scenario Overview and Objectives

In modern 6G deployments, highly sensitive telemetry and network traffic are continuously generated at the edge, spanning Radio Access Networks (RAN), Edge Clouds, and user terminals. Transporting this raw data to a central cloud server introduces severe bandwidth bottlenecks, increases latency, and poses major data privacy risks. To address these issues, Scenario 1 implements a fully Decentralised Federated Learning (DFL) Framework (CUMU02) where geographically distributed edge nodes collaborate to train a global Deep Learning model without ever exchanging raw local data.

Goal of the Scenario

The primary goal of Scenario 1 is to train a robust, high-performance network intrusion detection model, specifically named CyberNet, using the highly realistic and diverse TON_IoT dataset. Instead of aggregating raw training data at a central location, the DFL Framework moves the training computation directly to the edge nodes. These nodes execute local training iterations on their private

data partitions and synchronize their resulting model updates (weights/gradients) with their peers using an epidemic peer-to-peer (P2P) gossip communication protocol.

Core Objectives

- **Adversarial Robustness and Defense**: Open, decentralised 6G edge environments are highly vulnerable to malicious or compromised participants. These adversaries can launch active **Byzantine attacks** such as model poisoning designed to degrade classification performance, seed backdoors, or cause catastrophic training failure. The core objective is to validate the system's ability to maintain training integrity under active exploitation by implementing the spatial-distance-based **Krum aggregation filter (CUMU04)** inside the Gossip protocol to isolate and discard poisoned model updates.
- **Sustainability and Resource Efficiency:** Edge execution requires rigorous energy constraints due to battery and CPU limitations. To optimize the environmental footprint of the training lifecycle, the scenario integrates:
    1. **ADMM (Alternating Direction Method of Multipliers) (CUPD01):** For scalable, distributed optimization that minimizes communication overhead and accelerates local convergence.
    2. **SNN (Spiking Neural Network) Simulator (CUPD02):** To implement sparse, event-driven neural architectures that significantly reduce local compute power draw, memory footprints, and gradient serialization sizes.
- **Explainability and Trust Verification:** To provide absolute transparency and allow human-in-the-loop auditing, the training lifecycle integrates **XAI Services (CUMU05)**. Rather than performing post-hoc analysis, the XAI components evaluate the model's stability round-by-round. It quantifies **Feature Attribution Stability** (by calculating cosine similarity of Shapley feature importances via SHAP) and generates qualitative **Data Representation Visualisations** (using 2D t-SNE scatter plots) to confirm that the model is learning distinct, meaningful clusters for different data classes.

4.1.1.2 Position within the Architecture

Scenario 1 is positioned within the Trustworthy and Sustainable AI Services part of the ROBUST-6G architecture and demonstrates how decentralised edge training relates to model governance through the interaction between the DFL Framework and the Global Model Repository (GMR).

Mappings to the Trustworthy and Sustainable AI Services Layer

The core services involved in this scenario reside within the Trustworthy and Sustainable AI Services part of the architecture and are mapped to the following technical pillars:

- **[A] Adversarial ML Services (Robustness Dimension):** This component is realized by the **Enhanced AI/ML Model Robustness service (CUMU04)**. It operates dynamically within the edge environment to execute Krum Byzantine-robust aggregation algorithms, filtering out malicious or poisoned weight updates during P2P gossip weight synchronization.
- **[B] Privacy-Preserving AI Services (Privacy Dimension):** This corresponds to the **Privacy-Preserving and Security-Enhanced DFL service (CEBY02)**. It provides homomorphic encryption configurations, joint public key generation parameters, and partial multi-party decryption fusion algorithms to ensure zero exposure of raw edge telemetry.

- **[C] XAI Services (Explainability Dimension):** Mapped to the **XAI Integration for Model Explainability service (CUMU05)**, which continuously processes converged model versions to extract mean absolute Shapley values (SHAP) for quantitative feature attribution stability and 2D t-SNE scatter projections to verify latent representation quality.
- **[D] Sustainable AI Services (Sustainability Dimension):** Mapped to the **Sustainable Client Scheduling Scheme (CLIU01)**, **SNN Simulator (CUPD02)**, and **ADMM-based Aggregation System (CUPD01)**. These services evaluate node energy footprints (CPU/GPU power draw in Joules), optimize edge scheduling, and deploy sparse event-driven neural architectures to reduce edge compute constraints.
- **[E] Enhanced FL Services (Federation Dimension):** Represented by the core **Decentralised Federated Learning Framework (CUMU02)**, which coordinates edge client instantiations, establishes P2P mesh topologies, and governs the iterative weight gossiping communications.
- **[F] AI Service Management Layer (Governance Dimension):** Operates as the underlying manager that connects GMR registration endpoints, manages session transactions, and handles database operations.

Compute Infrastructure Layer Separation

The physical execution exhibits a clear separation across the Compute Infrastructure Layer:

- **Decentralised Training Environment (Far Edge & Near Edge):** The actual training of the **CyberNet** intrusion model is executed at the **Far Edge** and **Near Edge** compute infrastructure (e.g., RAN nodes, base stations, and user terminals). These nodes run local PyTorch containers and exchange weights asynchronously over the **P2P Gossip Network**. Raw network telemetry never leaves this edge environment, ensuring complete data privacy and minimising backhaul bandwidth.
- **Central Administrative Layer (Central Cloud):** The **Global Model Repository (GMR)** resides in the **Central Cloud** or core network premises. The GMR does not participate in training or touch raw edge data; instead, it acts as a secure, authenticated, PostgreSQL-backed central model registry that logs training telemetry, saves converged weights, and serves certified models via secure REST APIs.

4.1.1.3 Functional Flows Description

To operationalise the validation lifecycle, Scenario 1 consists of **five functional flows** originally formulated in D6.2. These have been fully integrated, implemented, and validated in D6.3.

Flow UC1_1_01: Baseline Federated Learning Flow (Privacy and Decentralization)

Objective: Establish and validate the baseline mechanics of decentralised collaborative ML/DL model training across edge nodes under a trusted (benign) environment. It confirms that the P2P Gossip protocol correctly distributes, synchronises, and converges model parameters without exposing raw local data.

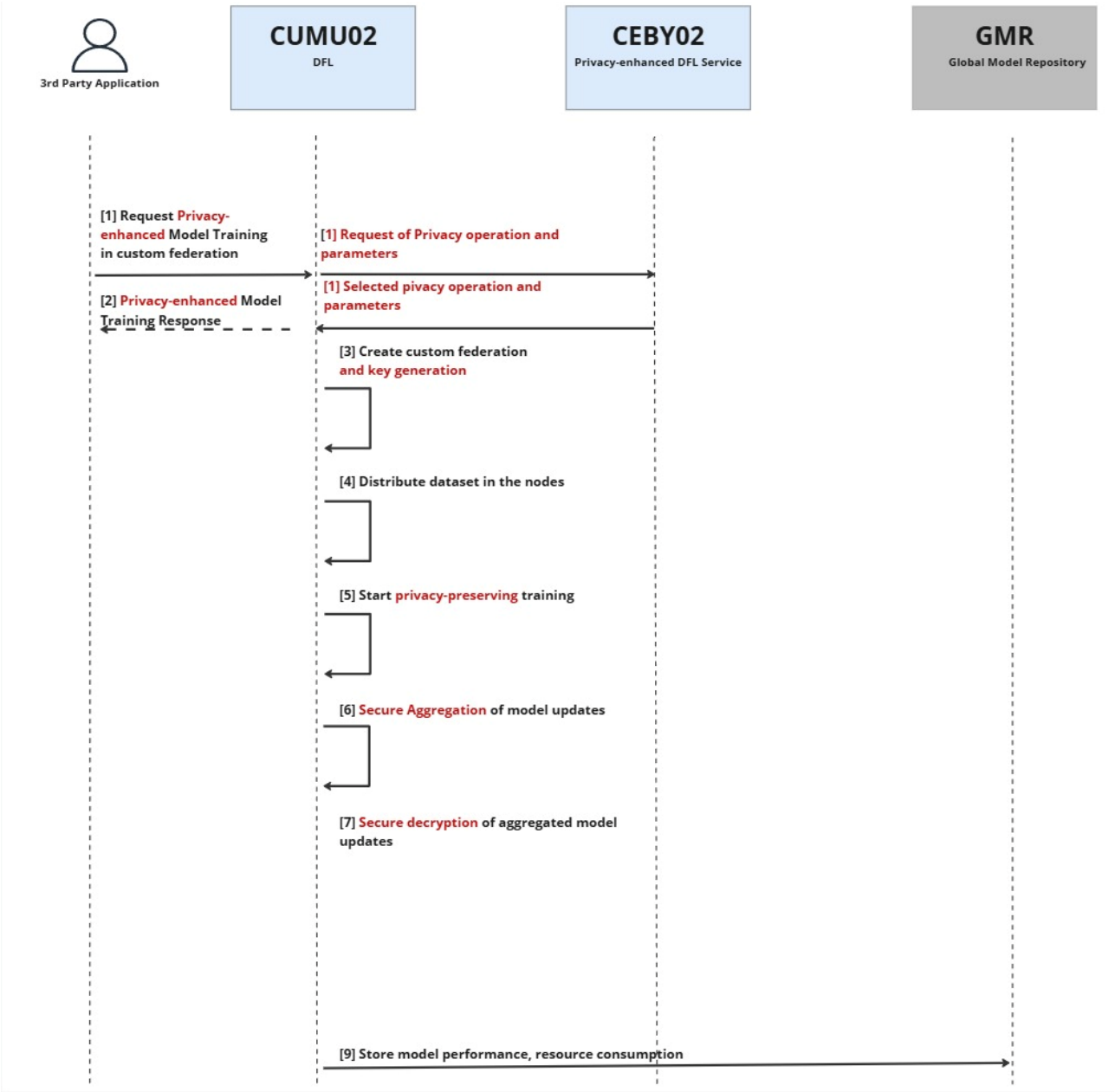

Figure 4.1 Flow UC1_1_01 – Privacy and decentralization flow diagram, benign nodes

Process Description: The process, as illustrated in the sequence diagram of Figure 4.1, is initiated by an external actor and managed through a coordinated interaction between the core components:

The process is initiated when an external 3rd Party Application or Orchestrator submits a request to the ROBUST Controller API (on Port 8000 at /api/robust/run/scenario) [TMJ+26a] 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 (e.g., CyberNet or SNN), dataset (TON_IoT), and training configuration (e.g., number of rounds, noise attacks, or Krum defense). The ROBUST Controller acknowledges the request, initiates the Docker Compose orchestration to spin up the decentralised base node containers, and responds to confirm the initialisation of the training process (Step 2).

Upon initialisation, the ROBUST Controller proceeds with the internal setup. It creates the custom containerised federation of base nodes via Docker Compose (Step 3), then the nodes dynamically load their local dataset partition using Dirichlet distribution scripts in pytorch/ton_iot/ton_iot.py (Step

4), ensuring strict local data isolation. Once the environment is prepared, the controller runs the framework start command on each node container to initiate the P2P training process (Step 5).

In the benign baseline flow, the edge nodes execute local training epochs on their dataset partitions and share their weights asynchronously using P2P Gossip protocols. During synchronization, the gossip engine (CUMU02) receives peer updates and computes baseline aggregation metrics for logging purposes (Steps 6 and 7). Since no active Byzantine attack is present in this baseline scenario, no defensive filtering is applied. Following the exchange of weights, the framework proceeds with standard P2P aggregation (e.g., FedAvg) to fuse the local models (Step 8).

At the conclusion of each training round, the nodes invoke their RepositoryClient to upload key artefacts to the Global Model Repository (GMR) FastAPI server on Port 8001. This includes the updated model weights (uploaded via /api/robust/upload/model), performance metrics (uploaded via /api/robust/upload/metrics), and execution logs, ensuring a complete transactional record persisted in the PostgreSQL database.

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.

Flow UC1_1_02: Evaluation of Model Robustness (Byzantine-Resilient & Reputation-Based Filtering)

Validate the system's ability to maintain training integrity under hostile conditions. It tests the resilience of the framework against active Model Poisoning attacks (Gaussian noise injection via attacks.py) by integrating the Krum Byzantine-Robust Aggregation Engine (CUMU04) directly within the Gossip protocol layer.

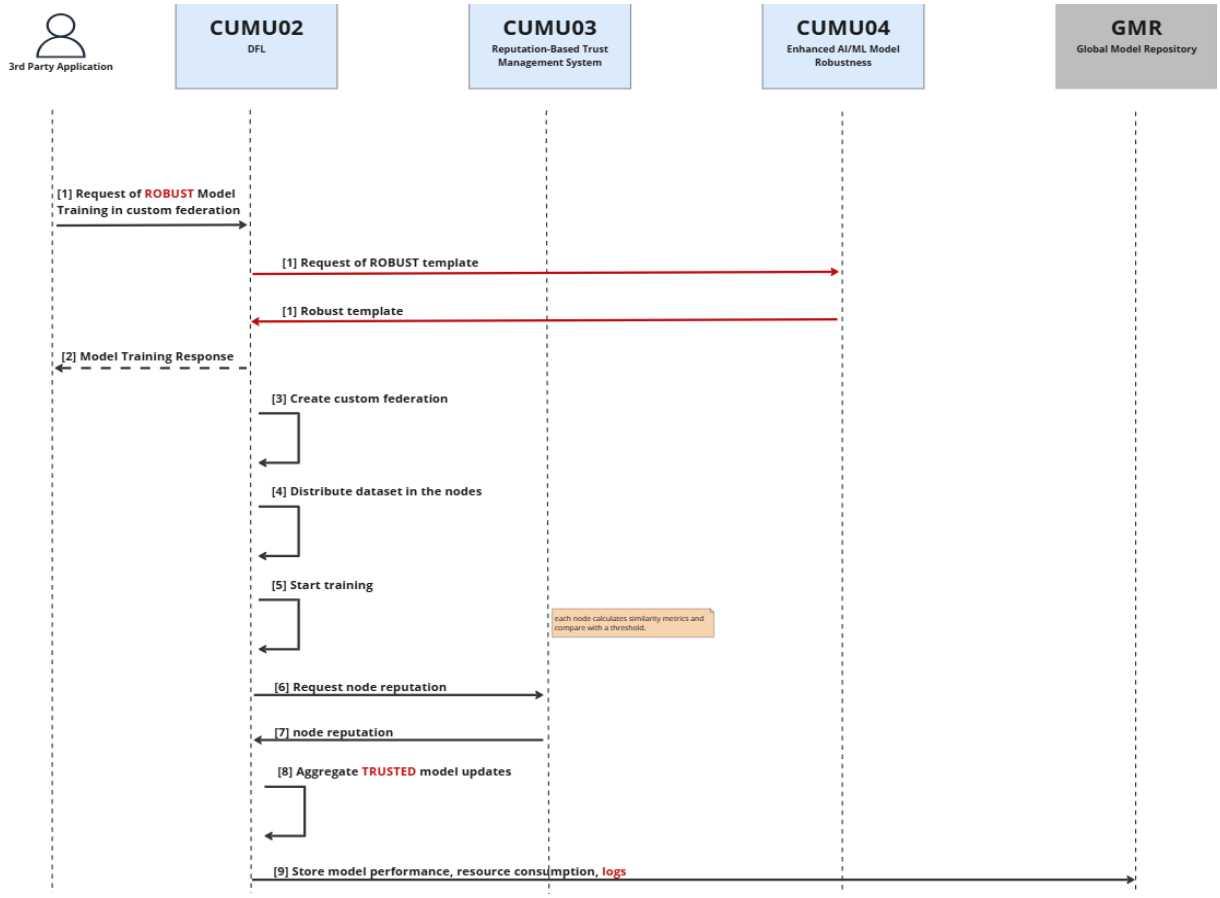

Figure 4.2 Flow UC1_1_02 – Sequence diagram for evaluating DFL system's robustness under attack

Process Description: The process, as depicted in the sequence diagram of Figure 4.2, integrates a robustness-aware workflow from the very beginning to mitigate active poisoning threats:

The workflow is initiated when a 3rd Party Application sends a request for a 'ROBUST' model training session to the ROBUST Controller API, specifying the activation of the Krum aggregation filter (Step 1). In response, the controller parses the security configuration and provisions the base nodes with the Krum spatial aggregation policy. Krum operates dynamically on the received weight vectors by calculating spatial distance and selecting the most central update, without requiring pre-configured templates or central reputation database checks.

The ROBUST Controller confirms the training session initiation to the 3rd Party Application (Step 2). It then initiates the Docker Compose orchestration, starting a custom federation that includes both benign base nodes and designated Byzantine attackers (Step 3). The TON_IoT dataset partitions are dynamically loaded by the edge containers (Step 4), and the P2P training process begins (Step 5). The malicious containers are programmed to automatically inject Model Poisoning Gaussian noise into their weight tensors before transmitting them to peers.

During each communication round, after completing local PyTorch training epochs, the base nodes exchange weight tensors over the P2P Gossip socket network. When a node receives weights from its neighbours, it routes them to the Krum Aggregator (CUMU04) (Step 6). Krum computes a complete pairwise spatial distance matrix among all received updates to identify outlier vectors.

Krum evaluates the spatial density of the updates by summing the Euclidean distances of each weight tensor to its closest neighbours (Step 7). Because Gaussian noise-poisoned weights represent extreme

statistical outliers in the high-dimensional parameter space, they lie far outside the cluster of benign updates and yield high distance scores. In the final aggregation step, Krum selects the update vector that minimizes the closest neighbour distance sum, automatically discarding the poisoned outliers, and returning the untainted representative weight vector for model fusion (Step 8).

At the end of each round, the edge nodes invoke their RepositoryClient to upload the converged model weights, the Krum spatial calculation metrics, and the round execution parameters to the GMR FastAPI server via Port 8001 (Step 9). The GMR backend persists this detailed execution trace and defensive metrics directly in the PostgreSQL database, providing a complete, auditable record of the federation's adversarial resilience for post-hoc analysis.

Flow UC1_1_03: Sustainability Evaluation (Energy-Aware / Sustainable Client Scheduling)

- **Objective:** Assess and optimize the environmental footprint of the decentralised AI lifecycle. It focuses on scheduling resource-constrained nodes, evaluating energy overhead during edge training, and utilizing energy-efficient network structures.

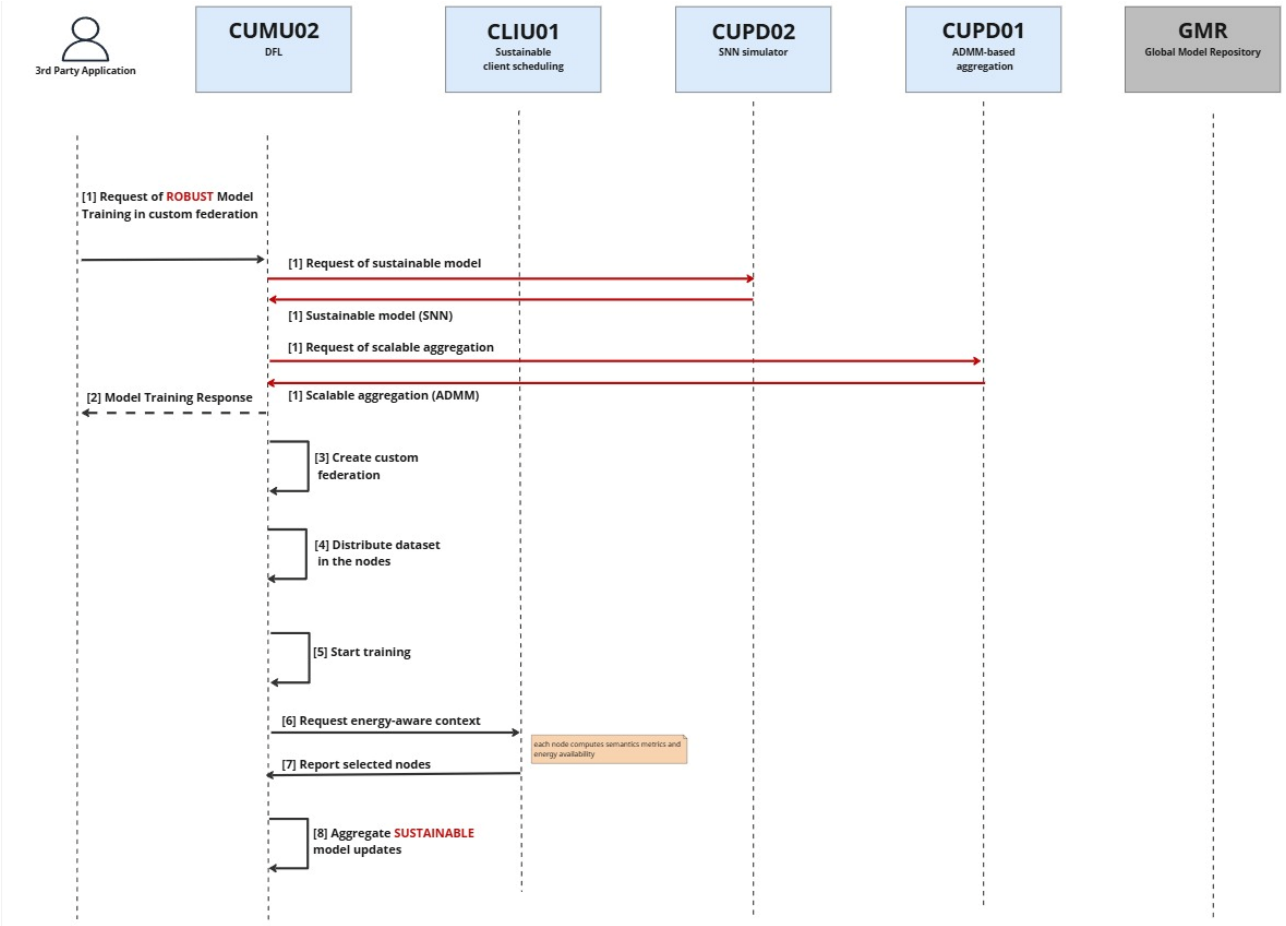

Figure 4.3 Flow UC1_1_03 – Sustainable and efficient evaluation of the model training lifecycle

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:

- 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.

- 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.
- 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).
- 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 (GMR)** (Step 9).

Flow UC1_1_04: Explainability of the Models Obtained (XAI Integration)

- **Objective:** Enable continuous, human-in-the-loop auditability and trustworthiness verification of the evolving AI model throughout the training lifecycle, rather than as a post-processing step.

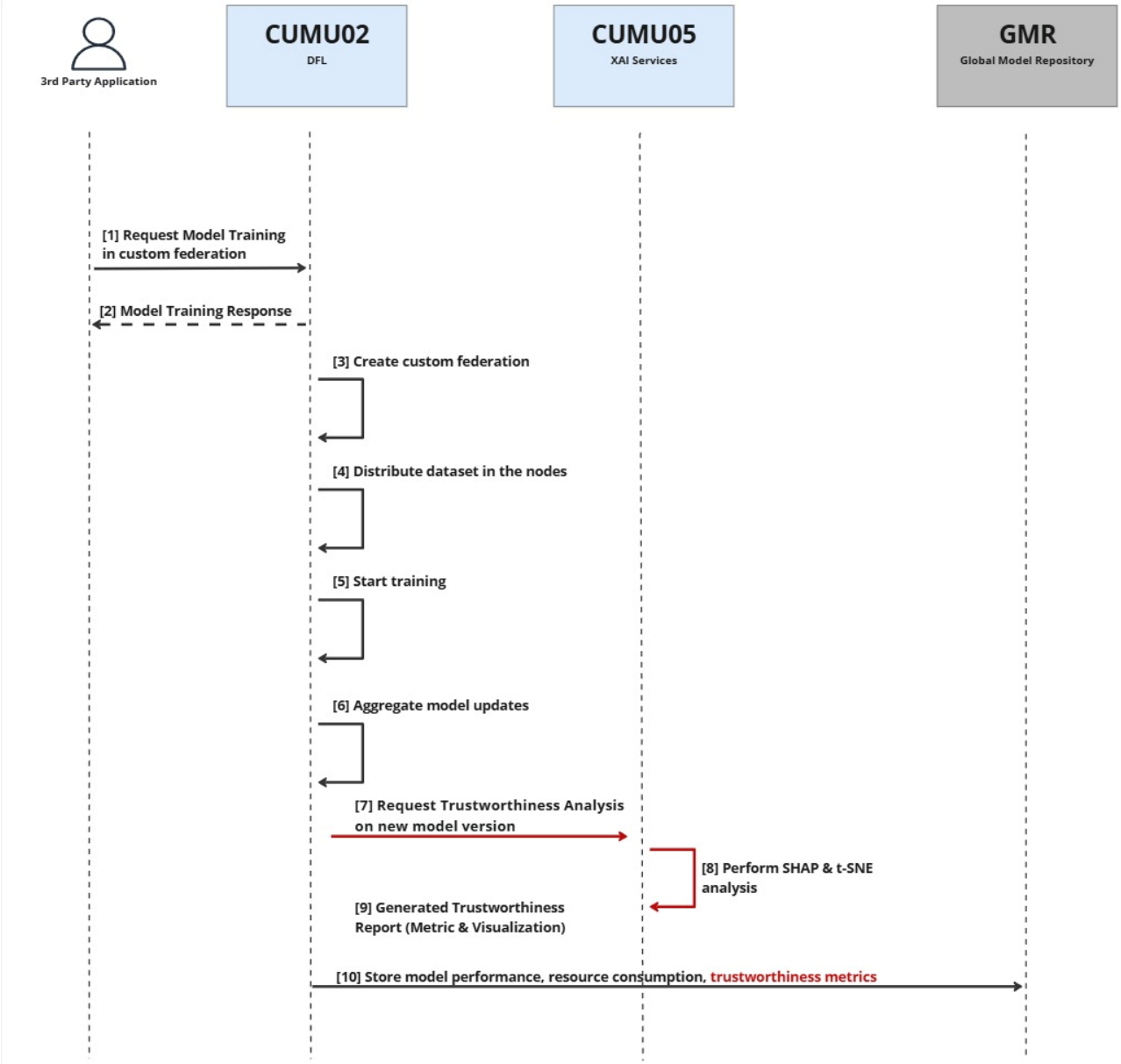

Figure 4.4 Flow UC1_1_04 – Continuous monitoring of explainability (SHAP/t-SNE) in federated training

Process Description: The process, as illustrated in the sequence diagram of Figure 4.4, integrates XAI analysis as a systematic monitoring step within each federated training round.

- 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).
- 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.
- 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). The XAI Services perform a comprehensive analysis on this specific model version (Step 8), generating one quantitative metric and one qualitative visual artefact:

- Metric 1 – Feature Attribution Stability (from SHAP): This metric provides a quantitative measure of the model's reasoning stability. The XAI Services calculate 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.
- Artefact 2 – Data Representation Visualisation (from t-SNE): This artefact provides a powerful qualitative tool for human-in-the-loop analysis. It uses the current version of the model to generate latent space embeddings for a validation dataset and then creates a 2D t-SNE visualisation 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 artefact. The XAI Services then compile the quantitative stability metric and the t-SNE visualisation into a structured Trustworthiness Report and return it to the DFL Framework (Step 9).
- 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 visualisation. 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.

Flow UC1_1_05: Privacy-Enhanced DFL (Secure Cryptographic Aggregation)

- **Objective:** Ensure complete privacy against "honest-but-curious" nodes. It guarantees that even if participating edge nodes attempt to reconstruct private training data from the exchanged gradients of their peers, they cannot decrypt individual updates.

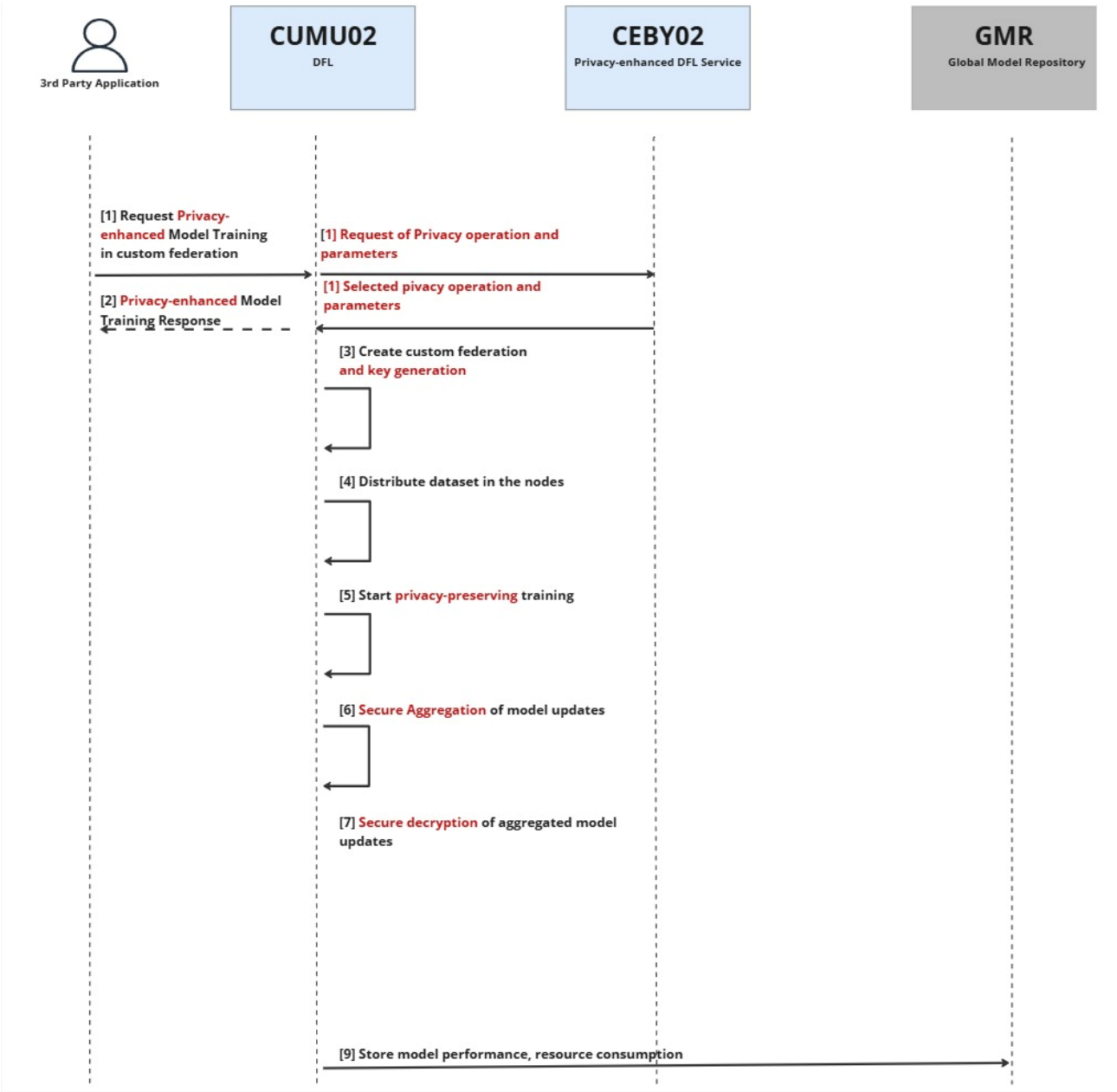

Figure 4.5 Flow UC1_1_05 – Privacy-enhanced collaborative model training flow diagram via secure aggregation

Process Description: The process, as illustrated in the sequence diagram of Figure 4.5, is initiated by an external actor and managed through a coordinated interaction between the core components:

- The process is initiated when an external 3rd Party Application submits a request to the ROBUST Controller API to start a new privacy-enhanced DFL training session on the testbed (Step 1). The request specifies standard training hyperparameters and requests secure communication protocols governed by the Privacy-Preserving DFL service (CEBY02). The ROBUST Controller acknowledges the request, provisions secure P2P link keys, and responds, confirming the initialisation of the secure training lifecycle (Step 2).
- The ROBUST Controller initiates Docker Compose orchestration to spin up the edge node containers (Step 3). During initialisation, each edge container invokes its local RSACipher and AESCipher modules (implemented in encrypter.py) [CDL-DFL]. The nodes generate local RSA key pairs, exchange their serialised public keys over base64-encoded channels, and establish shared symmetric AES keys to secure peer-to-peer communication sockets. The nodes then load their partitioned TON_IoT dataset chunks in a privacy-preserving manner (Step 4).

- Once the environment is prepared, the framework starts the iterative training loop (Step 5). Each base node trains the CyberNet model on its isolated data partition. Before sharing model weight tensors over the gossip mesh, the nodes invoke the AES-GCM cipher in encrypter.py to encrypt the serialised state dictionaries using fresh nonces and authentication tags. The encrypted weight packets are then safely transmitted to neighboring peer containers.
- Upon receiving encrypted weight packets from adjacent nodes, the receiving container decrypts the payload using its shared P2P socket key and verifies the GCM authentication tag, ensuring that in-transit data has not been modified or intercepted by curious edge observers (Step 6).
- Once the incoming weight updates are decrypted and authenticated at the receiving node, they are routed to the Krum aggregation filter (CUMU04) to dynamically check for Byzantine noise anomalies, ensuring that privacy-preserving communications are also robust against model poisoning (Step 7).
- The Krum aggregator selects the representative weight update and fuses it into the active local model. The decrypted, robustly aggregated global model parameters are then prepared for the next round of training, combining absolute link-level data privacy with robust Byzantine resilience (Step 8).
- At the conclusion of each training round, the edge nodes invoke their RepositoryClient to upload model weights, convergence metrics, and logs to the GMR FastAPI /api/robust/upload/model and /api/robust/upload/metrics endpoints over encrypted TLS channels, persisting the secure round transaction in the PostgreSQL database (Step 9).

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

4.1.1.4 Prototype in Use

The flows, components, and mathematical logic described in Scenario 1 are concretely implemented, executed, and validated using **Prototype 1: Trustworthy AI**, developed by **UMU**.

Mapping of Scenario Components to Prototype 1

- **The DFL Framework (CUMU02):** Realized as a modular, Python-based microservice orchestrating PyTorch computations and gossip protocols.
- **XAI Integration (CUMU05):** Implemented via specialized explainability modules (*mnist_shapley_explainer.py* and *shats_explainer.py*) that perform background SHAP and t-SNE processing on aggregated models using validation datasets.
- **Global Model Repository (GMR):** A REST API server (*robust_gmr/server.py*) written in FastAPI and backed by a **PostgreSQL database**. The GMR maintains a complete, normalised registry of all training sessions, rounds, nodes, trust metrics, and model binaries (stored securely with SHA-256 integrity verification hashes).
- **Decentralised Edge Nodes:** Deployed as isolated **Docker Containers**. These containers leverage hardware acceleration and communicate asynchronously using standard TCP sockets.
- **Frontend UI Dashboard:** A modern **dashboard** that allows researchers and system administrators to initiate Dirichlet-partitioned federations, monitor real-time accuracy/loss curves, trace node reputation, and visually inspect Shapley Values for Time Series Models (SHATs) stability and t-SNE scatter plots.

4.1.2 Scenario 2: Physical and Sensing Layer Trustworthiness and Resilience

4.1.2.1 Scenario Overview and Objectives

Use Case 1 – Scenario 2 (UC1.2) focuses on validating the trustworthiness and resilience of the physical and sensing layers in 6G networks through Physical Layer Security (PLS) mechanisms developed in WP5. These mechanisms include PHY monitoring, physical layer authentication, trusted

sensing and localization, and secret key generation, with the objective of ensuring the integrity, privacy and resilience of wireless communications while preserving the performance requirements of future 6G services.

This scenario demonstrates the **Physical Layer Closed Loop (PLCL)** as the main architectural mechanism for integrating PLS solutions into the ROBUST-6G architecture. As described in WP5 and outlined at architectural level in D6.2, the PLCL comprises three key stages: i) **monitoring**, where physical-layer observations and radio parameters are collected; ii) **analysis**, where this information is processed to assess trustworthiness, identify anomalous conditions and support authentication or confidentiality decisions; and iii) **actuation**, where appropriate PLS schemes are activated, including secret key generation and PHY control operations. This stage closes the loop by feeding updated RAN specifications back to the PHY layer for continuous adaptation, e.g., through adjustments to power or resource allocation parameters, and by propagating potential infrastructure-layer alerts to the orchestrator. For the purposes of this scenario, the validation focuses on components **CENS01, CENS03, CENS04 and CENS05**, as defined in D6.1, since they support the targeted functionalities of PHY monitoring, anomaly detection and localization, physical layer authentication, and secret key generation, respectively. These functionalities are directly aligned with the physical-layer observations available in the considered indoor massive-MIMO setting.

The scenario considers a representative indoor industrial environment, such as a smart manufacturing cell, where wireless devices, including industrial sensors, automated units and machines, operate in close spatial proximity and communicate with an access point or base station over a private 6G network. Such an environment requires highly reliable and low-latency communications, while the proximity of legitimate and potentially adversarial devices increases the importance of exploiting radio-level information as an additional security source.

The threat model considered in UC1.2 includes attacks affecting the availability, authenticity and confidentiality of wireless communications. These include jamming attacks intended to degrade or disrupt communication links, spoofing or impersonation attempts by adversarial devices, and eavesdropping attacks targeting confidential communications or secret keys established between legitimate entities. Accordingly, the scenario validates how PHY-layer monitoring, analysis and actuation can support the protection of indoor 6G industrial communications.

The **primary goal** of this scenario is to **validate the PLCL as the ROBUST-6G mechanism for integrating PHY-layer security in a representative indoor industrial 6G environment**. The **scenario is structured around three core objectives:**

- **PHY layer trustworthiness evaluation:** PHY-layer trustworthiness assessment through radio-link monitoring, detection and localisation of potential attacks
- **Mutual authentication:** Physical-layer mutual authentication, including the assessment of its trustworthiness limits under potential spoofing or impersonation attacks.
- **(Fast) Secret key generation:** Physical-layer secret key generation, including the assessment of its resilience against correlated eavesdropping attacks.

4.1.2.2 Position within the Architecture

UC1.2 primarily activates the **Physical Layer Security Closed Loop (PLCL)** located within the **6G RAN** of the **Network Layer** in the ROBUST-6G architecture presented in Section 2.1. The PLCL provides the architectural environment for integrating the monitoring, analysis and actuation functions required for physical and sensing layer trustworthiness and resilience.

With reference to Figure 2.1, the architectural functionalities covered by this scenario are the following:

- **Physical Layer Security Closed Loop – Monitoring.** Collects physical-layer observations and radio parameters from the RAN/PHY functions to support the evaluation of PHY-layer trustworthiness.
- **Physical Layer Security Closed Loop – Analysis.** Processes the monitored physical-layer information to detect anomalous or adversarial radio conditions and support authentication and confidentiality decisions.
- **Physical Layer Security Closed Loop – Actuation.** Activates appropriate PLS schemes and PHY control actions according to the analysis outcomes, including transmission power adaptation in response to detected SNR degradation.
- **6G RAN / RAN Network Functions.** Provides the physical-layer observations required by the PLCL and receives updated RAN configurations resulting from the closed-loop actuation process.
- **Potential interaction with the Zero-Touch Security Management Layer.** The PLCL may interface with the Zero-Touch Security Management Layer to receive upper-layer security context and propagate PHY-layer security information towards broader security management functions.

In this manner, UC1.2 is positioned at the physical-layer security branch of the ROBUST-6G architecture, demonstrating how PHY-layer observations and PLS mechanisms are integrated within the 6G RAN through the PLCL, while allowing potential interaction with higher-layer security management functions.

4.1.2.3 Functional Flows Description

The three objectives introduced for UC1.2 are demonstrated through three functional flows operating within the Physical Layer Closed Loop (PLCL):

- **Flow UC1_2_01 – PHY layer trustworthiness evaluation**,
- **Flow UC1_2_02 – Mutual authentication**,
- **Flow UC1_2_03 – (Fast) Secret key generation**

The following subsections describe the component interactions, exchanged information and sequence of operations associated with each flow.

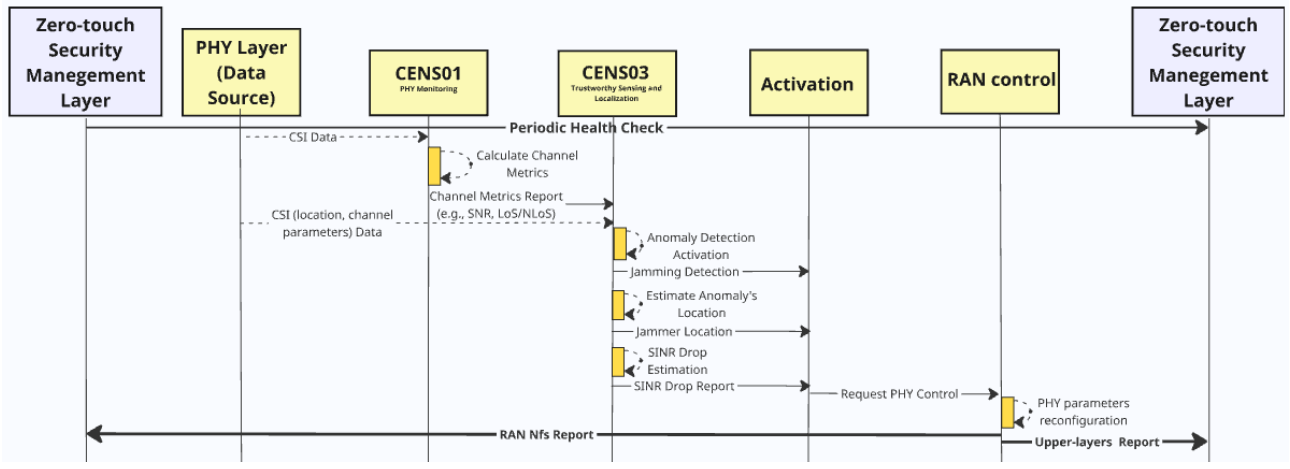

Figure 4.6 Flow UC1_2_01 – PHY layer trstworthiness evaluation

Flow UC1_2_01: PHY layer trstworthiness evaluation

- **Objective:** Assess PHY-layer trustworthiness through radio-link monitoring and the detection of potential attacks, including jamming. The flow supports the identification and localisation of the attack source and enables appropriate PHY-layer mitigation actions to preserve secure and reliable operation.

As illustrated in Figure 4.6, the flow combines **CENS01 – PHY Monitoring**, **CENS03 – Trustworthy Sensing and Localization**, the **Activation** function and **RAN control** in a monitoring–analysis–actuation sequence within the Physical Layer Closed Loop (PLCL).

The flow starts with radio-link information made available by the PHY layer in the considered indoor spatial setting. **CENS01** continuously monitors the observed SNR across the area of interest and provides corresponding measurements to the analysis stage. Under normal operating conditions, the monitored SNR variation is expected to be consistent with the path-loss behaviour characterizing the considered radio environment. In the presence of a jammer, the additional interference causes an unexpected SNR/SINR degradation that deviates from this expected behaviour.

The detection task is formulated as a lightweight binary hypothesis test. Under the normal-operation hypothesis, the monitored SNR observations remain consistent with the expected path-loss behaviour, whereas under the jamming hypothesis, the observations contain an additional spatial degradation pattern caused by an unknown jammer. To distinguish between these two conditions, **CENS03** applies a spatial Generalized Likelihood Ratio Test (GLRT). For each candidate jammer position in the monitored area, the analysis constructs the degradation pattern that would be expected according to the considered path-loss behaviour and compares it against the observed SNR/SINR degradation map. The candidate position providing the strongest match is retained as the estimated jammer location, and the corresponding score is compared against a configured threshold to determine whether the observed condition is consistent with a jamming attack.

To increase robustness during online operation, the spatial GLRT is complemented by a window-limited CUSUM (WL-CUSUM) mechanism. While the GLRT evaluates the spatial degradation pattern at each monitoring instance, WL-CUSUM follows the recent evolution of the detection scores over a bounded observation window. An alarm is raised when either the instantaneous spatial analysis identifies a sufficiently strong jamming-compatible pattern or the temporal analysis indicates persistent accumulated evidence of degradation.

Once a jamming attack is detected, **CENS03** provides the estimated jammer location together with the estimated SINR degradation affecting the legitimate communication link. The **Activation** function forwards this information to **RAN control**, which uses the estimated degradation and the path-loss information associated with the detected jammer position to determine and apply a transmission power increase. This PHY-level adaptation aims to compensate for the jamming-induced link degradation and restore the required communication quality.

Through this sequence, Flow UC1_2_01 demonstrates how monitored SNR observations are transformed into hypothesis-test-based jamming detection, online temporal confirmation, attacker localisation and adaptive PHY-layer mitigation, thereby closing the PLCL through an appropriate RAN control action. The implementation shared for this flow contains the baseline monitoring logic, spatial GLRT, WL-CUSUM, jammer localisation and SINR-drop estimation steps underpinning this procedure.

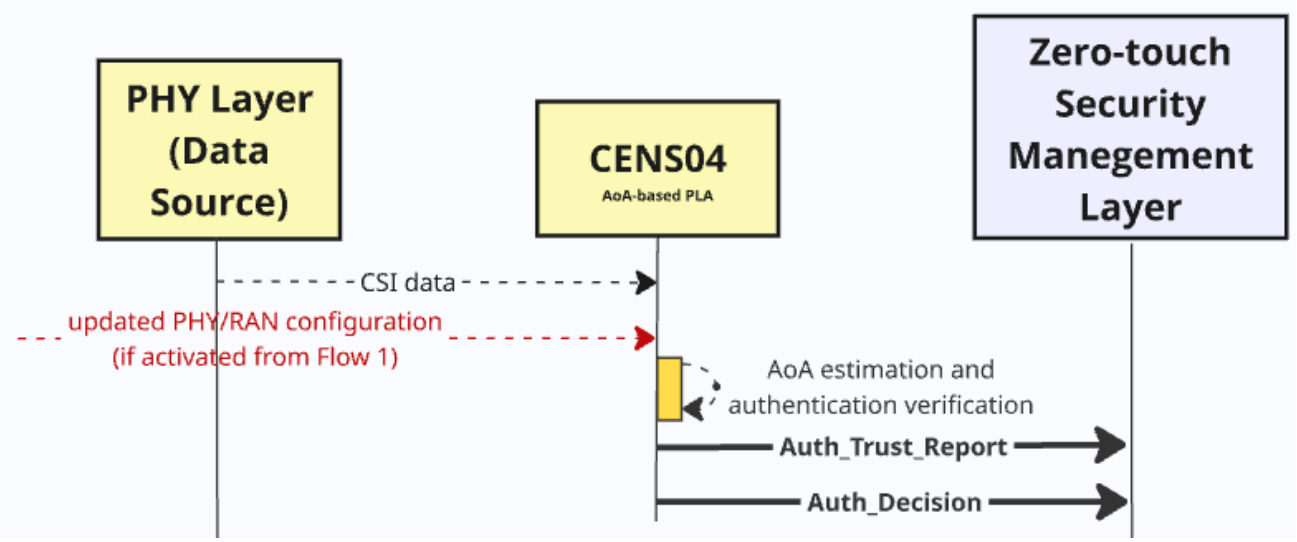

Figure 4.7 Flow UC1_2_02 – Mutual authentication

Flow UC1_2_02 – Mutual authentication

- **Objective:** Physical-layer mutual authentication through AoA-based PLA, including the assessment of its trustworthiness limits under potential impersonation attacks and the investigation of complementary authentication solutions for challenging radio conditions.

As illustrated in Figure 4.7 the flow uses physical-layer observations to determine whether a transmission is consistent with the trusted spatial signature of a legitimate communicating node, achieves the mutual authentication objective of UC1.2 through CENS04 – AoA-based PLA.

The flow starts with CSI data provided by the **PHY layer** under the current active PHY/RAN configuration. If Flow UC1_2_01 has previously detected a jamming condition and activated PHY-layer adaptation, the authentication procedure operates under the updated PHY/RAN configuration resulting from that mitigation action. Thus, the authentication function remains applicable under the adapted communication conditions produced by the preceding trustworthiness evaluation flow.

Within **CENS04**, the authentication procedure follows an enrolment and verification process. During enrolment, the AoA associated with a legitimate node is estimated from its CSI observations and stored as its trusted physical-layer reference. During verification, the AoA of a subsequent transmission is estimated and compared against the enrolled reference and its accepted angular region. A transmission whose observed angular signature is consistent with the legitimate reference is accepted, whereas a transmission arriving from outside the accepted angular region is identified as a potential spoofing or impersonation attempt. In the considered prototype implementation, the AoA of each transmission is estimated from the available CSI observations using a MUSIC-based procedure in a digital massive-MIMO receiver setting, enabling the assessment of robustness against spatial impersonation attempts.

Following the verification procedure, **CENS04** provides an Auth_Decision to indicate whether the transmitting node is accepted or rejected. In addition, it produces an Auth_Trustworthiness_Report, capturing the reliability limits of the selected authentication mechanism. In particular, AoA-based authentication is robust against adversarial nodes located outside the legitimate node's accepted angular region, but its discriminating capability is inherently limited when legitimate and adversarial nodes share the same angular direction with respect to the receiving array. This condition is therefore explicitly considered as part of the trustworthiness assessment of the authentication decision.

Beyond the AoA-PLA mechanism activated in the present prototype flow, WP5 has investigated complementary solutions and analytical extensions for strengthening physical-layer authentication and characterizing its trustworthiness under more challenging conditions. In particular, enhanced sensing-based authentication extends AoA-based verification with additional CSI-derived features, such as received signal strength (RSS) and Time-of-Flight (ToF), to improve the detection of proximal impersonation attempts in cases where legitimate and adversarial nodes are spatially close or exhibit similar angular signatures. This direction is especially relevant for addressing situations in which AoA alone provides insufficient discrimination between legitimate and adversarial transmissions.

[1] Further WP5 investigations consider architectural and propagation-driven extensions of the AoA-PLA principle ([R6G6-D52] and [R6G6-D53]). Cooperative base-station authentication combines AoA observations obtained from multiple receiving points, reducing the ambiguity introduced when a legitimate and an adversarial node are aligned with respect to a single receiver. Near-field authentication exploits the richer spatial signature available when both angle and distance influence the received signal, enabling discrimination even when two nodes share the same AoA but are located at different distances from the receiving array. These approaches provide potential extensions of the authentication flow for deployments in which the single-receiver, far-field AoA assumption is not sufficient. In parallel, WP5 activities also assess the reliability and implementation efficiency of AoA-based authentication. A spoofing-oriented analytical study derives decision-level reliability measures for AoA-PLA under model mismatch, including authentication thresholds and spoofing-detection, false-alarm and misdetection behaviour. Moreover, a learning-based AoA estimation approach investigates low-complexity, repeated CSI-to-AoA inference as an alternative to the computational processing required by conventional MUSIC-based estimation. Finally, the analysis of AoA-based authentication in analog-array architectures identifies receiver configurations in which AoA spoofing may become feasible, further contributing to the characterisation of the trustworthiness limits of AoA as an authentication feature.

These complementary investigations are not activated as separate components in the current UC1.2 prototype flow, which focuses on **CENS04-based AoA authentication** using CSI observations and MUSIC-based AoA estimation in a digital massive-MIMO setting. Rather, they provide additional mechanisms and trustworthiness evidence that can support physical-layer authentication capability within the PLCL.

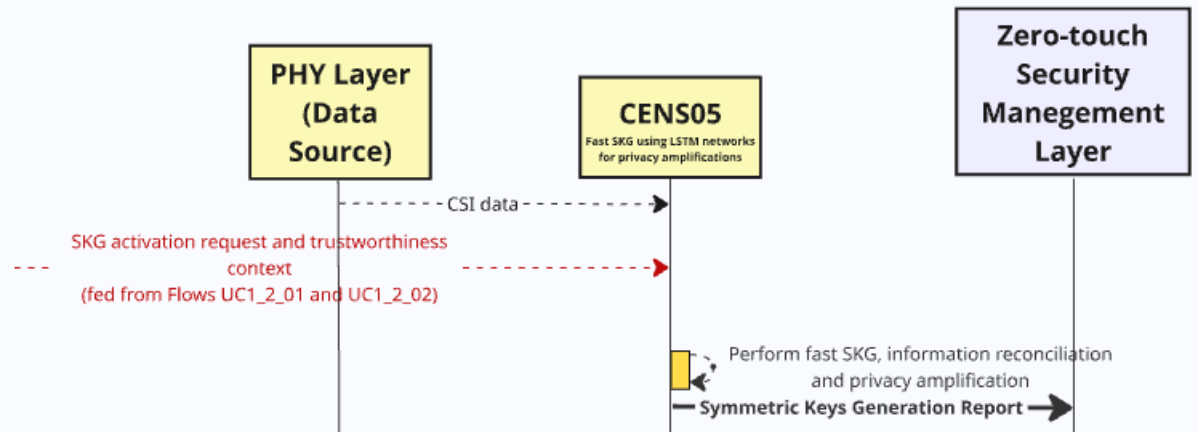

Figure 4.8 Flow UC1_2_03 – (Fast) Secret key generation

Flow UC1_2_03 – (Fast) Secret key generation

- **Objective:** PHY Layer fast SKG using observed CSI in massive MIMO OFDM under one-wavelength eavesdropping attacks. LSTM networks are considered for privacy amplification to enable fast SKG.

As illustrated in Figure 4.8, Flow UC1_2_03 triggers the SKG schemes mainly through the component **CENS05**. Using **PHY-layer** observations and the relevant previous information, CENS05 performs the full SKG framework including information reconciliation and privacy amplification.

The flow starts with the CSI data from the **PHY Layer** block. Both uplink and downlink CSI data are subsequently converted into binary sequences using uniform quantization. To obtain the binary sequences, Gray coding is applied to reduce bit mismatches caused by small variations in the estimated CSI and to improve reconciliation.

To reconcile the mismatched binary sequences between Alice and Bob, Slepian-Wolf decoding is implemented using Polar codes with cyclic redundancy check (CRC) as the error correction code. Alice then generates a syndrome using the Slepian-Wolf coding approach and transmits it over a public channel to Bob. This enables Bob to decode and recover Alice's key with high reliability despite the channel mismatch using Polar codes. Concerning the considered Polar codes for the reconciliation block, we use Gaussian approximation based polar code construction, which assumes that log-likelihood ratios (LLRs) remain Gaussian distributed through the polar transformation. The idea of Gaussian approximation is to evolve the densities and estimate the precise reliability of each channel. These reconciled sequences form the input to privacy amplification.

Following quantization and information reconciliation, privacy amplification is applied to compress the reconciled bit sequence into a shorter secret key while removing any residual information potentially available to the eavesdropper. This stage aims to ensure that the final secret key remains unknown to the eavesdropper despite channel correlation and public reconciliation information. To enable real-time SKG implementations, a supervised learning approach based on a Long Short-Term Memory (LSTM) network is proposed for the privacy amplification process. Conditional min-entropy (CME) is employed to quantify the amount of information potentially leaked to the eavesdropper. CME is estimated using the Fast Blackbox Leakage Estimation Algorithm Using Machine Learning (F-BLEAU). The LSTM is trained using the CME values produced by the F-BLEAU algorithm together with SKG parameters. Once trained, the LSTM predicts suitable hashing rates with very short inference times, significantly reducing the computational cost of privacy amplification while maintaining high prediction accuracy.

Another solution for privacy amplification is to adopt a conservative hashing rate deliberately chosen to be lower than the estimated requirement to ensure a sufficient security margin. Once the hashing rate is fixed, privacy amplification is performed by applying a cryptographic hash function to the reconciled bit sequence. Specifically, the reconciled vectors are processed using the AES-128 hash function, producing a secret key whose length is consistent with the selected compression rate. A Davies-Meyer compression function is used to build the hash function. At any point, two blocks of 128 bits are considered, and the current hash value is also XORed with the output of that iteration.

In summary, the SKG process in Flow UC1_2_03 consists of quantization, information reconciliation, and privacy amplification to generate secure shared secret keys from wireless channel measurements. The LSTM-based privacy amplifications framework enables real-time operation while reducing computational complexity.

4.1.2.4 Prototype in Use

UC1.2 is realised through **Prototype 4: Physical and Sensing Layer Trustworthiness and Resilience**, which demonstrates the operation of the **Physical Layer Closed Loop (PLCL)** in the considered indoor massive-MIMO setting. The prototype validates the closed-loop operation of the

monitoring, analysis and actuation stages through the scenario-specific use of **CENS01**, **CENS03**, **CENS04** and **CENS05**.

- **Flow UC1_2_01** realises the PHY-layer trustworthiness assessment chain: **CENS01** processes CSI-based observations to derive channel metrics, while **CENS03** uses these metrics for anomaly detection, jammer localisation and SINR-drop estimation; when degradation is detected, the actuation function requests PHY control actions, such as transmission power adaptation.
- The resulting channel metrics are subsequently exploited in **Flow UC1_2_02**, where **CENS04** performs AoA-based physical layer authentication and assesses its trustworthiness limits under impersonation attempts.
- Finally, **Flow UC1_2_03** activates **CENS05** for fast secret key generation and privacy amplification, using PHY-layer observations and the relevant information produced by the preceding flows.

In this manner, Prototype 4 integrates monitoring, trustworthiness analysis, authentication, secret key generation and PHY control within the PLCL, providing the demonstrable implementation of the three UC1.2 flows.

Prototype 5 further enhance this scenario through the interaction of the PLCL with the Zero-Touch Security Management Layer, enabling the exchange of upper-layer security context and the propagation of PHY-layer alerts towards broader security orchestration functions.

4.2 Use Case 2: Automatic threat detection and mitigation in 6G-enabled IoT environments

The Use Case 2 focuses on particular attacks against IoT/smart device environments and related remediations based on Security Services (SSe) composed of Security Closed-Loops (S-CLs) and Security Functions (SFs) that are deployed by the Zero-Touch Security Platform (ZTSP) to establish and maintain a Security Posture in the target infrastructure. The base idea is that not all the attacks cause anomalies detectable by only analysing the network traffic. In this regard, the UC proposes 3 scenarios with increasing complexity, where the security solution is based on the joint analysis of network traffic (using state-of-the-art tools and novel AI/ML techniques) and devices' alerts and telemetry. Use Case 2 was originally envisioned to showcase the ROBUST-6G end-to-end security architecture featuring integration with the Global Model Repository (GMR) and active remediation controls operating down at the physical layer. However, during the WP6 finalization phase documented in this deliverable, the consortium introduced the concepts of Prototypes. UC2 now operates as a highly focused validation for Prototype 2, demonstrating the multi-layer defender capabilities provided by the ZTSP. Prototype 5 (the Master Prototype) is explicitly designed as the cross-prototype demonstrator, bringing together and showcasing the integration of Prototypes 1, 2, 3, and 4 into a unified system. To avoid structural redundancy and overlapping validation scopes across the Master Prototype and the Prototype 2 and Use Case 2, the interaction with the GMR and the orchestration of the physical layer security closed-loop were extracted from the local boundaries of Use Case 2. Consequently, Use Case 2 has been refactored into a highly focused, localised architectural extension of Prototype 2 (Multi-Layer Zero-Touch Defender). Instead of validating a single standalone loop, UC2 now provides an iterative demonstration of closed-loop scalability, showing how a foundational zero-touch automation block can scale seamlessly across multi-loop and multi-tenant 6G configurations:

1. Scenario 1 (UC2.1): Validates a single, standalone security closed loop embedded inside an orchestrated security service instance. This Scenario completely validates Prototype 2, showing how a Security Service Level Agreement (SSLA) is translated using the support of semantic reasoning and GenAI4SOAR into a deployable and orchestrable Security Service capable of establishing and

maintaining a Security Posture via zero-touch S-CL execution driven by pervasive monitoring (provided by the PMP), standardised decision (using CACAO IRPs), and agnostic execution (via OpenC2 actuators).

2. Scenario 2 (UC2.2): Extends the framework into a dual-closed-loop architecture, showcasing how the Security Closed Loop Management (CNXW04) block coordinates a multi-stage delegation mechanism between an investigative loop and a resolutive loop.
3. Scenario 3 (UC2.3): Scales the architecture to a multi-tenant edge configuration with 5 distinct decentralised loops operating across independent fields, coordinated via an escalation hierarchy where a centralised, external long loop manages and overrides localised contradictions using the short-loop inputs extended with contextual data (meteorological information).

4.2.1 Scenario 1 – Device violation to cause an economic harm (a)

4.2.1.1 Scenario Overview and Objectives

This baseline scenario demonstrates the operation of a standalone, automated security closed loop embedded within a managed security service. The setting is a smart office environment where building infrastructure, specifically the Heating, Ventilation, and Air Conditioning (HVAC) system, is centrally managed via an IoT platform. The threat vector involves an attacker performing a brute-force attack to compromise the HVAC management logic, aiming to manipulate the environment to cause physical discomfort and trigger excessive, unauthorised energy consumption. To counter this, a zero-touch orchestration engine proactively deploys a security service that continuously monitors the environment. Upon detecting anomalous behavioural patterns, the analysis stage verifies the intrusion. The system then automatically formulates a remediation strategy without human intervention, which dictates a sequence of defensive actions: the compromised device is isolated, the offending user account is suspended, a hardened password policy is instantly enforced, and a secure reactivation process is initiated for the customer account.

4.2.1.2 Position within the Architecture

The Use Case makes use of the components belonging to the Zero Touch Security Management Layer of ROBUST-6G system architecture reported in Figure 2.1, in line with Prototype 2.

With reference to Figure 2.1, the layer's functionalities covered by this use case are the following:

- **Security Orchestration:** manage the security service provisioning request and consumes other functionalities to deploy the security service in the scenario. Main functionalities of this layer include the semantic-driven Security Service composition and the GenAI assisted Incident Response Plan (IRP) definition.
- **Security Closed-Loop Management:** provisions the security closed-loop upon a request from the ZTSO. The S-CL is indeed realised as a cloud application deployable into a target cloud environment and tailored for the security service.
- **Resource Orchestrator:** manages the resources of the different target cloud environment and request the explicit provisioning of the applications (including the S-CL stages).
- **Threat detection/prediction & incident response:** indicates the Security Service itself, with the monitoring capabilities provided by the PMP and the automated response

4.2.1.3 Functional Flows Description

The end-to-end operations of this standalone loop are formally captured in the accompanying sequence diagrams, which detail the component interactions across three main functional flows. Flow 1 (Proactive Security Enforcement Flow), shown in Figure 4.9, depicts the E2E orchestration of the Security Service starting from an SSLA. In steps 1-10, starting from the SSLA, the infrastructure context (set of Security Functions available/deployable/configurable and the available Target

Environments in the Infrastructure) are collected and provided to the GenAI4SOAR module for the generation of the IRP. Steps 8-9 represents the optional validation of the selected Security Functions and the generated IRP from an admin aimed to solve possible LLM hallucinations. In this Scenario, the resulting Security Service is composed of (i) two Security Functions: the PMP, already deployed on the environment, and a ThingBoard OpenC2 Consumer that is necessary to perform tailored actuations on the Target Environment; and (ii) a rule-based Security closed loop that delegates the monitoring and analysis to the PMP, while the decision and execution stages behave respectively as CACAO Playbook manager and interpreter. Steps 10-17 of the figure represents the deployment of the Security Functions and the S-CL.

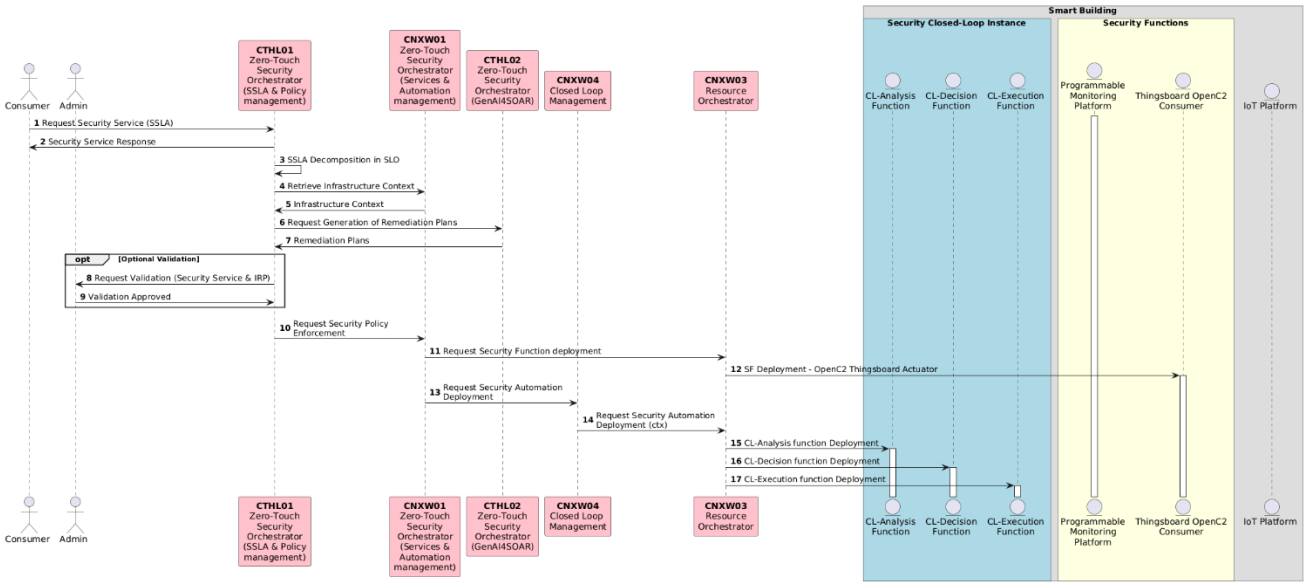

Figure 4.9 UC2.1 Proactive Security Enforcement Flow

Flow 2 (Threat Detection combining Network and IoT Data), depicted in Figure 4.10, details the continuous data collection and threat detection phase of the operationalised Security Service. As established in Flow 1, the Programmable Monitoring Platform (PMP) assumes the responsibilities of both the monitoring and analysis stages for this rule-based loop. In steps 1-3, the active Security Closed Loop (S-CL) stage dynamically configures the PMP via its Configuration Manager API. This includes an API call to deploy the necessary security tools, instructing the PMP to simultaneously ingest and analyse network traffic, utilizing T-shark and Snort 3 equipped with community rules, and monitor ThingsBoard IoT alerts. Following this configuration, the PMP exposes the endpoint for the alerts [CDL-PMPc], and the S-CL analysis stage starts listening on the communication bus (e.g., Kafka topics). Once fully operational, the PMP continuously correlates the ingested data streams. During this phase, an attacker executes a credential stuffing attack, successfully logging into the user dashboard to improperly start the HVAC device and cause economic harm. In the final step, upon identifying this anomalous pattern that violates the established security rules, the PMP generates a threat alert and notifies the S-CL Decision function via the communication bus, successfully triggering the reactive remediation process

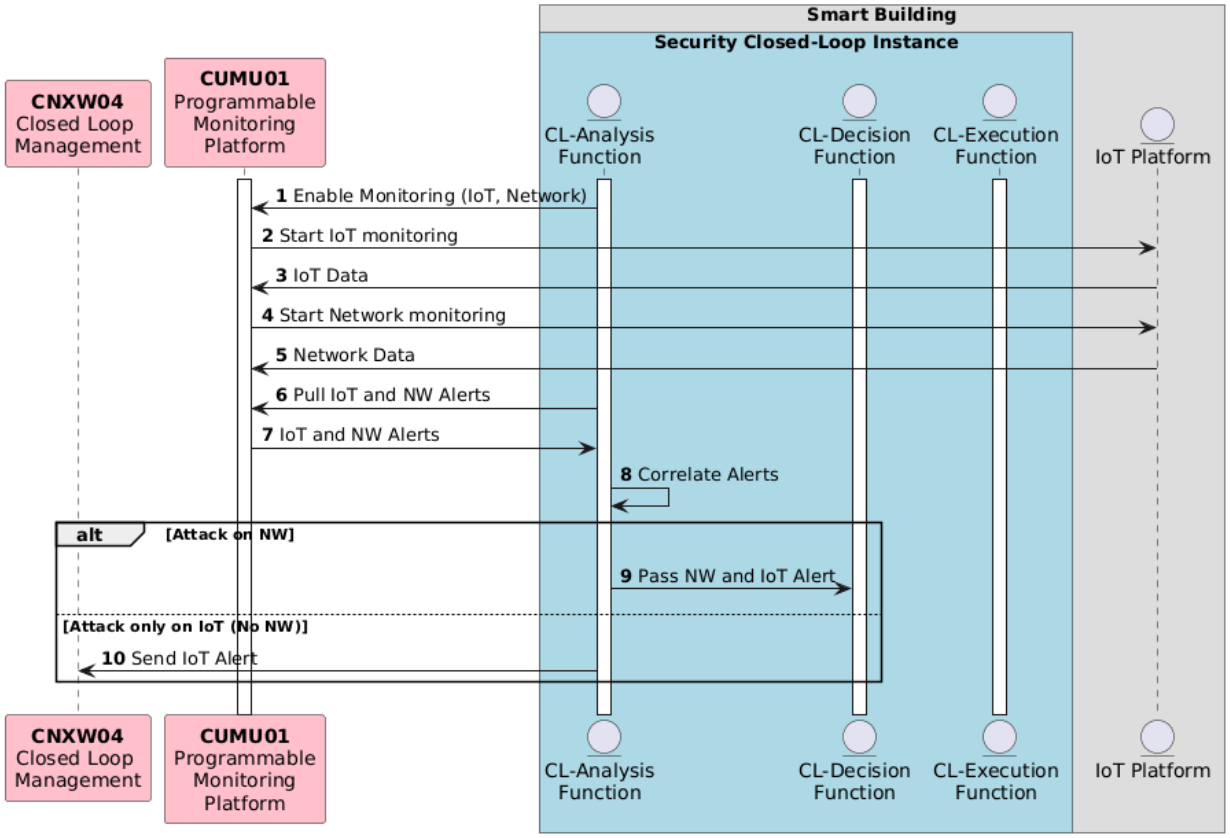

Figure 4.10 UC 2.1 Threat Detection combining Network and IoT Data

Flow 3 (Threat Mitigation via reactive plan execution) depicted in Figure 4.11, illustrates the automated incident response enforced by the security closed loop. In step 1, the sequence begins when the CL-Decision function is notified of the threat alert from the PMP. In steps 2-4, the decision stage queries the S-CL Knowledge base to retrieve the specific CACAO Incident Response Playbook (IRP) generated during the proactive deployment phase, validates it, and parses it for the execution stage. Steps 5-10 represent the continuous playbook execution loop, where the CL-Execution function behaves as a CACAO playbook interpreter and an OpenC2 producer. For every command in the playbook, it translates the actions into OpenC2 commands and pushes them via an MQTT broker (step 5) to the ThingsBoard OpenC2 Consumer, which retrieves the command and actuates the remediation directly on the Target Environment (steps 6-7). The loop autonomously enforces the required mitigation sequence by unassigning the compromised HVAC device, suspending the affected user account, enforcing a hardened password policy, and issuing an account reactivation email.

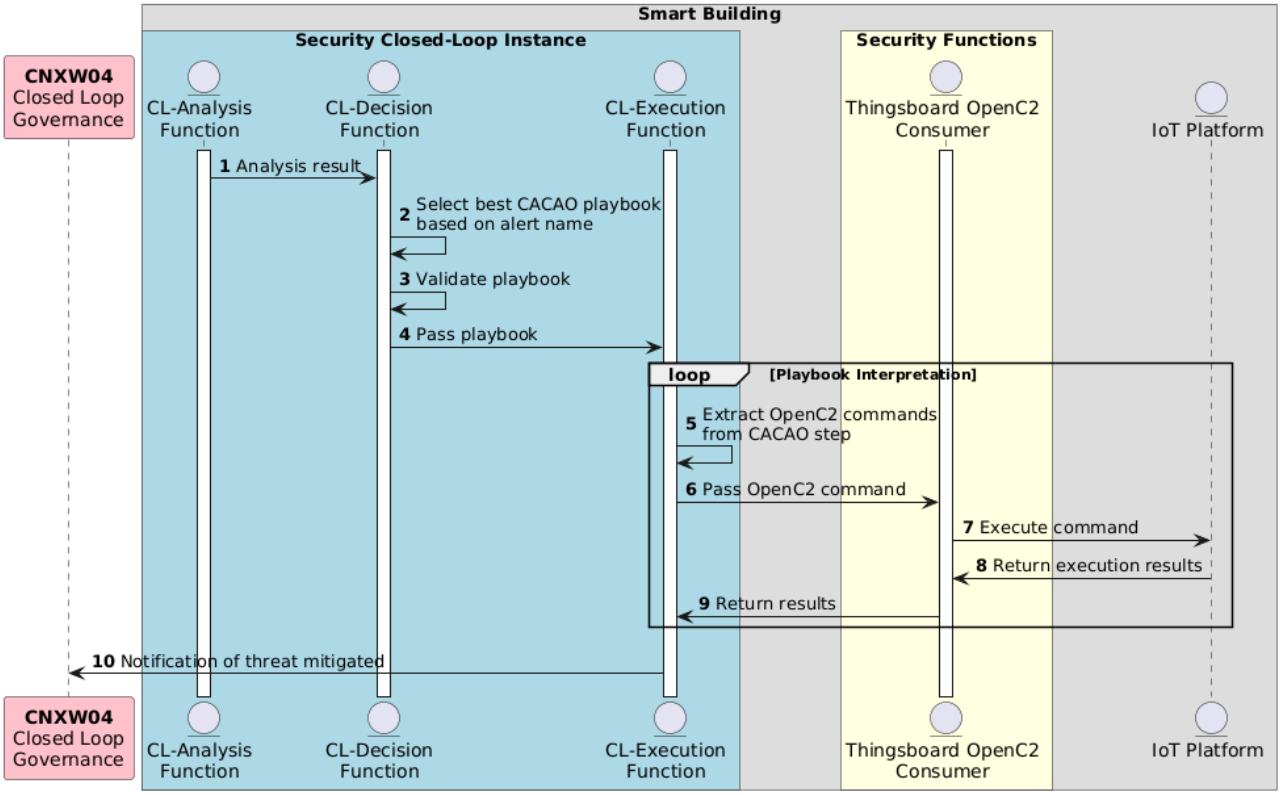

Figure 4.11 UC2.1 Threat Mitigation via reactive plan execution

4.2.1.4 Prototype in Use

This scenario demonstrates Prototype 2, in particular the Security Service composition, GenAI-based IRP Generation, Security Service Provisioning, and S-CL automation for Zero-Touch security posture management.

4.2.2 Scenario 2 – Fraudulent usage of device resources

4.2.2.1 Scenario Overview and Objectives

This scenario addresses stealthy, resource-hijacking threats within a smart building, specifically attackers who exploit IoT smart lamps to perform unauthorised cryptocurrency mining (cryptojacking). Because the compromised devices appear to function normally from a user perspective, and anomalous power consumption cannot, on its own, definitively indicate a cyberattack, the scenario introduces a two-stage closed-loop architecture in which a lightweight investigative loop dynamically triggers the on-demand deployment of a deeper, AI-driven resolutive loop. The primary objective is to demonstrate this dynamic escalation between closed loops, realised through service re-orchestration, together with the dynamic integration of an external AI algorithm for detection. To avoid redundancy with UC2.1, the scenario does not re-demonstrate the proactive Security Service composition phase (SSLA decomposition and remediation-plan generation); it begins from the deployment of the initial investigative security service and focuses strictly on closed-loop runtime execution, the dynamic escalation / re-orchestration mechanism, and the use of advanced AI for detection.

4.2.2.2 Position within the Architecture

This scenario exercises the components of the Zero-Touch Security Management Layer, leveraging the ZTSO, the S-RO, and the S-CL Manager. It expands its architectural footprint by establishing a direct dependency on an AI-driven Threat Detection Module. Within the boundaries of this specific scenario, the AI model is considered an already-registered Security Function (SF) in the ZTSO's Catalogue Manager and a pre-onboarded orchestrable artefact in the S-RO, but it is instantiated only on demand, when the investigative loop escalates. The complete end-to-end process explaining how this AI model can be initially fetched from the Trustworthy and Sustainable AI Services Layer, specifically from the Global Model Repository (GMR), and registered into the orchestrator is deferred to Prototype 5, where this cross-layer integration is extended and fully explained.

4.2.2.3 Functional Flows Description

Flow 1 (Investigative Service Deployment), depicted in Figure 4.12, shows the deployment of the initial investigative security service. The Zero-Touch Security Orchestrator (ZTSO, CNXW01) requests, from the S-CL Manager (CNXW04) and the S-RO (CNXW03), the deployment of a lightweight, two-stage investigative closed loop composed of a CL-Analysis and a CL-Decision function; the individual stages are instantiated by the S-RO in the target infrastructure. At this stage neither the resolutive loop nor the on-demand Security Functions (the ThingsBoard OpenC2 actuator and the AI Threat Detection Module) are deployed: they are provisioned later, only if the investigative loop escalates. The PMP is assumed to be already present and active, and no inter-loop coordination function is established.

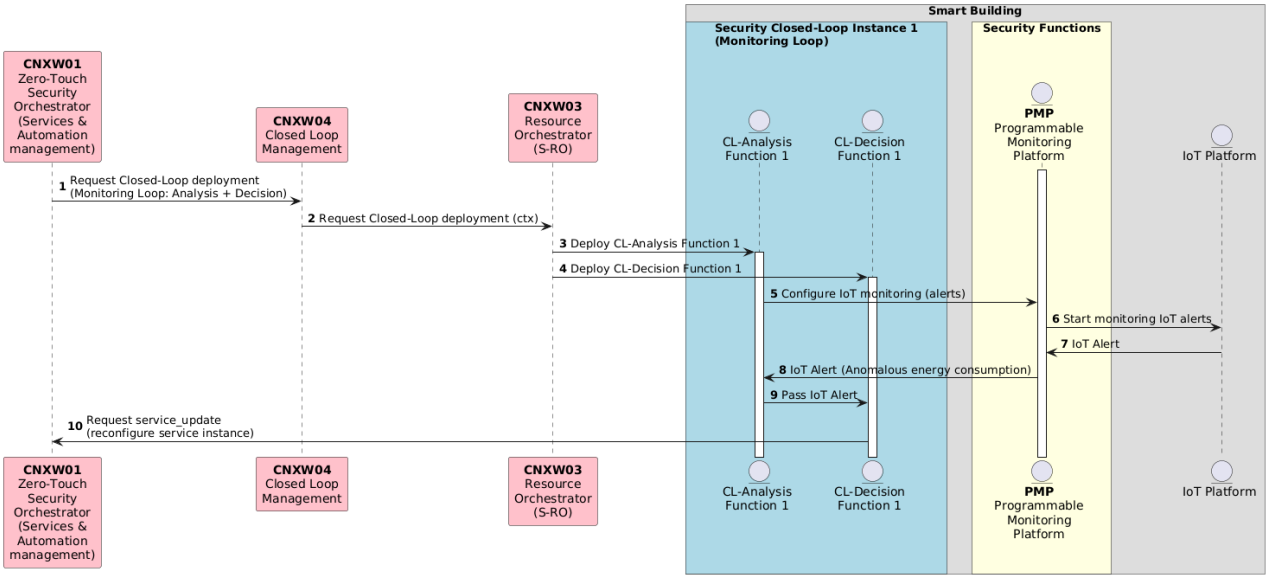

Figure 4.12 UC2.2 Investigative Loop

Flow 2 (Anomaly Detection, Dynamic Escalation and AI-Driven Mitigation), depicted in Figure 4.13, captures the runtime processing and the dynamic escalation from the investigative loop to the resolutive loop. Execution begins with the investigative loop: its CL-Analysis function instructs the PMP to collect IoT baseline telemetry (e.g., energy consumption) and listens for the resulting IoT alerts. Upon detecting a persistent anomalous power spike, the CL-Decision function, applying a conservative approach to avoid operational disruption from false positives, does not remediate directly; instead, it issues a service_update (reconfiguration) request to the ZTSO (CNXW01). This dynamic re-orchestration terminates the investigative loop and deploys, on demand, the ThingsBoard

OpenC2 actuator and the AI Threat Detection Module (via the S-RO) together with the resolutive closed loop composed of CL-Analysis, CL-Decision and CL-Execution functions (via the S-CL Manager and the S-RO). Once active, the resolutive loop reconfigures the PMP strictly as a telemetry pipeline, using its Flow Module to extract structured network flows (e.g., CICFlowMeter format) without performing rule-based analysis. The CL-Analysis function retrieves these raw network flows from the communication bus and feeds them directly to the pre-deployed AI Threat Detection SF. The AI algorithm analyses the network flows and confirms the cryptojacking attack that the power anomaly had only suggested, issuing a definitive threat verdict. The CL-Decision and CL-Execution stages then retrieve the corresponding CACAO playbook and translate it into OpenC2 commands to automatically enact the remediation sequence (e.g., isolating the compromised smart lamp by revoking its access credentials and issuing a secure Over-The-Air (OTA) firmware update).

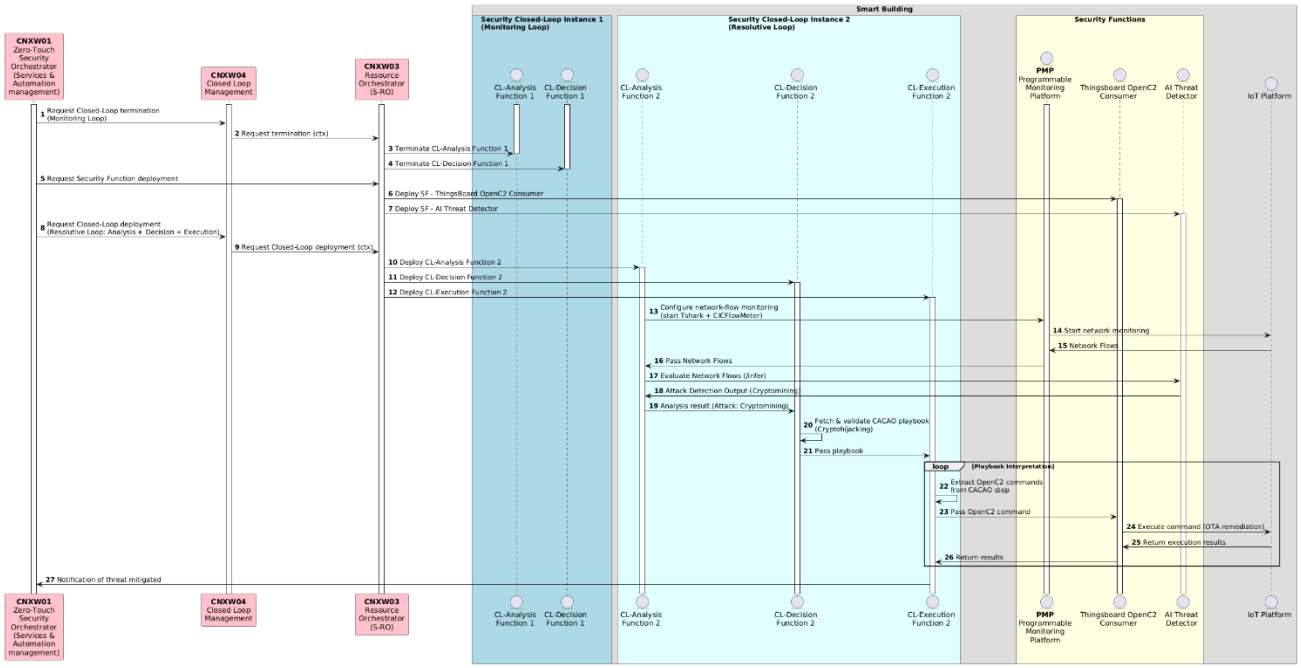

Figure 4.13 UC2.2 Resolutive Loop

4.2.2.4 Prototype in Use

This scenario demonstrates Prototype 2, with a focus on the core closed-loop lifecycle and execution and on the dynamic escalation between Security Closed Loops realised through on-demand service re-orchestration (service_update) driven by the investigative loop and enacted by the ZTSO (CNXW01), the S-CL Manager (CNXW04) and the S-RO (CNXW03), rather than through a static inter-loop coordination function. Additionally, it demonstrates part of Prototype 5 by using an AI/ML model for threat detection that was previously fetched from the GMR and made available to the ZTSP as an orchestrable Security Function; how the GMR models are made available to the ZTSP is demonstrated in Prototype 5 itself.

4.2.3 Scenario 3 – Device violation to cause an economic harm (b)

4.2.3.1 Scenario Overview and Objectives

This final scenario validates the scalability, conflict-resolution, and multi-tenant capabilities of the ROBUST-6G architecture within a distributed smart agriculture environment. It features 5 distinct smart farms, each operating its own independent IoT Platform and localised Security Closed Loop

(S-CL) system to govern local sensor arrays and actuators. The threat vector involves an attacker manipulating local temperature and humidity sensor readings to trigger unauthorised water irrigation, aiming to cause environmental damage and severe financial losses. Because an isolated temperature anomaly could simply be the result of a localised weather event or a legitimate sensor failure, relying strictly on a single loop could lead to severe false positives. Therefore, the primary objective of this scenario is to demonstrate the coordination of a centralised long loop that oversees the 5 peripheral internal loops. To strictly focus on the multi-loop coordination and cross-tenant data enrichment, this scenario does not encompass the proactive Security Service composition and deployment phases. It assumes that the localised S-CLs and the long loop are already fully operational. The workflow highlights how the long loop utilises the Data Fabric as an intermediate integration layer for data exchange, governance, and security context enrichment with external meteorological information.

4.2.3.2 Position within the Architecture

This scenario comprehensively exercises the Zero-Touch Security Management Layer, strongly relying on the S-CL Manager (CNXW04) for hierarchical loop governance. Crucially, it expands the architectural footprint to integrate the Data Management Platform. To enable the long loop to act as a centralised conflict-resolution mechanism, the architecture leverages the Data Fabric (CTID01) and Data Governance (CTID02) modules. The Data Fabric serves as the overarching aggregator, securely collecting telemetry from the 5 independent smart farms while enforcing strict access policies via the Data Governance module. By cross-referencing local IoT anomalies against the broader data lake and external meteorological APIs, the architecture demonstrates its capability to harmonize decisions across multi-tenant edge domains securely.

4.2.3.3 Functional Flows Description

Flow 1 (Loops and Coordination Deployment), depicted in Figure 4.14, details the end-to-end proactive deployment of the multi-tiered closed-loop architecture. The execution is driven by the Closed Loop Management component (CNXW04). Initially, CNXW04 deploys the core S-CL entities: it instantiates the three stages of the Master Loop (Master-Monitor, Master-Analysis, and Master-Decision). Subsequently, CNXW04 orchestrates the deployment of the peripheral S-CLs across the 5 Smart Farms, logically assigning them to their respective proximity groups: Zone A (Farms 1 and 2) and Zone B (Farms 3, 4, and 5). To finalize the hierarchical setup, CNXW04 distributes a specific CL Coordination Loop ID to the local decision functions of each of the 5 farms. This crucial step explicitly registers the independent peripheral loops with the centralised Master Loop, establishing the secure communication channels required for runtime escalation. Finally, as the last step in the workflow, the Data Fabric subscribes to the coordination bus to capture the streaming data. It intercepts both the local decisions generated by the short loops and the coordinated actions from the long loop. Through a continuous semantic lifting process, the Data Fabric transforms these raw data payloads into a unified knowledge graph. This ensures that all information flowing across the multi-farm system is semantically described, interoperable, and easily queryable, laying the foundation for the advanced reasoning detailed later in this document.

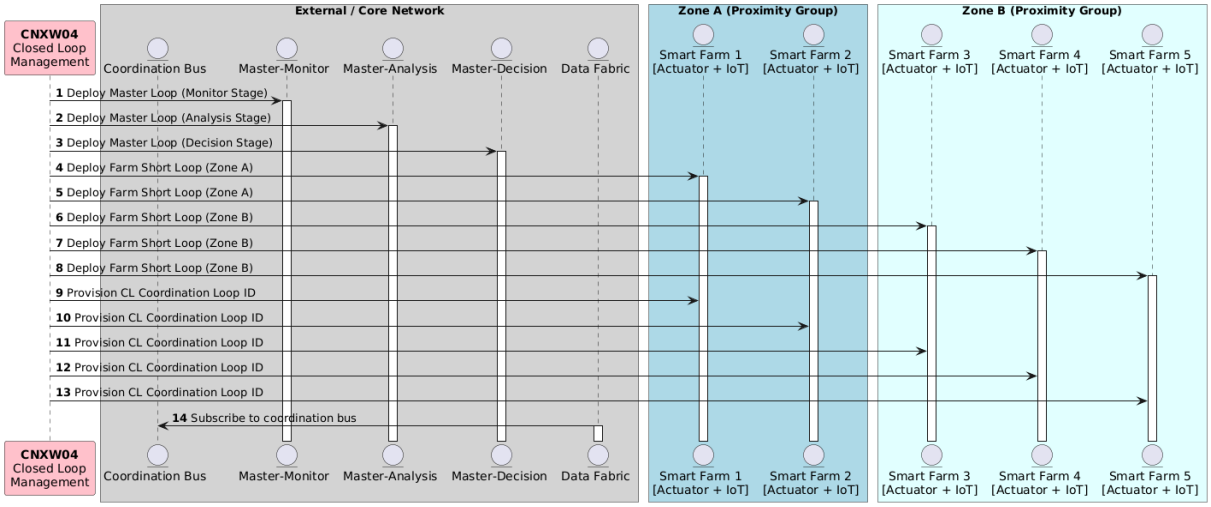

Figure 4.14: UC2.3 Loops and Coordination deployment

Flow 2 (Loops Execution and Conflict Resolution), depicted in Figure 4.15, captures the continuous runtime processing and the collaborative conflict resolution between the local loops and the Master Loop. During the synchronous execution phase, all 5 Smart Farms independently complete their local monitoring, analysis, and decision cycles. Instead of immediately executing potentially conflicting physical remediations, each farm publishes its localised decision output (e.g., "Irrigate" or "Do Not Irrigate") directly to the coordination bus. The Master-Monitor continuously grabs this aggregated decision payload from the coordination bus and passes it to the Master-Analysis stage for a global correlation check. The Master-Analysis cross-references the localised decisions based on their geographical zones. If the correlation reveals inconsistencies between different zones (e.g., Zone A deciding to irrigate while Zone B does not), the Master-Analysis evaluates this as a valid regional microclimate difference, reporting an "OK" status to the Master-Decision to accept the local decisions without requiring a global override. Conversely, if severe inconsistencies are detected within the same proximity group (e.g., Farm 1 requests irrigation while adjacent Farm 2 does not), the Master Loop, using the aggregated data enriched with meteorological information, identifies the compromised local loop and formulates a harmonized remediation strategy, effectively overriding the manipulated sensor to halt unauthorised irrigation.

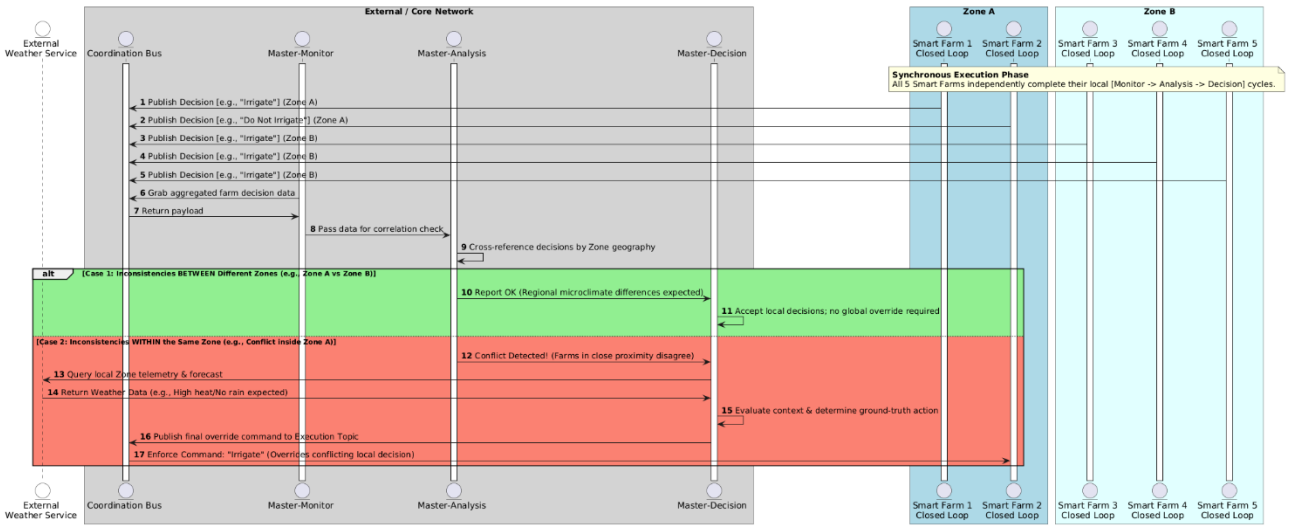

Figure 4.15: UC2.3 Loops execution and conflict resolution

4.2.3.4 Prototype in Use

This scenario demonstrates the multiple closed-loop management capabilities of Prototype 2, managing the core execution and hierarchical delegation between localised internal loops and the centralised long loop via the S-CL Manager. Furthermore, it demonstrates part of Prototype 3 by leveraging the Data Fabric (CTID01) and Data Governance (CTID02) components as an intermediate layer for data collection, cross-tenant aggregation, and meteo enhancement.

4.3 Use Case 3: Security Capabilities Exposure (NetSecaaS)

This section covers the exposure of ROBUST-6G security capabilities to third-party applications through the NetSecaaS Gateway. It builds on the prototype description provided in Section 3.3 of this deliverable and on the intermediate validation results reported in D6.2, and gives a flow-by-flow account of how the prototype is exercised in the federated testbed.

4.3.1 Scenario Overview and Objectives

Use Case 3 addresses the scenario in which an external actor — typically an application developer, an enterprise tenant, or a vertical service provider — needs to consume security capabilities offered by the ROBUST-6G platform, but does not have, and is not expected to develop, the network-security expertise required to configure or query those capabilities directly. The objective of the use case is to demonstrate that the ROBUST-6G platform can expose its security capabilities through a Network-Security-as-a-Service (NetSecaaS) interface, characterised by a level of abstraction that frees the consumer from the technical details of the underlying components and aligns the interaction with the developer-friendly, outcome-oriented style of the GSMA Open Gateway and CAMARA initiatives.

The scenario validates two complementary patterns of consumption. On the one hand, third parties retrieve security-relevant information generated within the platform — for example explainability reports for AI/ML-driven security events, access-governance metadata, or the catalogue of available security capabilities. On the other hand, third parties configure or trigger security actions, in particular through a simplified SSLA endpoint that translates a high-level intent into the corresponding orchestrated response. In both cases, every interaction is mediated by intent-based REST APIs, protected by authentication and authorisation enforced by the Data Governance plane, and traceable through audit logs maintained by the platform.

4.3.2 Position within the Architecture

Within the ROBUST-6G reference architecture introduced in Figure 2.1 of Section 2.1 of this deliverable, Use Case 3 is linked to the Exposure Framework as its primary anchor point and involves slice of the rest of the system. The NetSecaaS Gateway (CTID03) sits at the boundary of the architecture and is the only externally visible component; everything behind it remains hidden from third-party consumers. Authentication, authorisation and policy enforcement are delegated to the Data Governance plane (CTID02), and data retrieval is served from the Data Fabric (CTID01), which holds a semantic representation of the outputs produced by the underlying security capabilities. When the scenario exercises the configuration direction, the Security Orchestrator (CTHL01) of the Zero-Touch Security Management Layer is reached through the Gateway, in order to communicate also with the Zero-Touch Security Orchestration components (CNXW01).

In terms of internal capabilities consumed by the scenario, two are central in the current phase: the XAI services (CUCD03 and CEBY03), whose explainability outputs are exposed through one of the

four functional flows, and the Security Orchestrator (CTHL01), which executes the SSLA artefacts produced by the Transformation Function.

4.3.3 Functional Flows Description

Three functional flows have been defined for Use Case 3, covering the main interaction patterns exposed by the NetSecaaS Gateway: security-capabilities discovery (UC3_01), XAI analytics data exposure (UC3_02), and simplified SSLA enforcement (UC3_03).

Flow UC3_01 enables a third party to retrieve security-relevant information about access policies and audit trails through a CAMARA-style API exposed by the NetSecaaS Gateway. It enables a third party to discover the security capabilities currently exposed by the platform, along with their status and the high-level parameters that the consumer can specify when invoking them. The structural pattern of the flow is the same depicted in Figure 4.16: a request issued to CTID03 is first evaluated for authorisation by CTID02, and, if authorised, is mapped by the Transformation Function into a query against CTID02 which holds the catalogue of capabilities described according to the platform's access policies. The Data Governance returns the subset of capabilities that the requester is entitled to discover, and the Gateway delivers the catalogue back to the consumer.

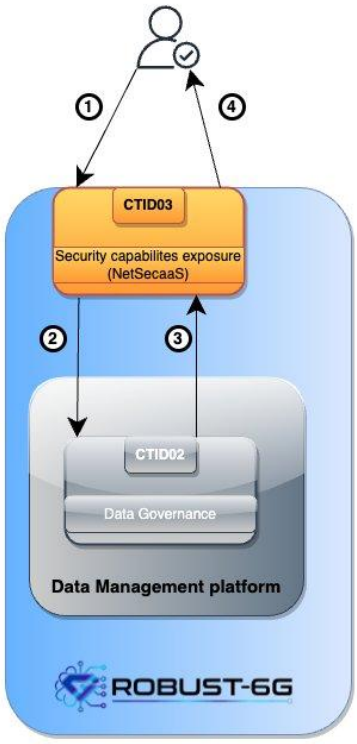

Figure 4.16:UC3_01 — Security Capabilities discovery data exposure

Throughout the flow, the interactions between CTID03 and CTID02 follow the access-control pattern established in the Data Management Platform: authentication is performed against the platform's Identity Provider, authorisation decisions are computed by the Policy Decision Point, and enforcement is applied transparently at the API-gateway boundary.

Flow UC3_02 exposes the historical and on-demand explainability reports produced by the XAI services (CUCD03 and CEBY03) for AI/ML-driven security events. The flow operates in two phases. A preliminary, offline phase has the XAI services continuously analysing raw security information, generating explainability artefacts alongside the detection or prediction outputs, and pushing both into the Data Fabric (CTID01) through the declarative ingestion mechanism like the ones described

in Section 3.3.3. The result is a progressively populated Knowledge Graph of XAI-enriched security data, ready to be consumed by third parties.

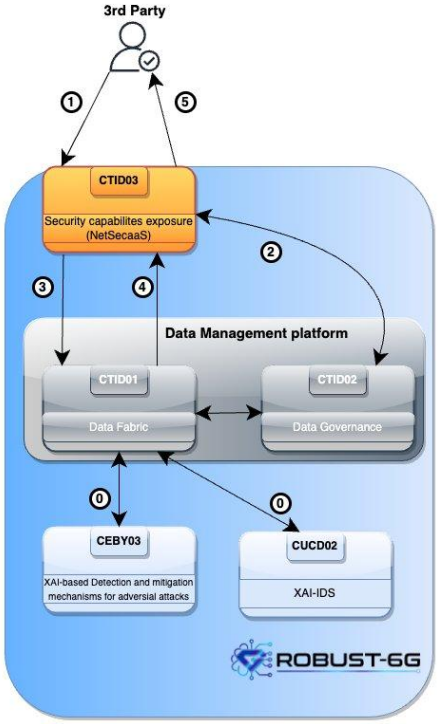

Figure 4.17: UC3_02 — XAI Analytics Data

The on-line phase, depicted in Figure 4.17, is triggered by a third-party request specifying, for instance, the type of attack of interest or the time window for which explainability reports are sought. The request is received by the NetSecaaS Gateway (CTID03), which delegates authorisation to the Data Governance plane (CTID02). Once authorisation is granted, the Transformation Function maps the request parameters into a SPARQL query against the Knowledge Graph and the Data Fabric returns the matching subset of XAI-derived security data. The Gateway then delivers the resulting dataset, together with the associated explanations, back to the requester. The flow therefore exposes both the platform's analytical conclusions and the rationale supporting them, contributing to the transparency and auditability requirements of the use case.

Flow UC3_03, as shown in Figure 4.18 , enables a third party to submit a high-level request expressing the security outcome it wishes to enforce, for instance, the activation of a specific protection mechanism on a given resource. CTID03 forwards the request to CTID02 for authorisation; upon a positive decision, the Transformation Function converts the high-level intent into a structured Security Service Level Agreement (SSLA) that captures the parameters and constraints of the requested operation. The SSLA is then forwarded to the Security Orchestrator (CTHL01), which interprets it and executes the corresponding security actions on the underlying ROBUST-6G capabilities, optionally interacting with the Zero-Touch Security Orchestration components (CNXW01) to align with broader closed-loop activities. Optionally, ROBUST-6G security capabilities can be configured by leveraging National Institute of Standards and Technology (NIST) SP 800-53 Rev. 5 control families as a reference data source for baseline security controls. Once enforcement is complete, CTHL01 returns the status back to CTID03, which delivers a simplified, consumable response to the requester. If authorisation fails at any step, the flow is interrupted with an access-denied response and no SSLA is generated, preserving the integrity of the orchestration plane.

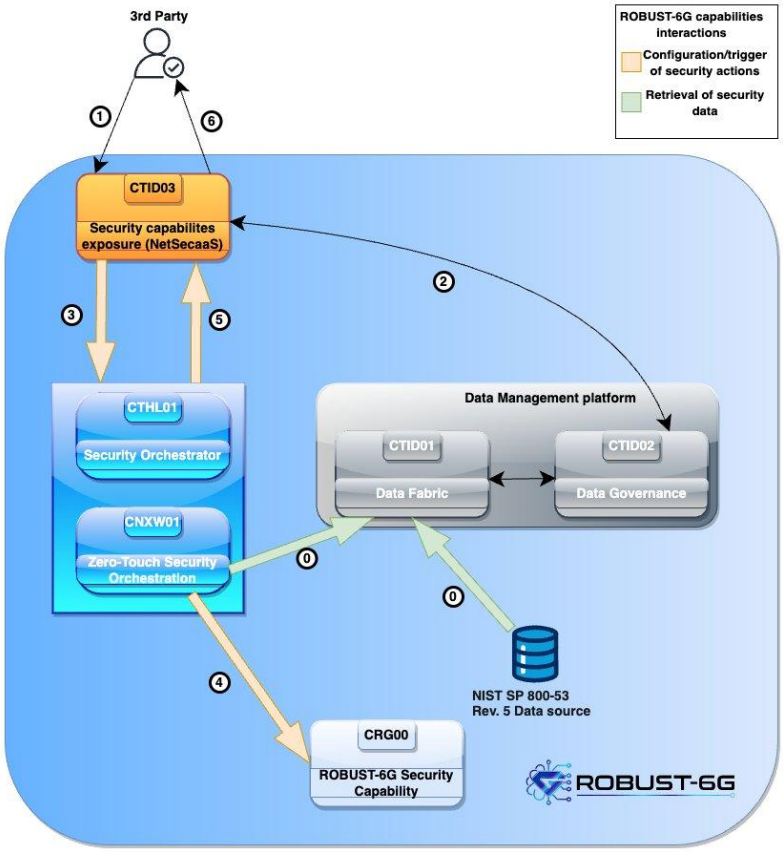

Figure 4.18: UC3_03- Simplified SSLA enforcement

Through this flow the cornerstone capability of the use case is demonstrated: a third party with no detailed knowledge of the internal security mechanisms is nevertheless able to enforce a meaningful security outcome through a single, intent-based API call, while the platform takes responsibility for translating the intent into the corresponding orchestrated actions.

4.3.4 Prototype in Use

Use Case 3 is realised by Prototype 3 (NetSecaaS Gateway), whose components and overall architecture are documented in Section 3.3 of this deliverable. In the context of this scenario, the prototype is deployed across two partner testbeds, as shown in Figure 4.19. The bulk of the components — namely the NetSecaaS Gateway itself (CTID03), the Data Fabric (CTID01), the Data Governance plane (CTID02), and the XAI services (CUCD03 and CEBY03) — are deployed in the TID testbed (TTID01); the Security Orchestrator (CTHL01) is hosted in the Nextworks testbed (TNXW01) and is reached over a secure interconnection from TTID01. The Gateway is the only component directly accessible from outside the TID premises; all other components are reachable exclusively through authorised API calls intermediated by CTID03.

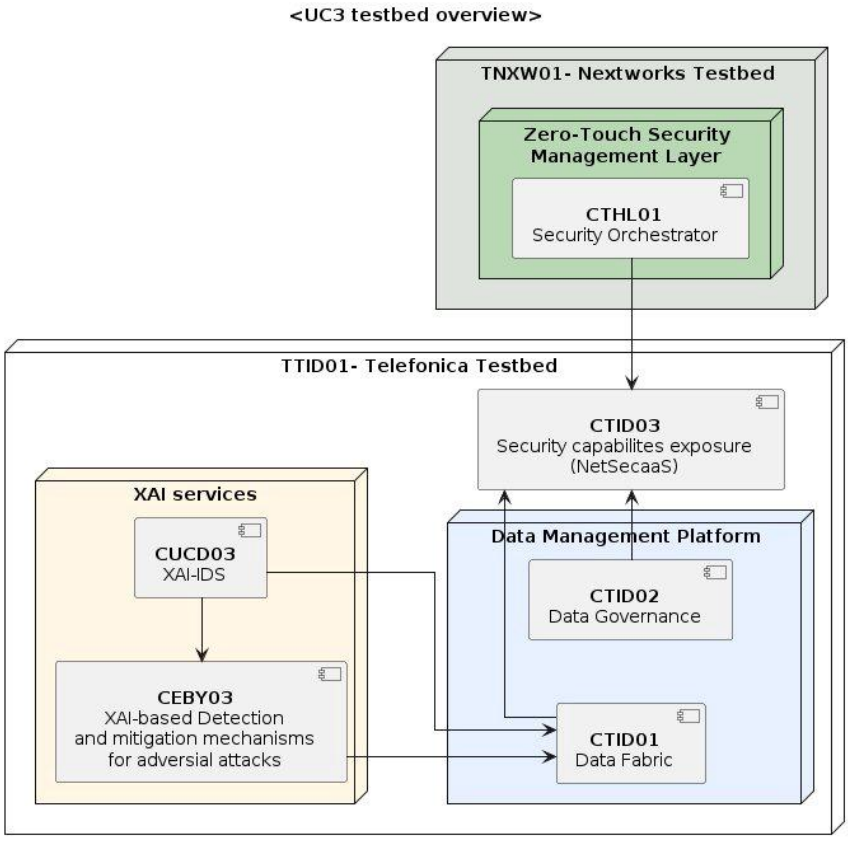

Figure 4.19 Deployment of Prototype 3 across the federated testbeds for UC3

From a deployment perspective, the Data Fabric and Data Governance components run as services within a Kubernetes cluster, while the NetSecaaS Gateway and the XAI services are instantiated as standalone Docker containers, all within TTID01. The interconnection with TNXW01 is established to enable the communication between CTID03 and CTHL01 required by Flow UC3_03. Each functional flow of the scenario maps onto a specific subset of these components: UC3_01 exercises the CTID03–CTID02 path; UC3_02 exercises CTID03–CTID02–CTID01 with the XAI services as upstream data producers; UC3_03 exercises the CTID03–CTID02–CTHL01 path. The status of the integration of each flow at the intermediate validation stage is reported in D6.2 and is consolidated, alongside the final KPI attainment results, in Chapter 5 of this deliverable.

5 Overall Validation Summary

This chapter brings together, in a single place, the validation evidence produced across the whole of the ROBUST-6G project, so that the platform can be assessed as one system rather than as a set of separate demonstrators. It consolidates the results obtained for the five prototypes and the three use cases described in Chapters 3 and 4 and organises them along two complementary levels. The first level is the prototype validation assessment in Section 5.1, which records, for each prototype, what was demonstrated, which objectives were met, and which limitations remain, together with the supported TRL. The second level is the use-case KPI attainment in Section 5.2, which reports the quantitative outcomes per use case and scenario, treating each KPI as a dedicated test with explicit traceability between the tested functionality, the measured evidence, and the corresponding DoA target.

These two levels feed the project-wide assessment in the remainder of the chapter. Section 5.3 verifies the project Global Objectives defined in the DoA at the integrated project level, mapping each objective to the use-case KPIs and source-deliverable evidence that substantiate it. Section 5.4 then draws the threads together into an overall evaluation of the degree to which the DoA commitments have been met. The supporting appendices provide the full component and flow validation summaries, the KPI and Objective Traceability Matrix, and the per-prototype TRL assessment, so that every figure reported here can be traced back to its origin.

The evidence consolidated here is drawn both from the WP6 integration activities and from the technical results reported by the contributing work packages: WP3 for trustworthy and sustainable AI, WP4 for zero-touch security management, and WP5 for physical and sensing layer security. Where a KPI was first established and measured within a source work package, this chapter validates it again at platform level on the integrated prototypes and, for the physical and sensing layer, on real measured data, rather than restating it in isolation. The cross-prototype integration that turns these individual capabilities into a single, coherent security platform is demonstrated by the Master Prototype (Prototype 5) and is reflected throughout the assessment that follows.

5.1 Prototype Validation Assessment

5.1.1 Prototype 1: Trustworthy AI

Prototype 1 validates the ROBUST-6G Trustworthy AI capability as an end-to-end decentralised AI lifecycle for 6G security services. The prototype is centred on the ROBUST-6G Decentralised Federated Learning (DFL) Framework and the Global Model Repository (GMR). Together, these components support privacy-preserving model training, attack-aware aggregation, explainability artefact generation, metric collection, and controlled reuse of validated AI assets by other ROBUST-6G components.

The validation assessment builds on the WP3 KPI evidence reported in D3.4 and focuses on the integrated prototype behaviour expected in WP6: successful deployment of the DFL nodes, execution of collaborative training under benign and adversarial settings, collection of learning and system metrics, and registration of the resulting models and metadata in the GMR. In line with the D3.4 KPI analysis, the validation also distinguishes between KPI targets already demonstrated at WP3 level and those that require final consolidation in the WP6 integration environment.

5.1.1.1 Validation Setup

The prototype was validated using a server deployment of the ROBUST-6G DFL Framework. The framework supports both CPU and GPU execution, with GPU support intended for more demanding PyTorch training workloads. Each participant is instantiated as an independent DFL node configured with its own participant profile, local dataset partition, training parameters, aggregation method, communication settings, and optional adversarial behaviour. The GMR is deployed as a complementary service stack composed of the repository API, PostgreSQL persistence, and web/database inspection interfaces.

The validation setup exercises the complete DFL lifecycle. First, nodes are initialised and connected according to the configured peer-to-peer topology. Node discovery, liveness monitoring, and message propagation are handled through the framework heartbeating and gossiping mechanisms. Each node then trains locally on its assigned data partition, exchanges model updates with neighbouring peers, and performs decentralised aggregation. Depending on the validation run, the aggregation method may use baseline FedAvg or robust aggregation such as Krum, which is included to mitigate Byzantine and poisoning-style model updates. Model parameters may be encrypted before exchange, and metrics are collected throughout execution.

The GMR is used as the prototype governance and evidence layer. At the end of training and evaluation steps, the DFL nodes persist model weights, learning metrics, system metrics, logs, and related metadata locally and, when repository upload is enabled, upload them to the GMR through the REST API. This validates the role of the GMR as the single access point for versioned trustworthy AI assets and as the bridge between decentralised training nodes and the wider ROBUST-6G platform.

5.1.1.1.1 Testbed Configuration

The testbed configuration consists of multiple DFL participant nodes running the ROBUST-6G framework in containerised deployment mode. Each node is configured with:

- A list of neighbouring peers used for decentralised message and model exchange.
- A local training configuration, including the number of rounds, number of epochs, accelerator type, and model family.
- An aggregation configuration, including FedAvg, Krum, ADMM, or Douglas-Rachford splitting (DRS) where applicable.
- Optional adversarial configuration, such as no attack, label flipping, data poisoning, or model poisoning.
- Metric and repository settings controlling local evidence collection and GMR upload.

The default framework configuration supports encrypted model exchange, periodic heartbeat messages, gossip-based dissemination of model parameters and protocol messages, configurable aggregation timeouts, and periodic resource reporting. This setup allows the prototype to be validated both as an isolated WP3 component and as a reusable AI-service layer for integration with the wider ROBUST-6G architecture.

5.1.1.1.2 Datasets

The framework includes modular dataset support and can be extended with new datasets and models. For the validation of Prototype 1, the relevant implemented datasets are:

- MNIST, used for controlled validation of the DFL workflow, robustness mechanisms, and explainability artefact generation.

- TON-IoT, used to exercise the framework against an IoT/cybersecurity-oriented dataset aligned with the 6G-enabled IoT threat-detection context.
- CIC-IDS2017, available in the framework for intrusion-detection experimentation with SNN-based models.

The DFL data module supports both Independent and Identically Distributed (IID) and non-IID partitioning strategies. Non-IID operation is particularly important for the ROBUST-6G validation because it reflects realistic 6G edge deployments where nodes observe heterogeneous traffic, device behaviour, and attack distributions. This setup enables comparison between standalone local models and collaboratively trained DFL models, which is required for the KPI target on federated accuracy improvement over isolated local training.

5.1.1.1.3 Integration Details

Prototype 1 integrates four main technical blocks:

- DFL node runtime: manages node initialisation, local training, peer-to-peer exchange, aggregation, evaluation, and lifecycle control.
- Communication and gossip protocol: supports the decentralised propagation of control messages, model parameters, aggregation status, and peer state.
- Robust and decentralised aggregation layer: supports FedAvg as baseline aggregation and robust/decentralised alternatives such as Krum, ADMM, and DRS, enabling validation under benign and adversarial training conditions.
- GMR integration: stores and exposes trained models, training and validation metrics, test metrics, system metrics, logs, and metadata through the repository API.

The validation also covers the interaction between the DFL framework and the trustworthiness mechanisms reported in D3.4. These include attack simulation in the training loop, poisoning-aware model evaluation, privacy-preserving update handling, membership-inference mitigation evidence, and explainability artefacts such as SHAP-based outputs. The resulting artefacts can be stored in or associated with the GMR so that model consumers, dashboards, trustworthiness evaluators, or other ROBUST-6G prototypes can retrieve both the model and its supporting evidence.

5.1.1.1.4 Inputs and Outputs

The main validation inputs are:

- Dataset partitions assigned to DFL nodes, including IID and non-IID configurations
- Model and training hyperparameters, such as model type, number of rounds, epochs, learning rate where configured, batch size, and accelerator mode.
- Aggregation configuration, including the selected algorithm and Krum fault-tolerance parameter where applicable.
- Adversarial settings, including the number or percentage of malicious nodes, attack type, poisoning ratio, and target labels where applicable.
- Repository and tracking settings controlling local metric logging and GMR upload.

The expected validation outputs are:

- A converged collaboratively trained model or set of model versions.
- Learning metrics such as accuracy, precision, recall, F1-score, loss, and confusion matrices where supported by the model.
- Robustness evidence under adversarial settings, including poisoning detection and mitigation outcomes.
- Privacy and inference-risk evidence, including membership-inference mitigation results where the corresponding module is exercised.

- System metrics such as CPU usage, memory usage, disk usage, network counters, and thread count.
- Model, metric, log, and explainability entries stored locally and, when enabled, registered in the GMR.

5.1.1.2 Validation Outcomes

UC1.1 Validation Outcomes

The validation demonstrates that Prototype 1, Trustworthy AI, successfully executes the complete DFL lifecycle and integrates with the GMR under both benign and adversarial conditions. The validation is structured into three main operational phases: DFL and Attack Configuration, Robust Training and Model Poisoning Mitigation, and Robust Time-Series Model Explainability and GMR Registration.

DFL and Attack Configuration

The validation process begins with the deployment and initialisation of the containerised DFL framework. The frontend web interface, running via Uvicorn, is utilised to define the topology, parameterise the learning process, and initiate peer-to-peer training. Through the graphical configuration dashboard, the participant nodes are registered, the peer-to-peer network topology is mapped, and the training configurations are defined:

- **Dataset and Model Configuration**: The **TON-IoT** dataset is selected, pairing it with the corresponding **CyberNet** model. The underlying TONIoTDataset class partitions the cybersecurity log data, which includes network flow parameters such as connection state, protocols, and packet counts, using non-IID partitioning to simulate heterogeneous traffic conditions at edge nodes.
- **Participant Nodes**: Five containerised participant nodes are deployed in a fully-connected topology.
- **Adversarial Configuration**: Node 4 is explicitly configured as a malicious node tasked with executing a **Model Poisoning** attack. The attack parameters are set with a poisoning strength of 10000 and a poisoning percentage of 1, 100% of local epochs poisoned.
- **Aggregation Algorithm**: **Krum** is selected in the aggregation dropdown, with the Byzantine tolerance parameter set to 1, enabling the system to handle one malicious update per aggregation step.
- **Accelerator**: **GPU** is selected in the accelerator dropdown to leverage hardware acceleration, via PyTorch CUDA, for the training workloads across participant nodes.
- **Explainability**: The **time_series** explainability method is selected, which triggers the shats_explainer module at the end of each round.

As shown in Figure 5.1, the frontend provides a user-friendly DFL configuration screen. Here, the target datasets can be selected, the accelerator set to GPU, the adversarial parameters for Node 4 configured, and the connection status of all containerised participant nodes verified, ensuring they are in a ready state before launching the execution.

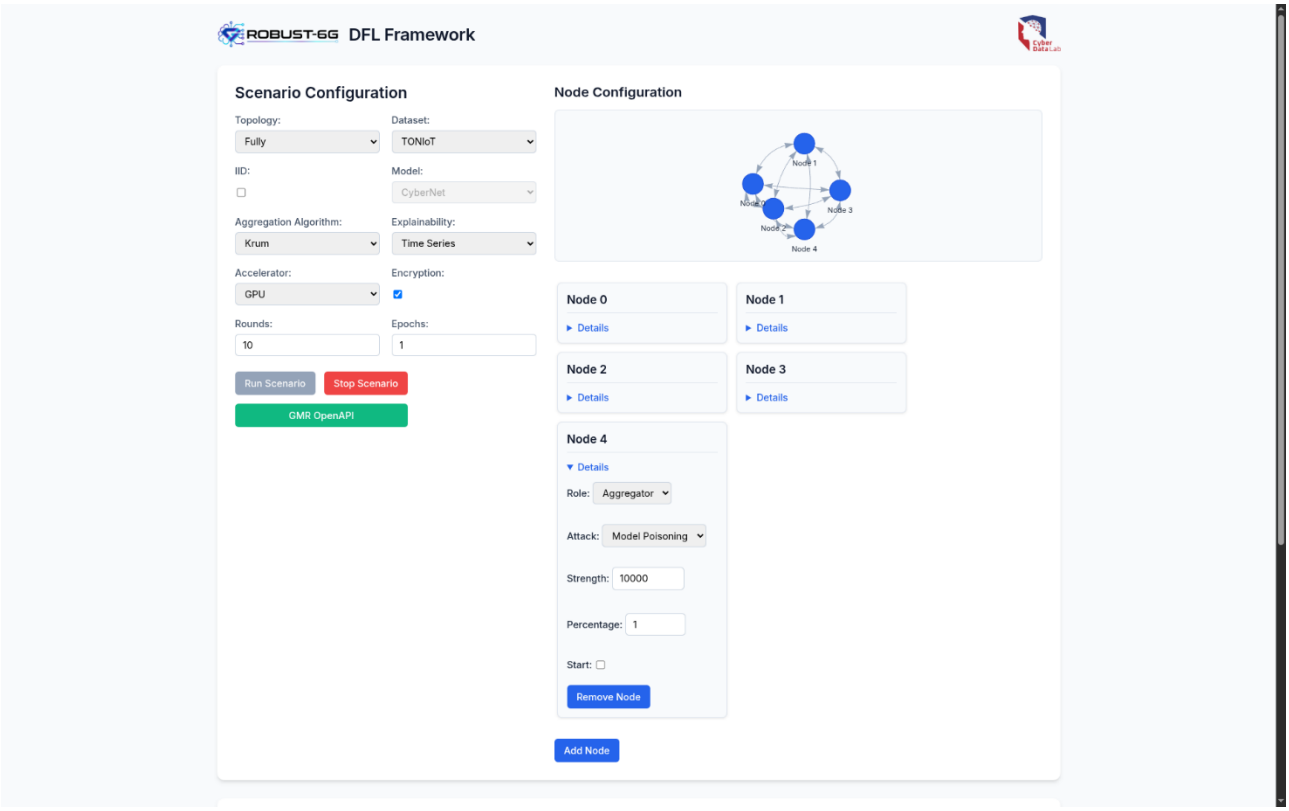

Figure 5.1 DFL Framework Frontend and Attack configuration

Once the configurations are submitted, the DFL framework generates the respective participant manifests and initialises the communication protocol. Peer discovery is achieved via a dedicated heartbeat thread running on each node heartbeater.py, which periodically broadcasts status messages to maintain the active neighbour list. The gossip protocol in gossiper.py then governs update dissemination, choosing a random subset of neighbours per round based on the gossip_models_per_round and gossip_models_frec parameters to balance network load.

Robust Training and Model Poisoning

Once the training job starts, the benign nodes perform local training rounds using the CyberNet model accelerated by GPU. Simultaneously, Node 4 executes local training epochs but maliciously corrupts its weight matrices by injecting parameter poisoning with a strength multiplier of 10000 before serialising and exchanging its update. Model updates are serialised and encrypted using symmetric AES-256-GCM encryption before P2P exchange, with the payloads fragmented into blocks governed by the block_size parameter.

During the model aggregation step, the robust aggregation layer intercepts the received weights. Instead of using baseline FedAvg, Krum calculates the pairwise Euclidean distances between the updates received from all participants. For a given set of updates, Krum computes a score for each node as the sum of distances to its closest neighbours, adjusting for the defined Byzantine tolerance parameter. The update with the lowest cumulative distance score is selected as the representative parameter set for the next round. Because Node 4's poisoned updates deviate significantly in the parameter space due to the high poisoning strength, they yield exceptionally high scores. Krum successfully isolates and discards the updates from Node 4, aggregating only the benign updates from the other participant nodes.

As depicted in Figure 5.2, Figure 5.3, and Figure 5.4, the DFL frontend and terminal outputs provide live progress indicators, training execution steps, and aggregation logs. The console logs print the

distance matrices and explicitly declare the detection and exclusion of the poisoned updates from Node 4.

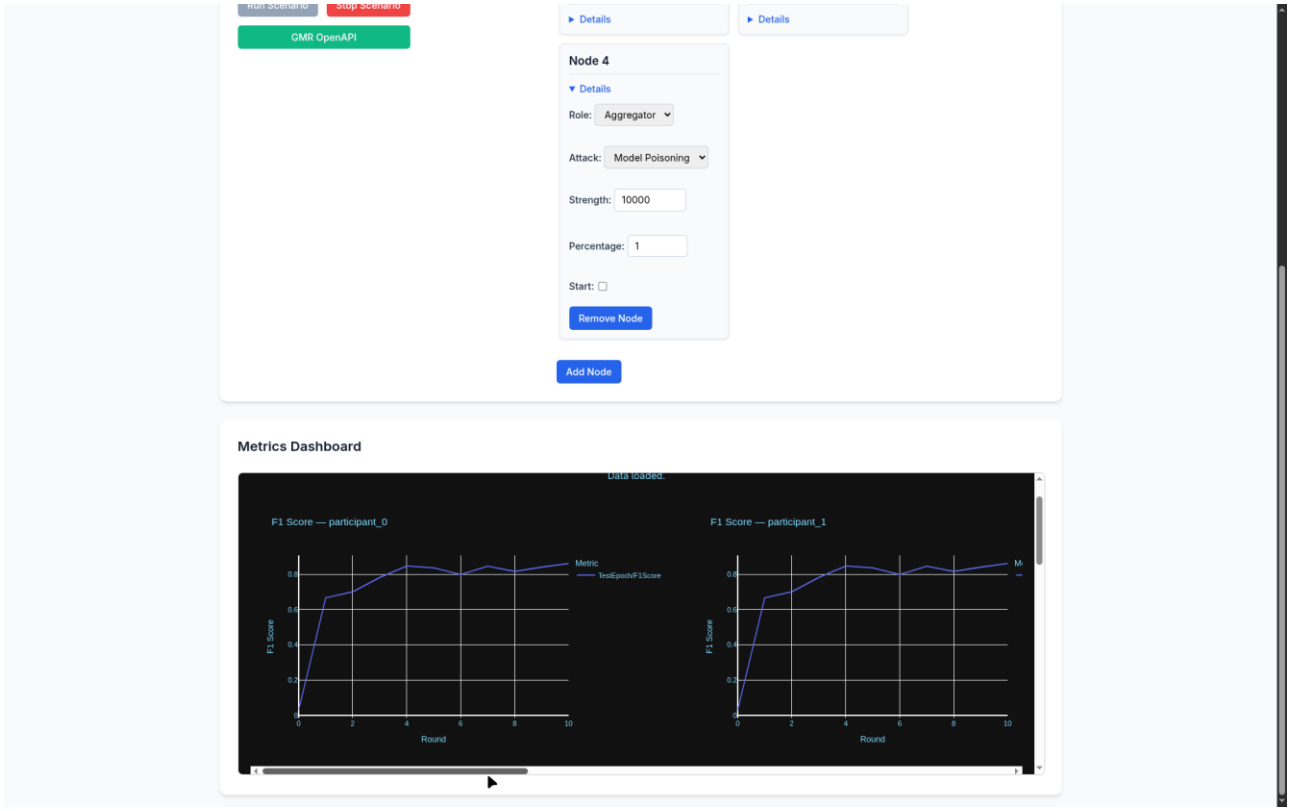

Figure 5.2 DFL Framework Frontend Active Training Progress Metrics

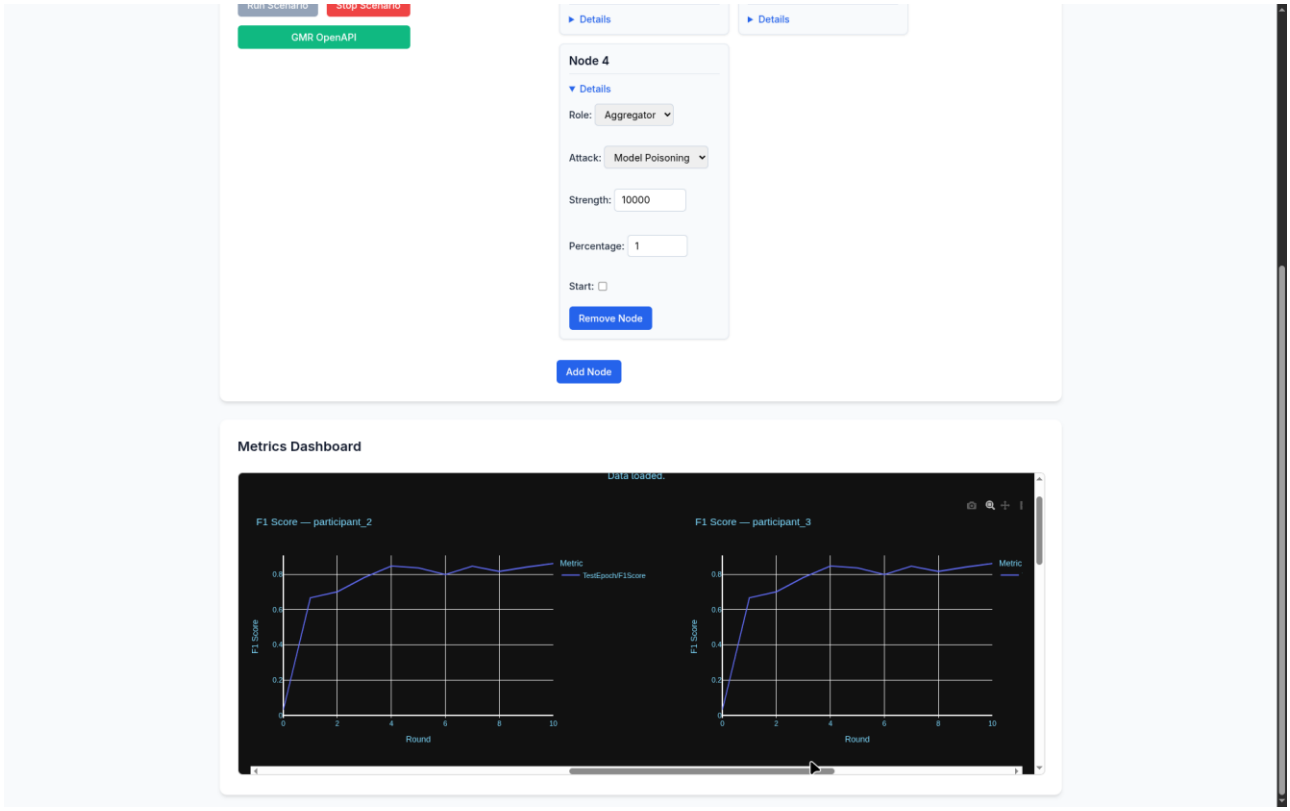

Figure 5.3 DFL Framework Frontend Active Training Progress Metrics

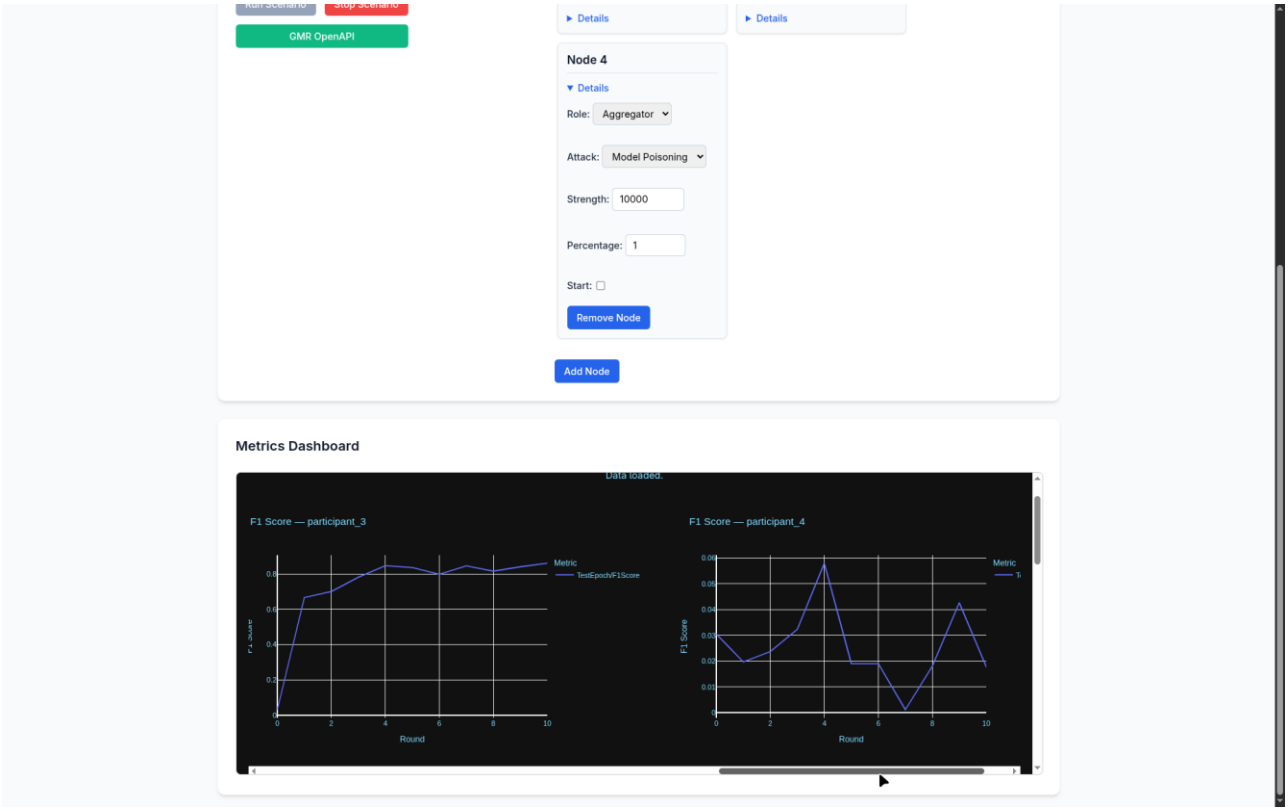

Figure 5.4 DFL Framework Frontend Active Training Progress Metrics

Krum's mitigation preserves the training stability of the federated model under active attack. The final aggregated model achieves an average accuracy improvement of at least 5% over standalone local training, and an accuracy improvement of at least 6.6% compared to the vanilla FL baseline. The update sharing latency averages 1.96 seconds, well within the 30-minute DoA threshold.

Robust Time-Series Model Explainability and GMR Registration

At the final round, the completed aggregated model is evaluated. To generate transparency evidence, the framework automatically triggers the shats_explainer module.

The explainer utilizes the FastShaTS library to compute Shapley feature importance values for the robust model. It draws a background dataset composed of 100 randomly sampled benign traffic instances and evaluates a test window of 100 samples. Using the FeaturesGroupingStrategy, the variables are mapped to the specific network and system log features of the TON-IoT dataset, such as connection state, source/destination ports, packet sizes, and flow durations. Explaining how individual features contribute to classifying traffic as an attack or normal.

The module generates a feature importance visualisation plot that provides a clear overview of the most influential features in the model (e.g., shats_features_plot_round_9_class_ddos.png), showing the feature weights, as depicted in Figure 5.5 and Figure Figure 5.6.

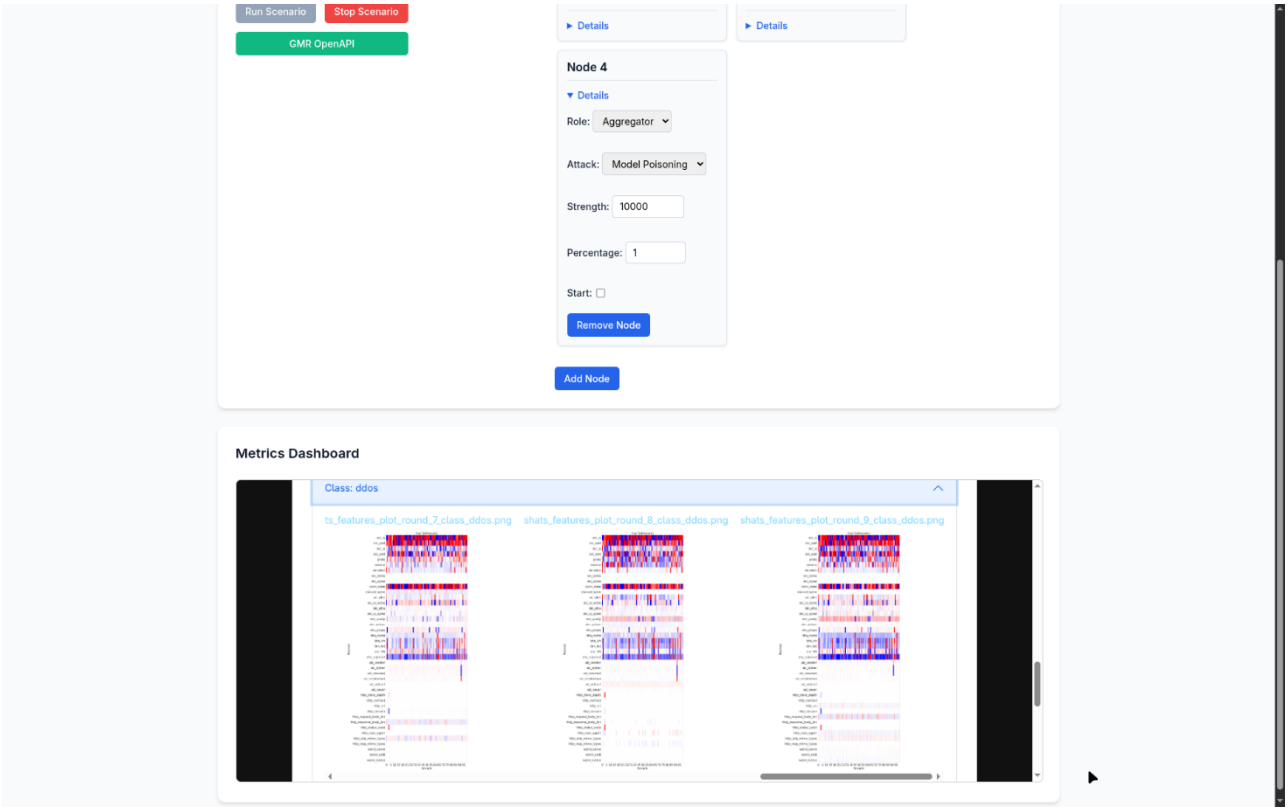

Figure 5.5 SHATs Feature Importance Plot in the DFL Framework

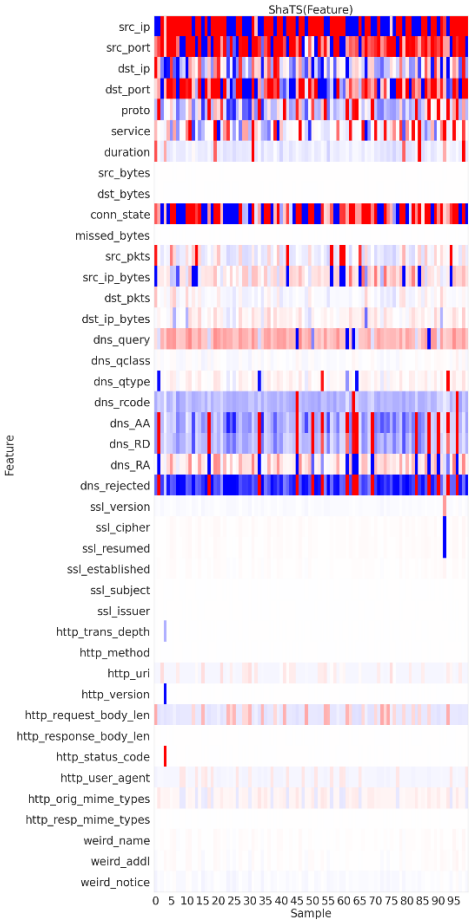

Figure 5.6 DDoS Anomaly Explanation Participant 0 round 9

Simultaneously, the DFL framework uses its RepositoryClient to package the PyTorch model weights, training metrics, system performance logs CPU/GPU/RAM metrics, and the generated ShaTS plots and JSON files. These are uploaded via REST API POST requests directly to the GMR service running on port 8001.

The GMR Web Dashboard is accessed to verify the publication. As depicted in Figure 5.7, Figure Figure 5.8, and Figure 5.9, the GMR interface displays the registered model and scenario in the catalog, showing its version, metadata details, accuracy performance, and download links for both the weights and explainability logs.

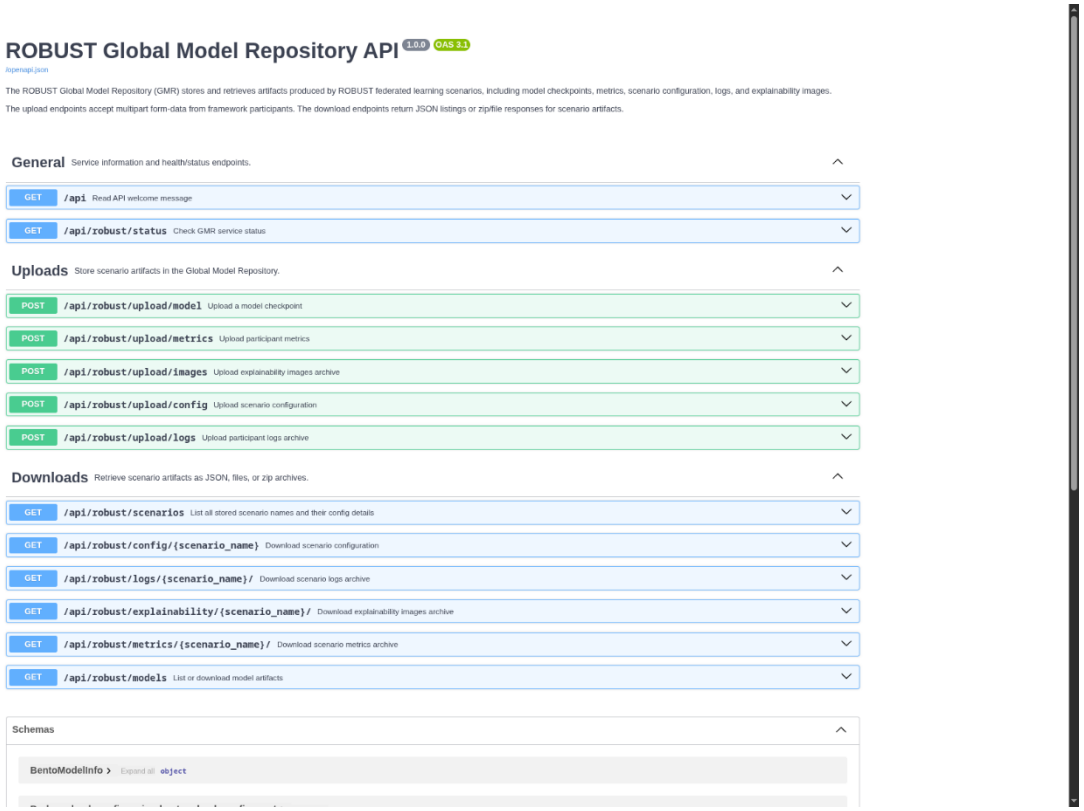

Figure 5.7 GMR API dashboard

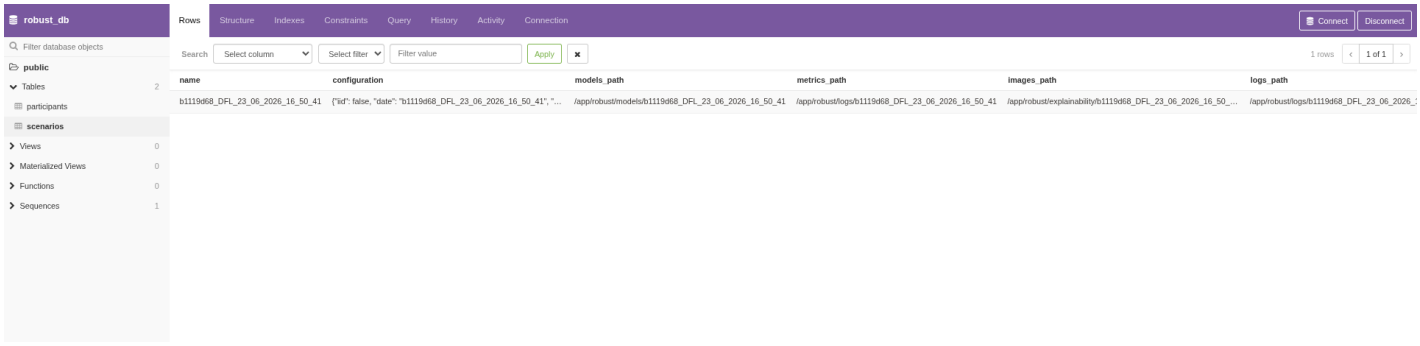

Figure 5.8 GMR Postgress Database - Table scenarios

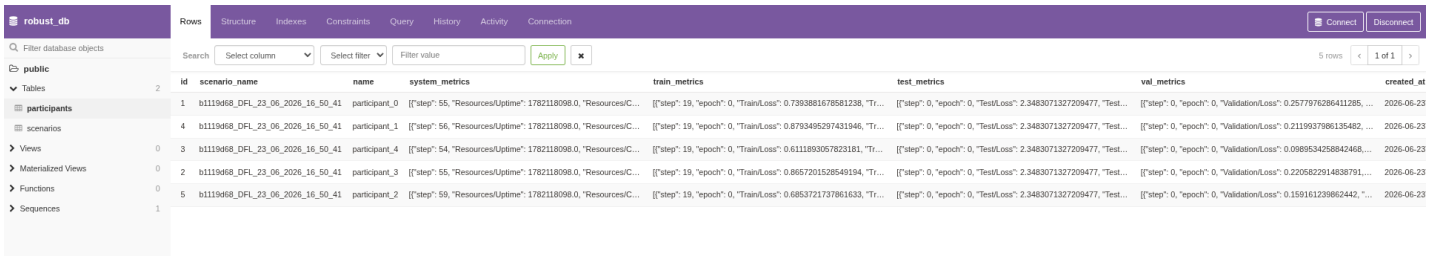

Figure 5.9 GMR Postgress Database - Table participants

Overall, Prototype 1 supports the following validation conclusion. The DFL Framework and GMR provide an operational, modular, and repository-backed trustworthy AI layer for ROBUST-6G. The prototype successfully demonstrates decentralised training, attack-aware evaluation, metric collection, model registration, and reuse of AI assets. KPI evidence from D3.4 confirms achieved status for federated accuracy improvement, model update sharing time, trust-based accuracy improvement, energy-related contributions, and trustworthiness/performance trade-off control. The composite trustworthiness score, full adversarial robustness score, and membership-inference reduction target are supported by the available evidence, which indicates substantial progress and provides clear measurement hooks for the validation runs.

The supported TRL claim for Prototype 1 is TRL 5-6. This is justified by the operational containerised deployment, execution of distributed DFL workflows, integration with a repository-backed model governance layer, native support for adversarial experimentation, and availability of measurable training, robustness, explainability, and system metrics. The main limitation is that some KPI values still require final scenario-level consolidation in the integrated WP6 testbed, especially those depending on the definitive adversarial attack set, final dataset configuration, and end-to-end integration with other ROBUST-6G prototypes.

5.1.2 Prototype 2: Multi-Layer Zero-Touch Defender

5.1.2.1 Validation Setup

The validation setup for Prototype 2 is designed to demonstrate the ROBUST-6G Work Package 4 components for AI-driven, multi-layer zero-touch security. The execution environment integrates the complete stack of the Zero-Touch Security Platform (ZTSP), which is triggered once a Security Service Level Agreement (SSLA) is received, tasking the platform to deploy and configure a cohesive security service.

5.1.2.2 Testbed Configuration

In the first two scenarios of Use Case 2, as depicted in Figure 5.10, the validation is completely executed using the NXW testbed, where the entire Zero-Touch Security Platform (ZTSP) is deployed. The only exception to this localised deployment is the GenAI4SOAR component, which operates externally and is hosted at the THALES premises. Here, as already discussed in D4.4 [R6G26-D44], it is important to note how the ZTSP is a consumer of the GenAI4SOAR Service, which is seen as a black-box capable of providing an IRP given the context extracted and provided by the ZTSP. For the third scenario (UC2.3), as depicted in Figure 5.11, the testbed configuration is expanded with an additional K3S cluster running on six different VMs and the link to the Data Management Platfom achieved though secure VPN access.

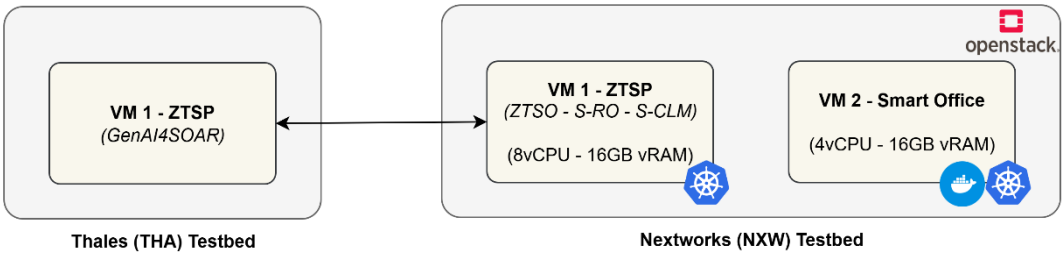

Figure 5.10: UC2 Scenario 1 and 2 - testbed configuration

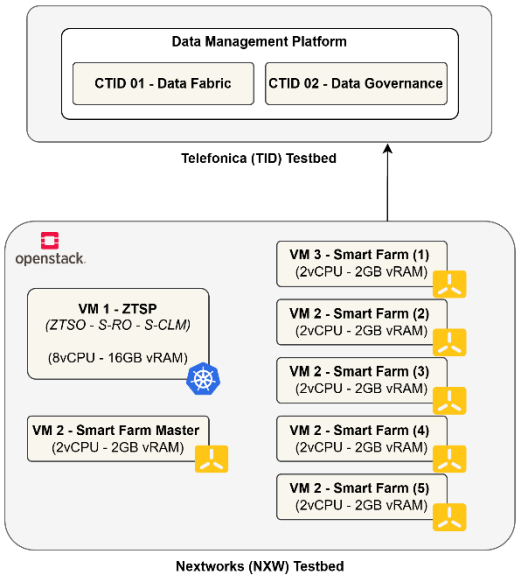

Figure 5.11: UC2 Scenario 3 - testbed configuration

5.1.2.2.1 Datasets

Given that the focus of this use case is on red team activities rather than blue team operations, the validation utilises the ToN-IoT dataset to simulate realistic network attacks. This dataset provides raw .pcap files that are injected directly into the network interfaces. The Programmable Monitoring Platform (PMP) processes these injected traces differently depending on the scenario:

- In Scenario 1, the PMP utilises the traces to create alerts using the internal TShark and SNORT modules.
- In Scenario 2, the PMP processes the network traces collected with TShark into network flows using the internal CIC Flowmeter tool. These networks are subsequently used by AI/ML models for threat detection.

5.1.2.2.2 Integration Details

The internal integration details of the ZTSP Platform, including the interactions between the Zero-Touch Security Orchestrator (ZTSO), the PMP, the Secure Resource Orchestrator (S-RO), and other core components, have been extensively reported in D4.4 [R6G26-D44]. As already explained in the document, the remaining external integrations with broader ROBUST-6G Platform components (such as the Global Model Repository (GMR), PHY layer, and NetSecaaS Gateway) are intentionally deferred and are fully reported in Prototype 5.

5.1.2.2.3 Inputs and Outputs

Focusing on Scenario 1, which serves as the primary demonstrator for UC2 and Prototype 2, the functional validation relies on the following inputs:

1. An SSLA (Security Service Level Agreement) defining the specific Security Service requested (e.g., enforcing a password policy with a minimum entropy).
2. The Security Functions (SFs) and S-CL templates and descriptors, which are already loaded and onboarded onto the ZTSP (as heavily described in D4.4 [R6G26-D44] workflows).
3. The ToN-IoT dataset used to simulate the network attacks.

The outputs of the validation are divided into intermediary and final outputs:

- Intermediary Outputs: the automated generation of a CACAO incident response playbook (via the GenAI4SOAR crew) and the successfully composition and deployment of the Security Service.
- Final Output for Verification: the actual zero-touch remediation that is autonomously actuated on the IoT Platform when the attack occurs, neutralizing the threat without human intervention

5.1.2.3 Validation Outcomes

UC2.1 Validation Outcomes

This section details the functional validation of Use Case 2 Scenario 1, where an attacker attempts to compromise an HVAC system within a smart office to cause economic harm and discomfort. The validation begins with the initialisation of the Zero-Touch Security Orchestrator (ZTSO), whose workflows are described in D4.4 [R6G26-D44]. As shown in Figure 5.12, at setup time, the administrator interacts with the Ontology Manager to populate the semantic model (TBox and ABox) of the ZTSO. The visual evidence of the Knowledge Graph confirms that the ZTSO Ontology is correctly configured to describe the TBOpenC2Actuator, which is mapped through semantic reasoning to the suitable activities and environments.

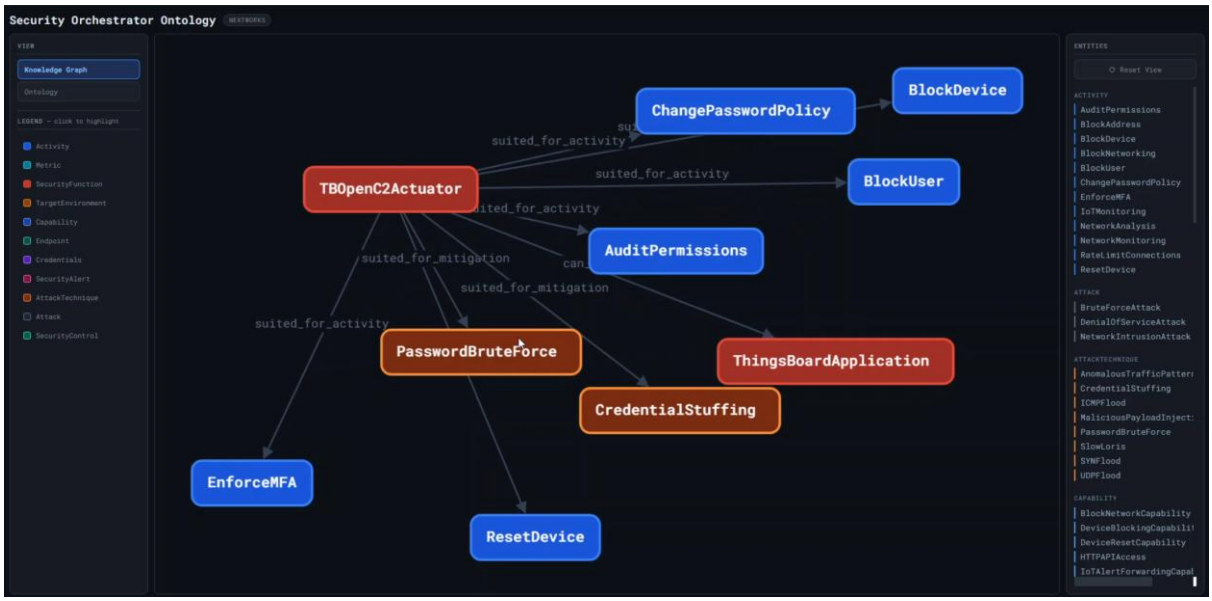

Figure 5.12: ZTSO - TBOpenC2 Actuator Knowledge Graph

As depicted in

Figure 5.13, all the available Security Functions are correctly uploaded on the ZTSO Catalogue that shows all the available Security Applications and Security Actuators. Finally, Figure 5.14 depicts the target infrastructure of this scenario, the ROBUST-6G Smart Office, with its registered environments (ThingBoard, Kubernetes, and Docker) and the available Security Applications (Programmable Monitoring Platform deployed on Docker).

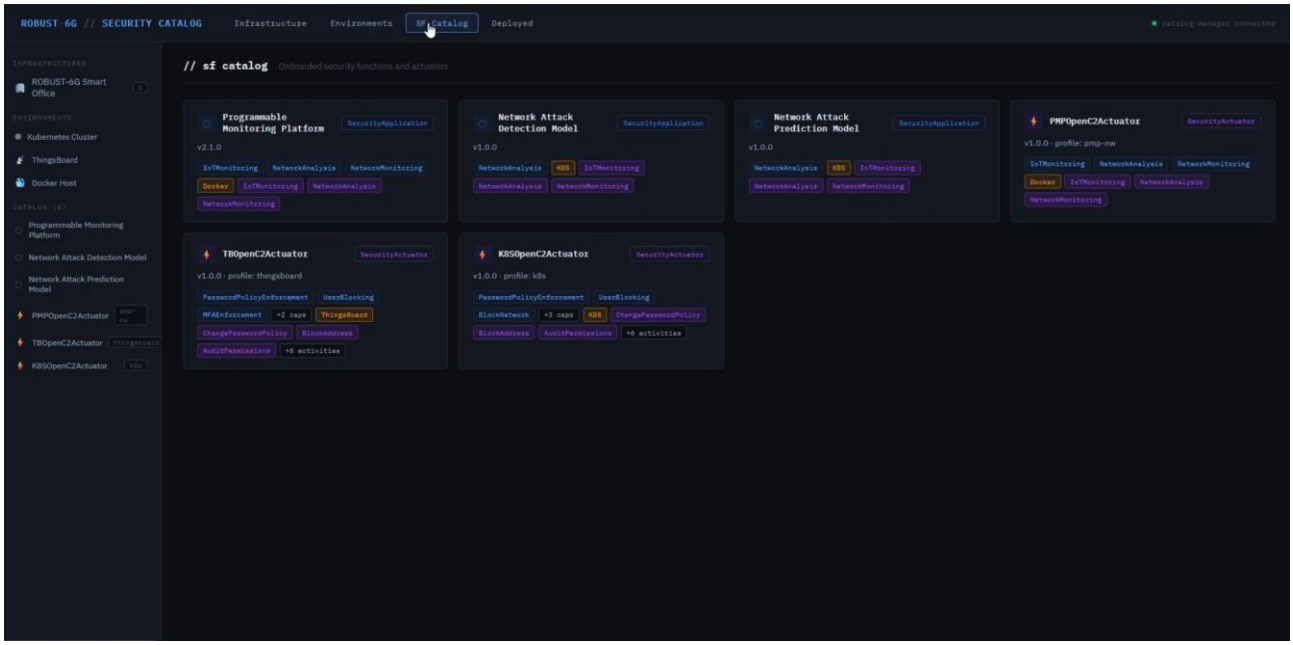

Figure 5.13: ZTSO - Catalogue of Security Functions

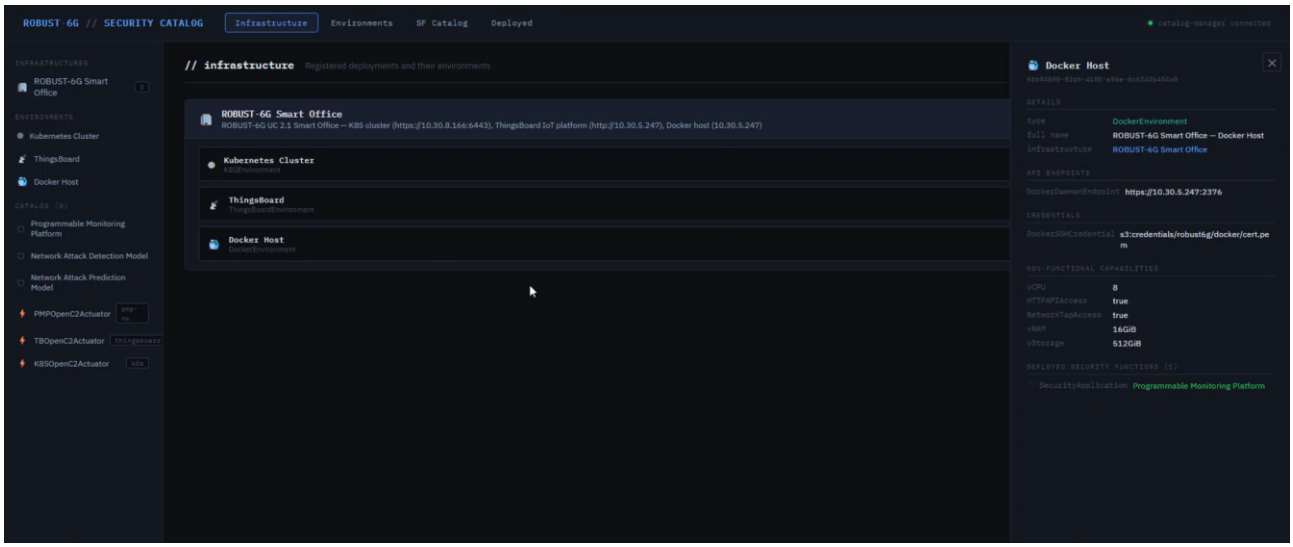

Figure 5.14: ZTSO - Infrastructure Environments

During the ZTSP Setup, the Platform Manager of the S-RO is loaded with the needed platforms that can be used as targets for the orchestration of Security Functions (SF) and Security Closed Loops Functions (S-CLF). As depicted in Figure 5.15, in the scope of UC2.1 and UC2.2, a single platform is loaded as possible orchestration target: the Smart Office K8S Cluster that was previously loaded in the ZTSO Catalogue. Moreover, all the artefacts related to the SFs and S-CLF are onboarded on the Service Template Catalogue of the S-RO, as shown in Figure 5.16.

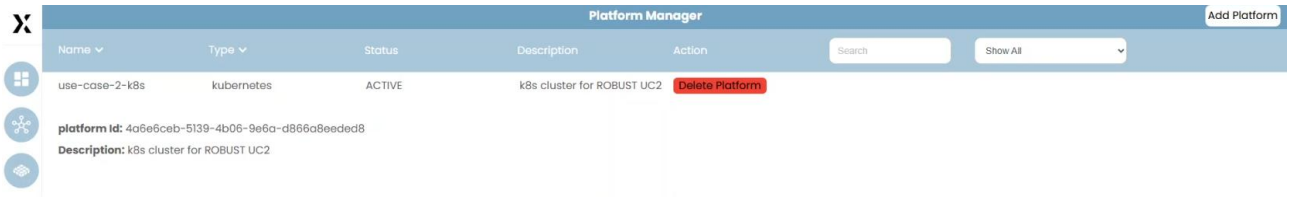

Figure 5.15: S-RO - Onboarded Environments

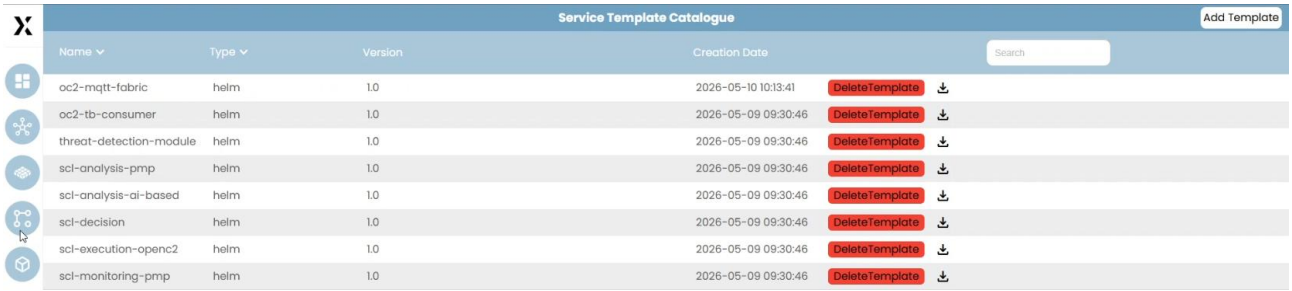

Figure 5.16: S-RO - Onboarded SFs and S-CL Functions Artefacts

As final step of the ZTSP configuration, as shown in Figure 5.17, the S-CL Mgmt Platform is onboarded with the needed descriptors for Security Closed Loop Functions (S-CLF). Then, the S-CLF descriptors are used to compose S-CL descriptors. Figure 5.18 depicts the S-CL Descriptor of the Rule Based S-CL that is used in this particular scenario.

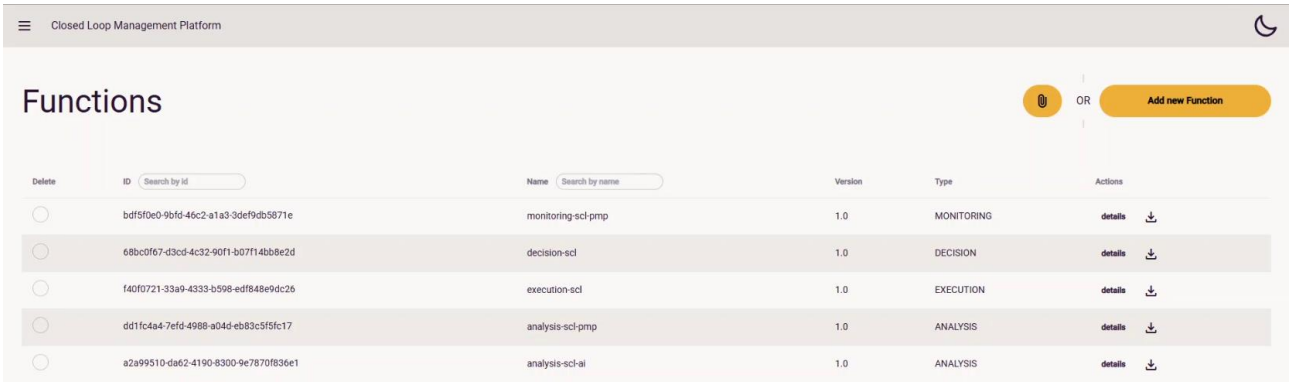

Figure 5.17: S-CL Mgmt - Onboarded S-CLF Descriptors

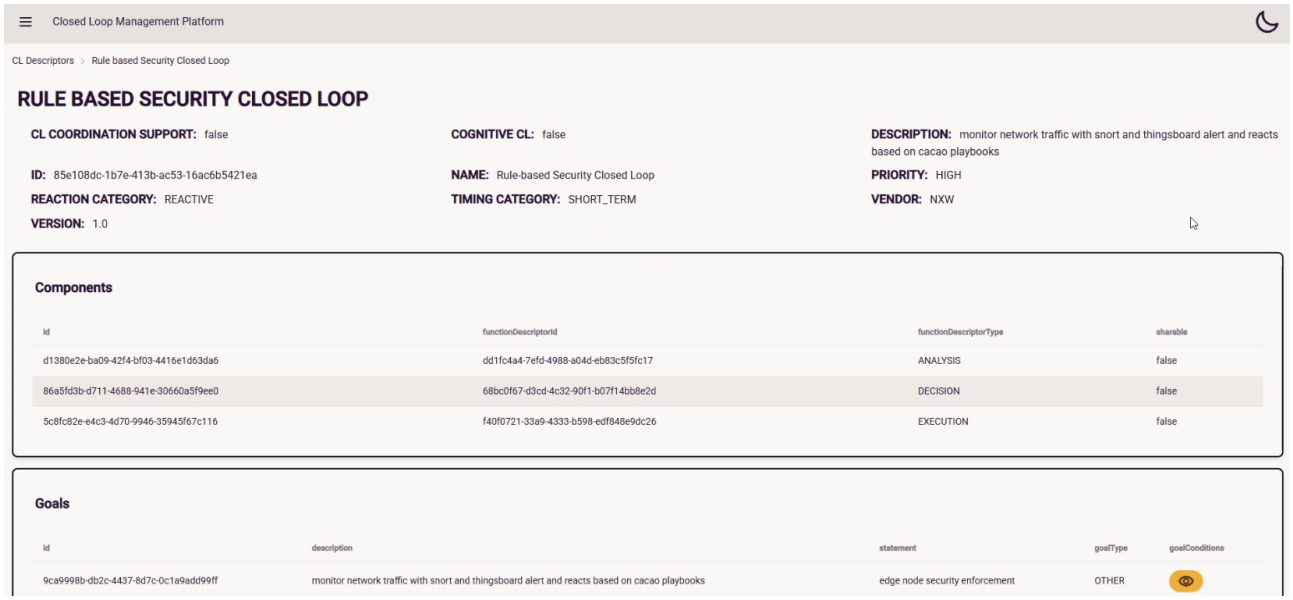

Figure 5.18: S-CL Mgmt - Onboarded Rule-based S-CL Descriptor

Once the ZTSP is correctly set up, as part of the scenario the ThingBoard IoT platform is configured to host the HVAC system of the smart office (three simulated devices: HVAC Actuator, Temperature Sensor, and Presence Sensor) and provides a dashboard, reported in Figure 5.19, to the final user for overriding the HVAC configuration to Always ON, Always OFF, or Normal (follows the opening hours of the office and keeps a stable and comfortable temperature). The Platform Security settings are left untouched from the installation, having a Password Policy for new users, depicted in Figure 5.20, that is enforcing only a minimum password length of 6 characters.

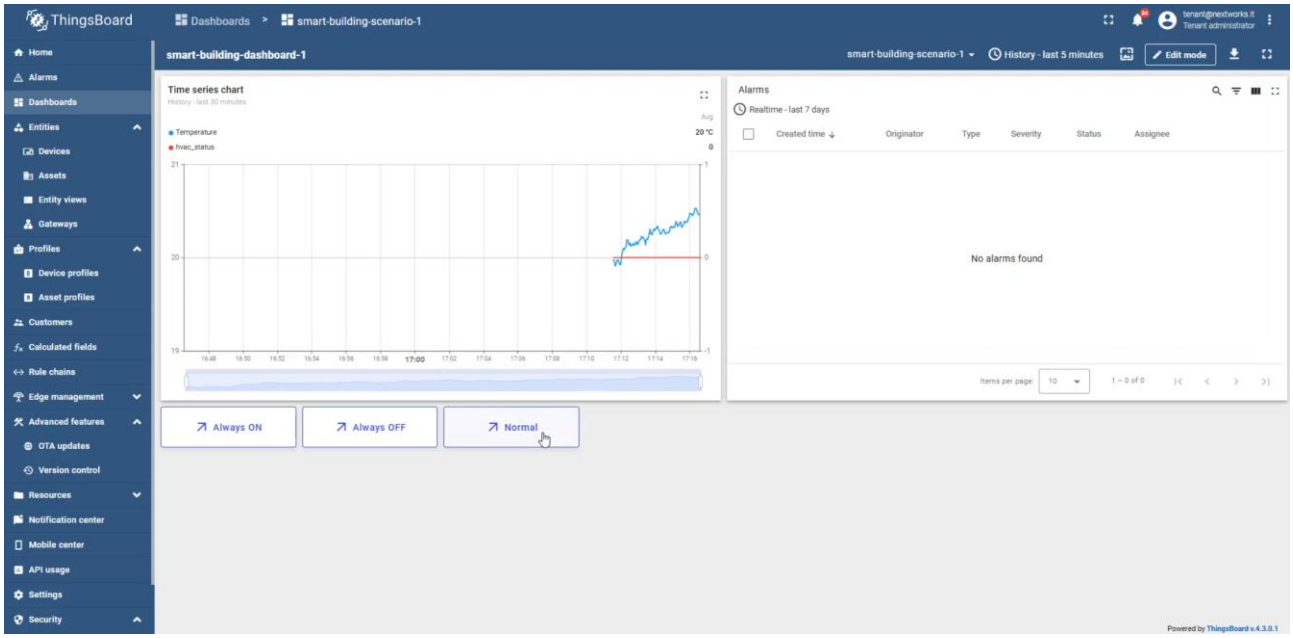

Figure 5.19: ThingBoard User Dashboard

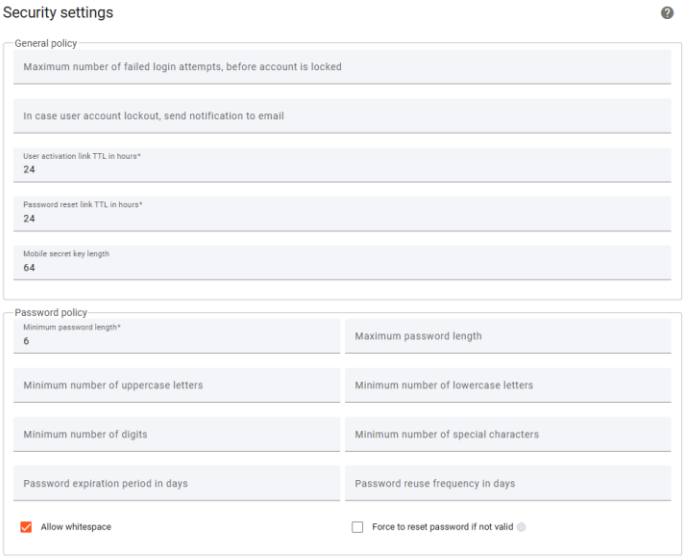

Figure 5.20: ThingBoard default Security Settings

To protect this infrastructure, the infrastructure owner submits an SSLA, reported in [THA26a] to the ZTSP triggering the Security Service Provisioning. This SSLA specifies the required activities for network and IoT monitoring, and requests a specific security automation: the capability to dynamically enforce a hardened password policy guaranteeing at least 100 bits of entropy. After SSLA Ingestion and decomposition, the Security Context Manager (SCM) intercepts the request and validates it using the semantic ontology. By evaluating the requested activities against the available infrastructure, the SCM successfully maps the security requirements to the deployable Security Functions and available Target Environments, generating a scoped infrastructure context which is then provided to the Policy Manager to be sent to the GenAI4SOAR component for SFs selection and IRP generation. Figure 5.21 reports the SCM logs where the semantic check is performed and the scoped context for the infrastructure is retrieved.

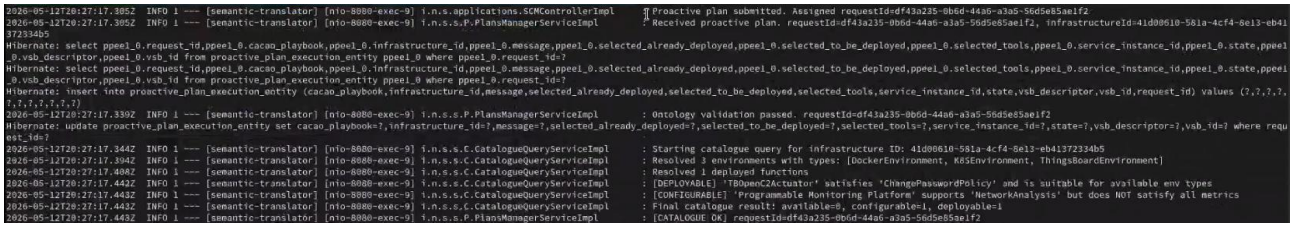

Figure 5.21: ZTSO SCM Plan submission and context retrieval

To define the automated response strategy, the GenAI Gateway forwards this scoped context to the GenAI4SOAR component. The logs of the GenAI Gateway reported in Figure 5.22 depict the GenAI crew of specialized agents collaborating to parse the context and generated autonomously the CACAO v2 and OpenC2 compliant Playbook. The specific agents and the details regarding their role in the crew are reported in D4.4 [R6G26-D44] while the Agents prompts can be found in [THA26b]. The generated playbook, reported in Figure 5.23, consists of four consecutive steps: 1) unassign devices from the user, 2) suspend the user account, 3) enforce the new >100-bit password policy, and 4) send a reactivation email to the user.

```
[22:34:19.056] INFO: started remediation playbook {"crew":"brynn","sessionID":"520ff1db-6c66-41c3-bb04-1cb5744e3314"}
[22:34:19.056] INFO: acquire stream response from generation {}
[22:34:19.265] DEBUG: response {"data":"{\"type\": \"crew_status\", \"status\": \"RUNNING\"}"}
[22:34:19.317] DEBUG: response {"data":"{\"type\": \"task\", \"status\": \"started\", \"task_description\": \"Analyse the alert context and generate a textual an
[22:34:52.911] DEBUG: response {"data":"{\"type\": \"task\", \"status\": \"completed\", \"task_description\": \"Analyse the alert context and generate a textual
[22:34:52.996] DEBUG: response {"data":"{\"type\": \"task\", \"status\": \"started\", \"task_description\": \"Review the description you received and transform i
[22:35:28.357] DEBUG: response {"data":"{\"type\": \"task\", \"status\": \"completed\", \"task_description\": \"Review the description you received and transform
[22:35:28.477] DEBUG: response {"data":"{\"type\": \"task\", \"status\": \"started\", \"task_description\": \"Review the context you got as a detailed mitigation
[22:36:40.527] DEBUG: response {"data":"{\"type\": \"task\", \"status\": \"completed\", \"task_description\": \"Review the context you got as a detailed mitigati
[22:36:40.589] DEBUG: response {"data":"{\"type\": \"task\", \"status\": \"started\", \"task_description\": \"Review the context you got as a high-level step-by-
[22:37:04.874] DEBUG: response {"data":"{\"type\": \"task\", \"status\": \"completed\", \"task_description\": \"Review the context you got as a high-level step-b
[22:37:05.000] DEBUG: response {"data":"{\"type\": \"task\", \"status\": \"started\", \"task_description\": \"Review the context you got as a high-level step-by-
[22:38:26.901] DEBUG: response {"data":"{\"type\": \"task\", \"status\": \"completed\", \"task_description\": \"Review the context you got as a high-level step-b
[22:38:26.902] INFO: CACAO playbook generated {"result":"{\n  \"type\": \"playbook\",\n  \"spec_version\": \"cacao-2.0\",\n  \"id\": \"playbook--d1b7f9a2-3c4e-41
```


Figure 5.22: GenAI Gateway IRP Generation

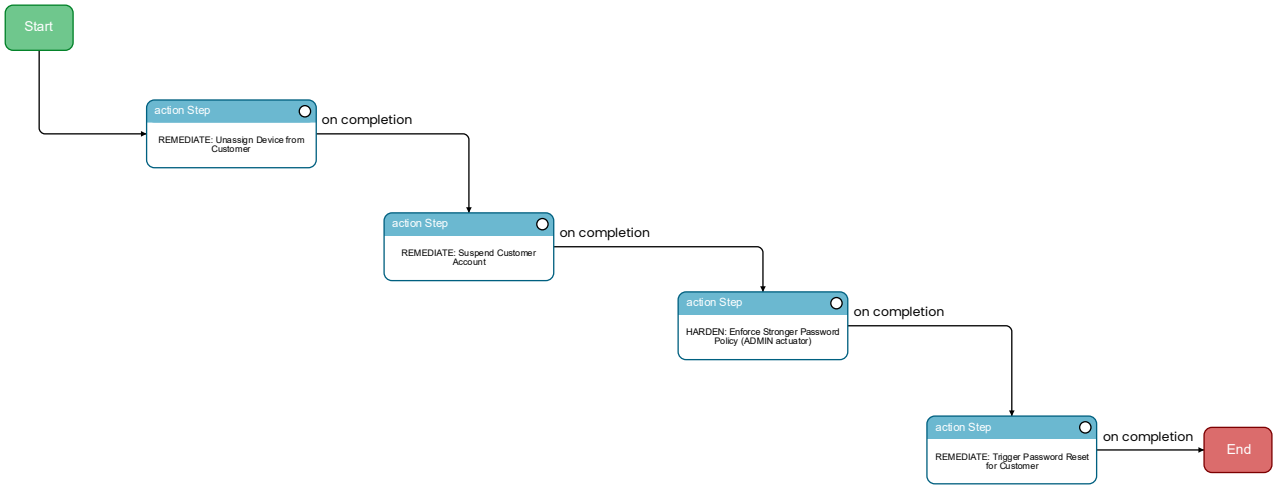

Figure 5.23: Use Case 2 Scenario 1 IRP

At this stage, the Security Functions are selected and the IRP is ready to be used by the S-CL. As a final step, the Security Service composed of the S-CL and the selected SFs (ThingsBoard OpenC2 Actuator) are deployed in the target infrastructure. Figure 5.24 depicts the GUI of the Security Service Lifecycle Management (LCM) component of the SCM, where the service is depicted as INSTANTIATED while Figure 5.25 shows the Smart Office K8S Platform where in the security-functions namespace, the ThingBoard OpenC2 actuator (composed of two actuators: one using tenant and another using admin credentials together with the MQTTfabric needed by the actuators to enable OpenC2) is deployed, and in the closed-loop-functions the three stages of the S-CL (monitoring, analysis, and decision) are deployed.

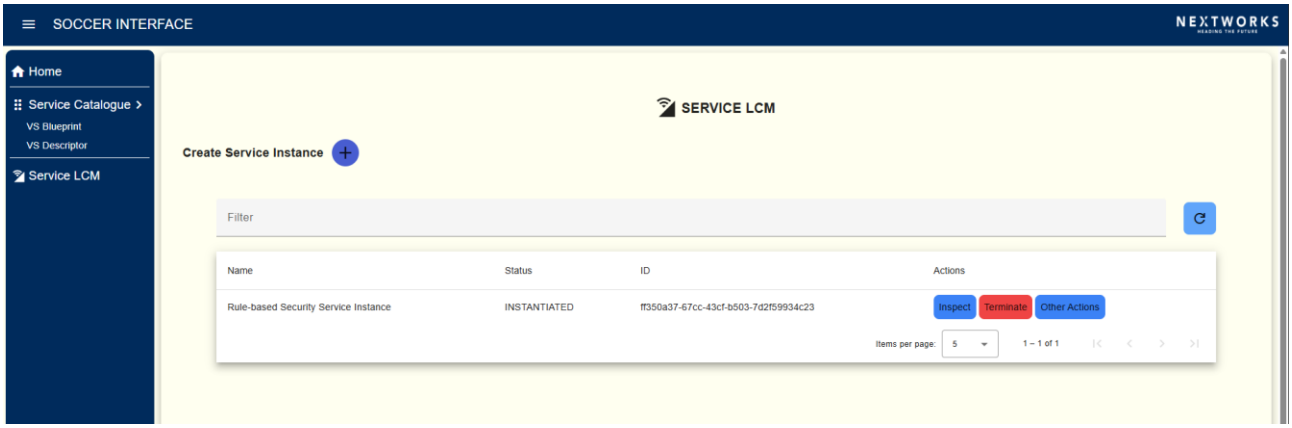

Figure 5.24: ZTSO SCM - Instantiated Security Service

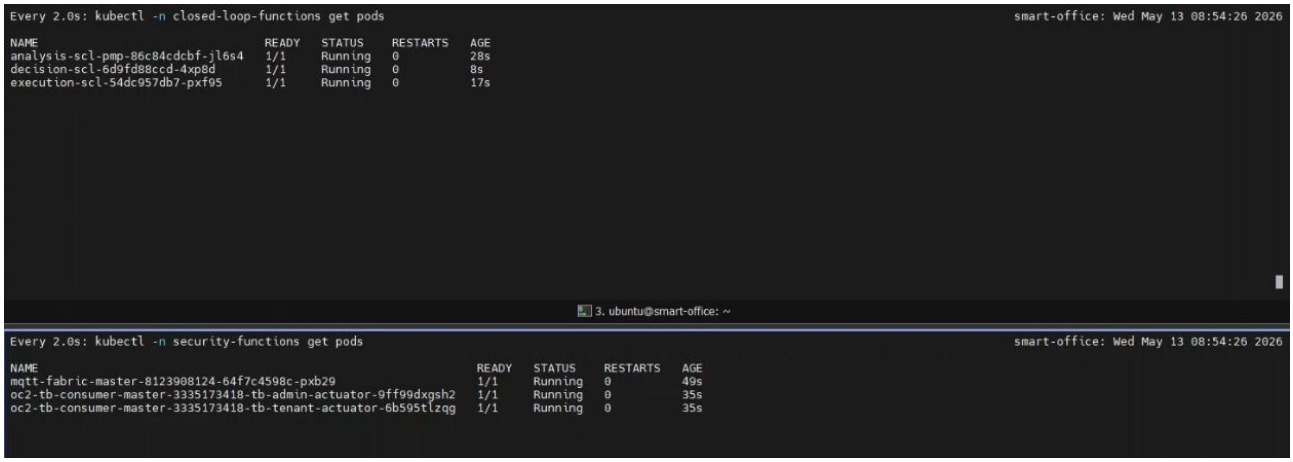

Figure 5.25: Instantiated Security Service on the target Infrastructure

As already described in D4.4 [R6G26-D44], the first step of the S-CL Analysis function, depicted in Figure 5.26, is the configuration of the PMP through its configuration manager API to deploy three tools: TShark for network collection, SNORT3 for network analysis, and ThingBoard monitor to monitor network alerts on ThingBoards devices. Once the tools are successfully deployed, the Analysis function starts to listen on the Kafka topics provided by the PMP for Network and IoT events.

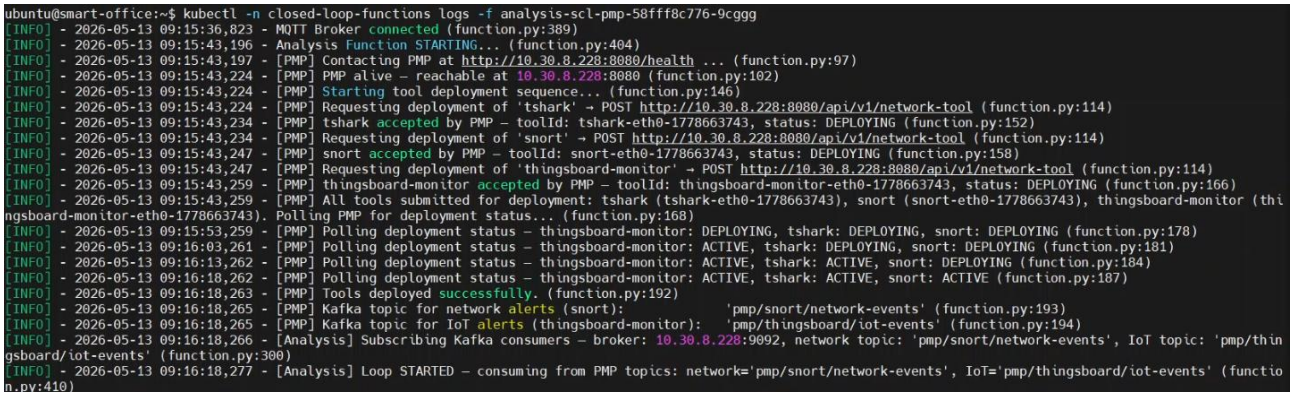

Figure 5.26: UC2 S1 - PMP Configuration

Following the deployment of the Security Service and the configuration of the PMP, the system enters the continuous monitoring phase. When the PMP detects the simulated brute-force attack (with .pcaps injected on the interfaces where the PMP TShark and IoT alert generated by manually setting the HVAC ON during closure, as depicted in Figure 5.27) attempting to manipulate the HVAC device, it triggers the Security Closed Loop (S-CL).

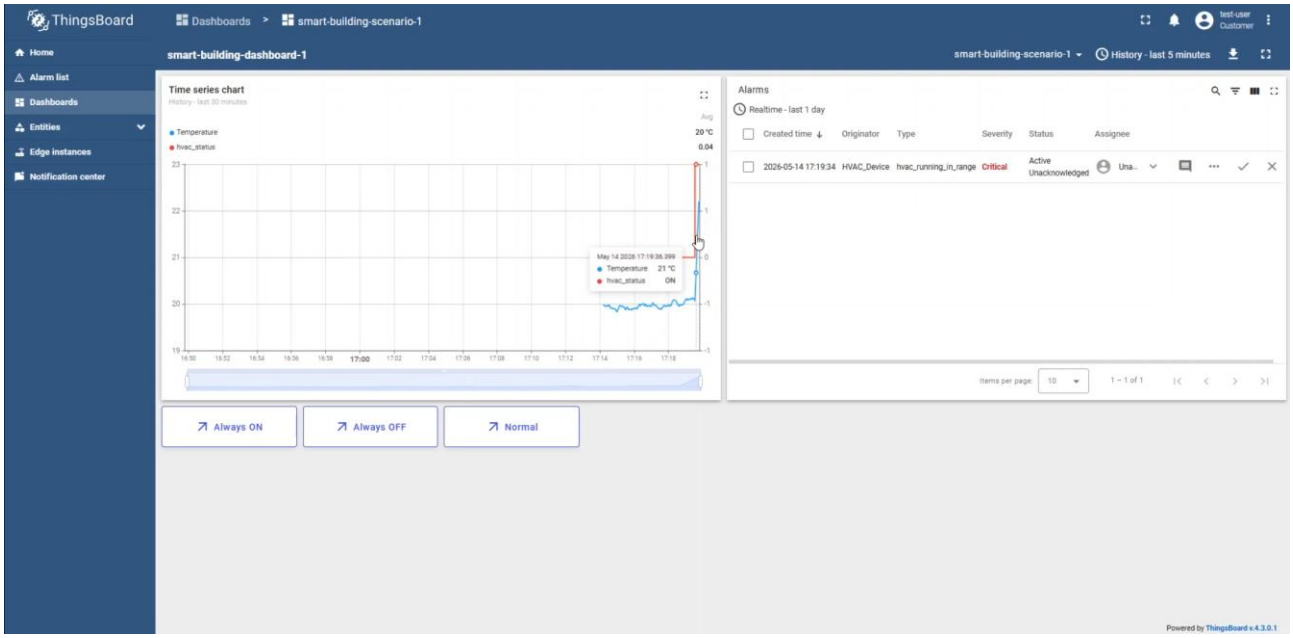

Figure 5.27: Scenario 1 - Attack from the Dashboard

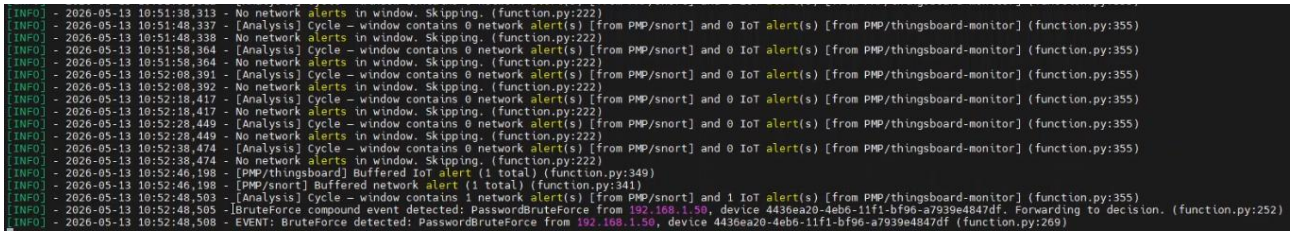

Figure 5.28: UC2 Scenario 1 - S-CL Analysis execution

The S-CL Decision function, as shown in Figure 5.29, retrieves the previously generated CACAO playbook, and the Execution function translates it, pushing the commands to the ThingsBoard OpenC2 actuator. Figure 5.30 reports part of the execution, namely the steps 1 and 2.

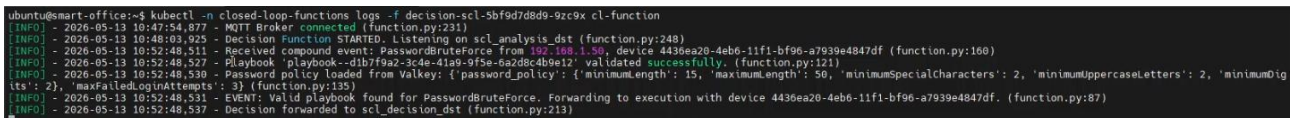

Figure 5.29: UC2 Scenario 1 - S-CL Decision execution

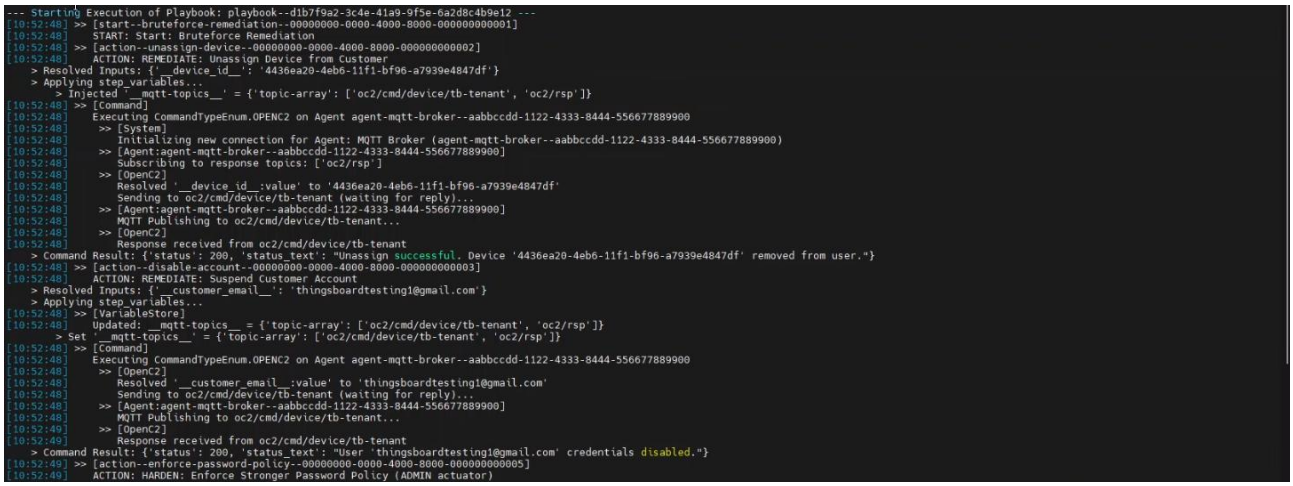

Figure 5.30: UC2 Scenario 1 - S-CL Execution execution

The final validation evidence confirms the successful, zero-touch execution of the four distinct mitigation steps defined in the playbook:

1. The compromised HVAC device is correctly unassigned from the customer, as can be seen from Figure 5.31.
2. The affected user account is disabled to halt the intrusion.
3. The new, robust password policy (>100 bits of entropy) is successfully enforced on the system as can be seen from Figure 5.32.
4. As depicted by Figure 5.33, an account activation link is sent to the user via email, forcing them to comply with the new password policy upon reactivation (Figure 5.34).

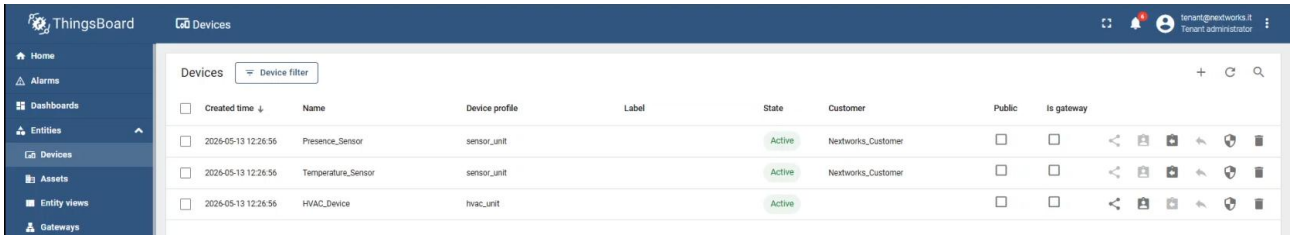

Figure 5.31: UC2 Scenario 1 - Device unassigned from the user

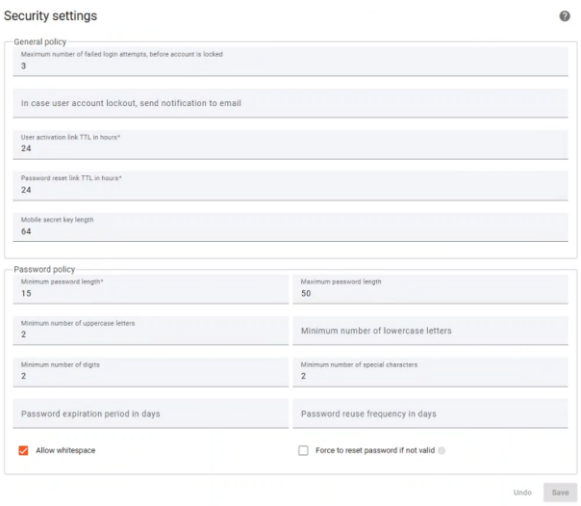

Figure 5.32: UC2 Scenario 1 - Improved Password Policy

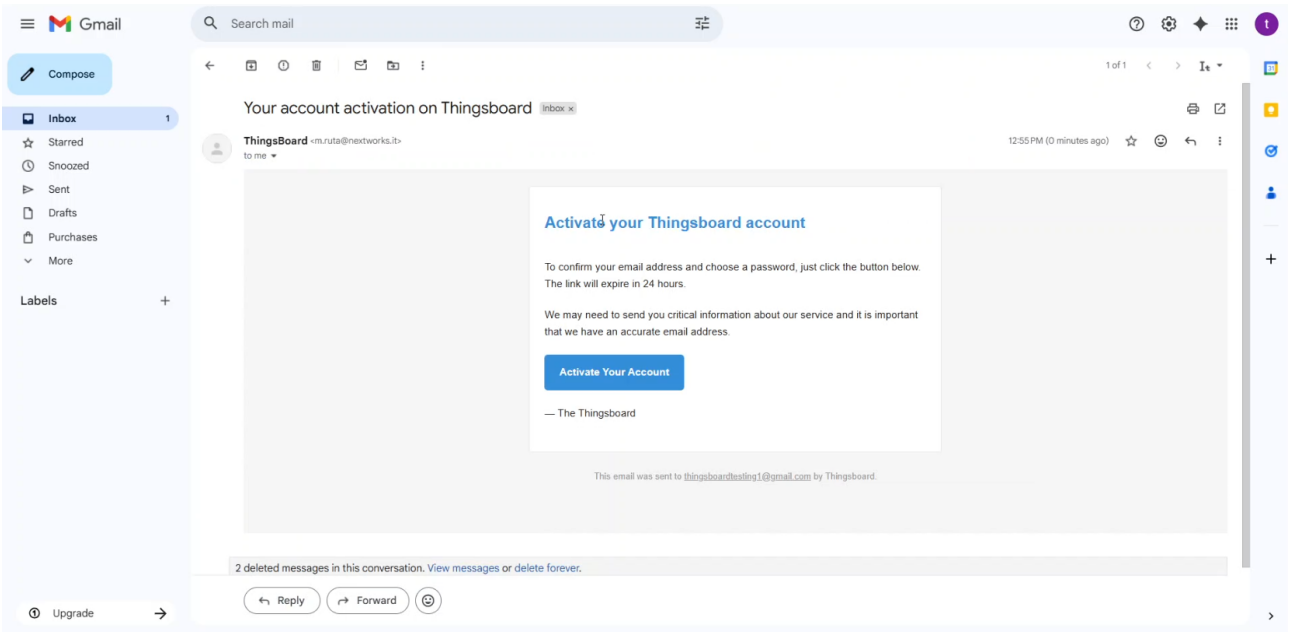

Figure 5.33: UC2 Scenario 1 - User Account re-activation e-mail

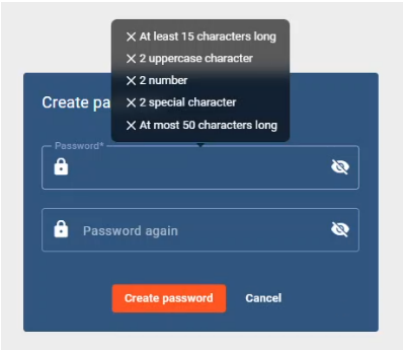

Figure 5.34: UC2 Scenario 1 - New password policy enforcement

UC2.2 Validation Outcomes

This section details the functional validation of Use Case 2 Scenario 2, where an attacker fraudulently exploits the resources of a compromised IoT device to perform cryptomining (cryptojacking). Differently from Scenario 1, which is remediated by a single reactive loop, this scenario validates a two-tier, escalating response: a lightweight investigative Security Closed Loop continuously watches the IoT platform and, on a suspicious symptom, triggers the runtime deployment of a heavier resolutive loop that performs deep, AI-based network-flow analysis and enacts the actual remediation. The escalation is realised as an in-place security service update driven by the investigative loop itself, which required a targeted extension of the Security Context Manager (SCM) lifecycle described later in this section. The target infrastructure is the ROBUST-6G Smart Office IoT deployment managed through ThingsBoard. As shown in Figure 5.35, the ThingsBoard dashboard exposes the monitored devices and their telemetry, while Figure 5.36 shows the Over-The-Air firmware package registered on ThingsBoard that the resolutive remediation will later push to the compromised device. The two security services that implement the escalation are onboarded on the ZTSP as two distinct Security Closed Loops: Figure 5.37 depicts the Investigative S-CL descriptor, composed of two stages (analysis and decision), and Figure 5.38 depicts the Resolutive S-CL descriptor, composed of three stages (analysis, decision and execution). As the operational starting point of the scenario, only the lightweight investigative service is instantiated, as shown in Figure 5.39; the resolutive service instance is created but left dormant, ready to be activated on demand.

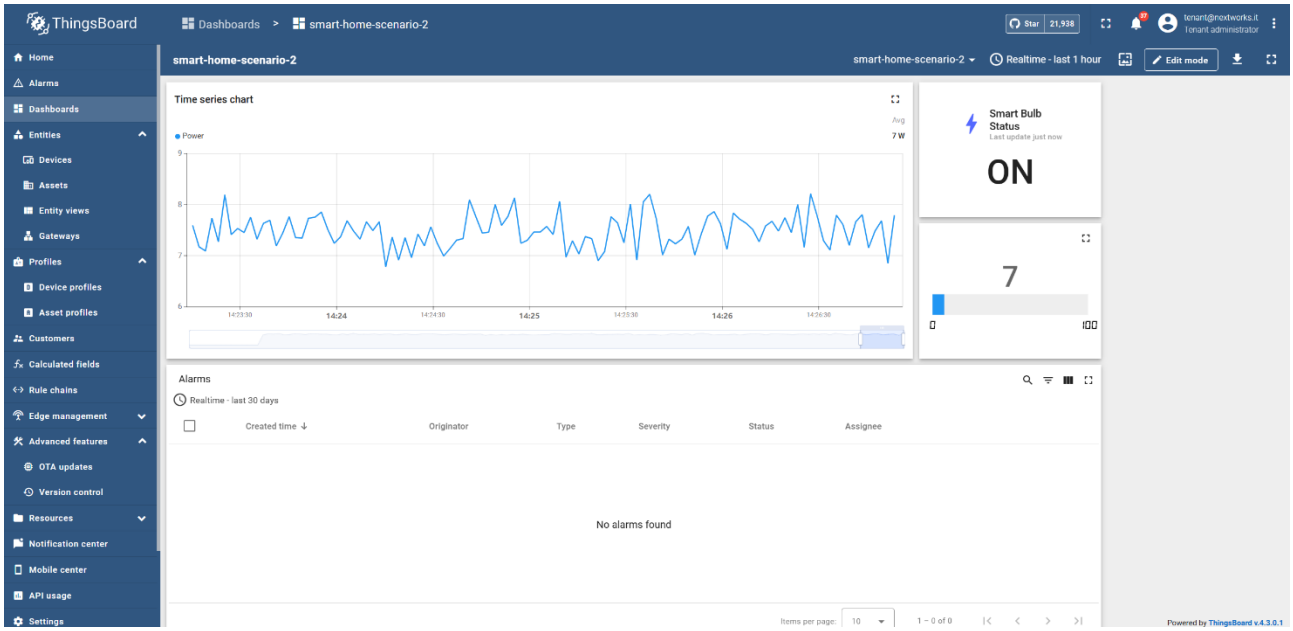

Figure 5.35: UC2 Scenario 2 - ThingBoard Dashboard

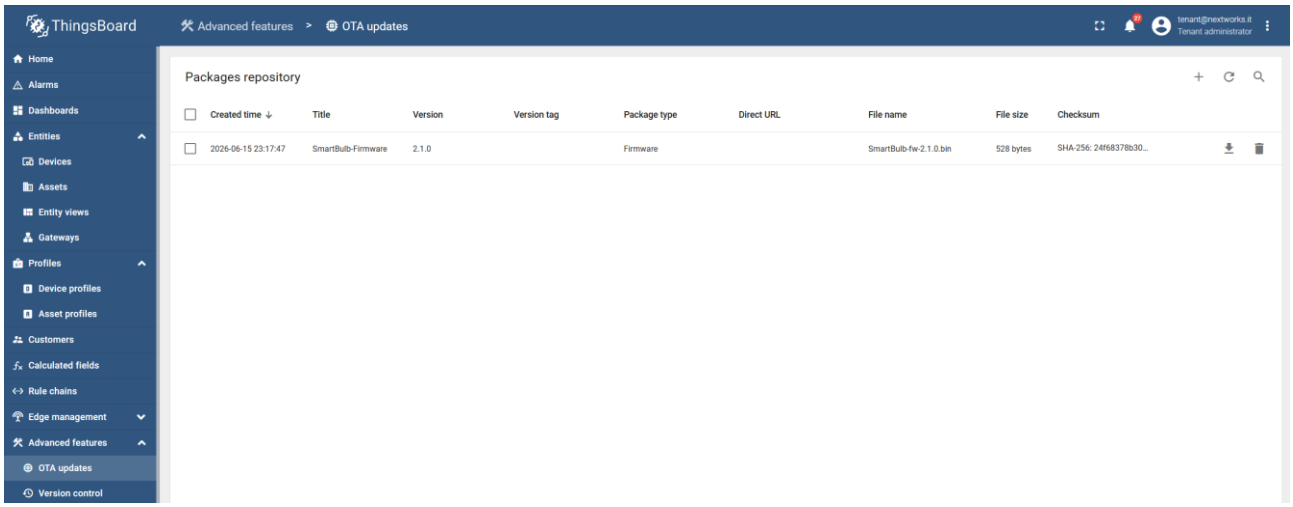

Figure 5.36: OTA Package on ThingBoard

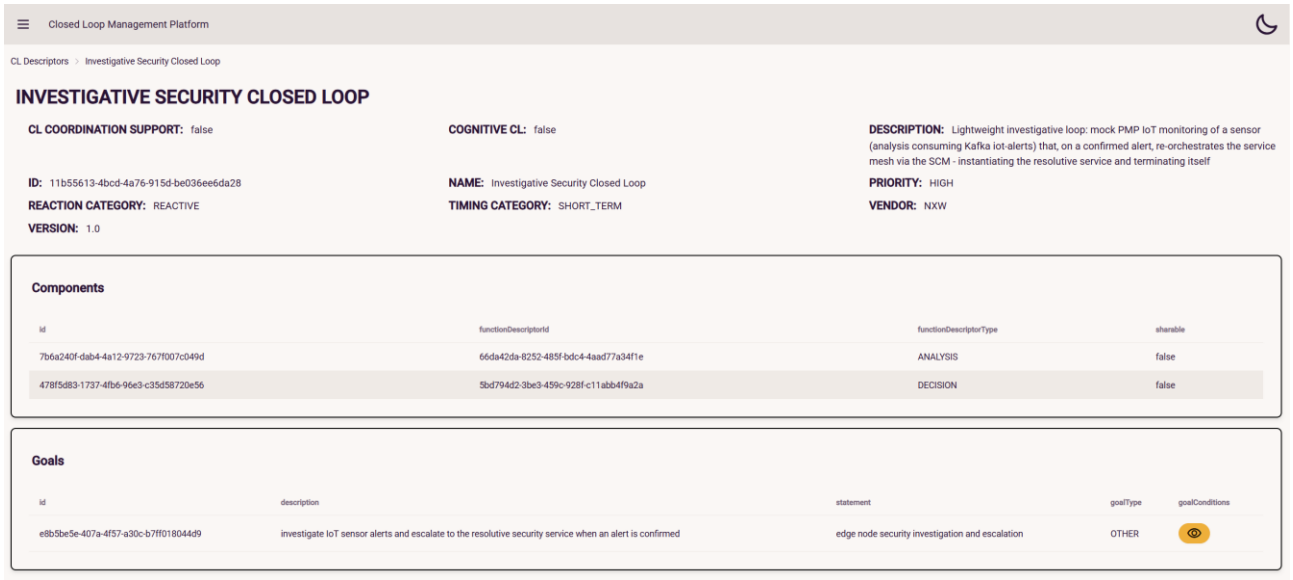

Figure 5.37: Investigative S-CL Descriptor

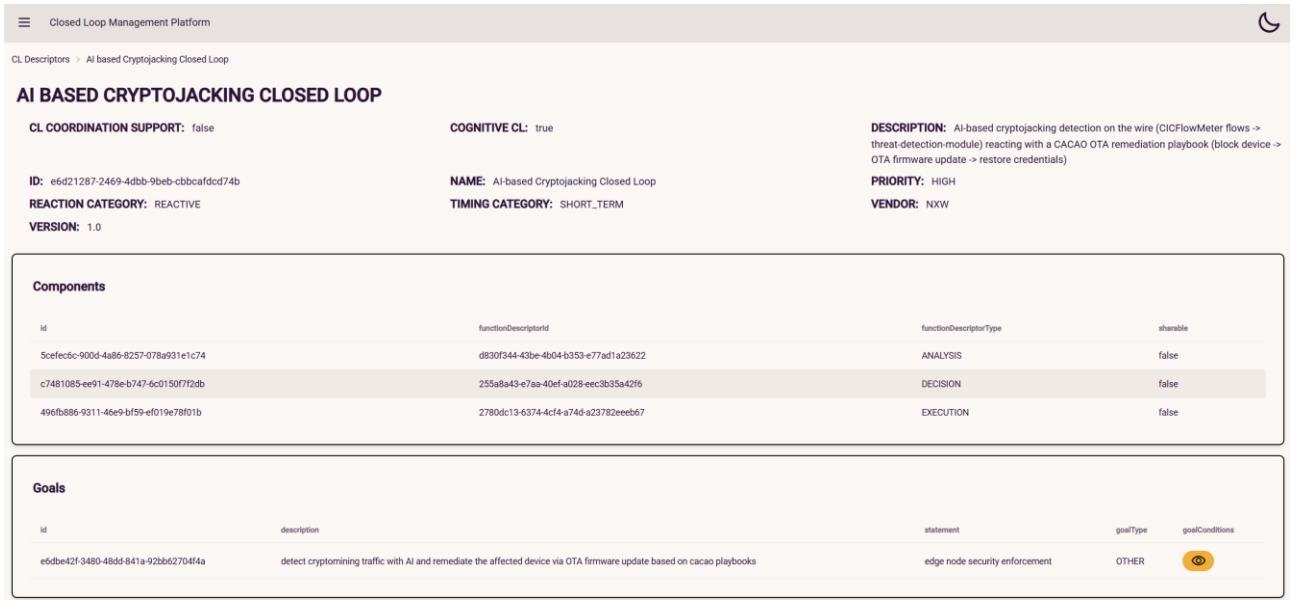

Figure 5.38: Resolutive S-CL Descriptor

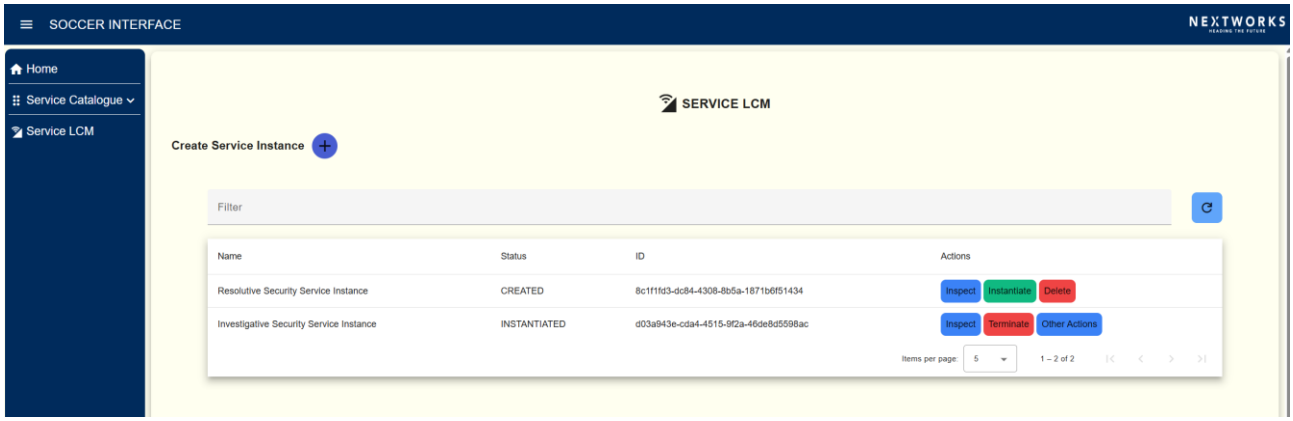

Figure 5.39: Investigative Service Instantiated

During the synchronous monitoring phase the investigative loop watches the IoT platform for the coarse symptoms of resource abuse. As depicted in Figure 5.40, the attack manifests on the ThingsBoard dashboard as an anomalous power-consumption spike on the affected device, the macroscopic signature of the hidden mining workload. Figure 5.41 shows the investigative analysis stage detecting this IoT alert: faithfully to the Scenario-1 monitoring narrative, the stage logs the interaction with the Programmable Monitoring Platform (PMP) to provision IoT monitoring for the device, ingests the alert, and forwards it to the decision stage. Figure 5.42 then shows the investigative decision stage reacting to the alert: rather than attempting a remediation it cannot perform, it issues a security-service update request to the SCM, asking the platform to replace the running investigative service with the resolutive one. The decision log reports the request together with the two service instances involved: the identifier of the service to terminate (the investigative one) and of the service to instantiate (the resolutive one), each resolved from its Vertical Service Blueprint (VSB).

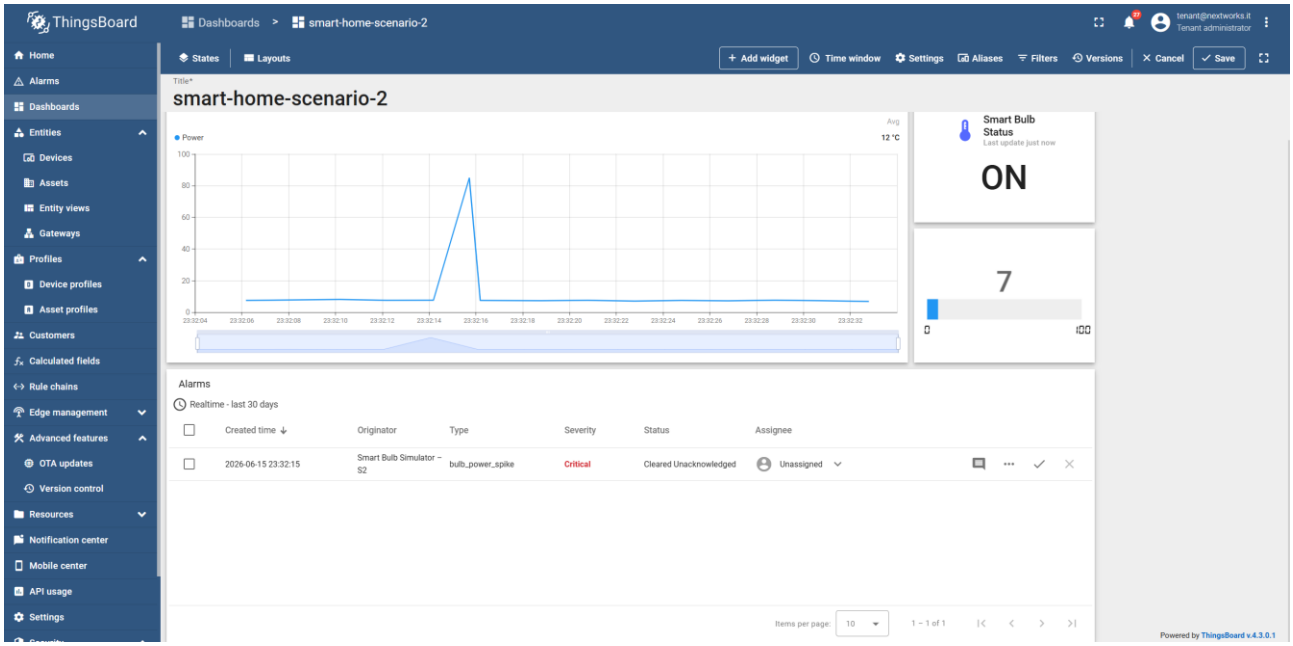

Figure 5.40: IoT Alert from ThingBoard Dashboard - power spike

Figure 5.41: Investigative S-CL Analysis - IoT Alert detected

Figure 5.42: Investigative S-CL Decision - Security Service Update Request

This runtime escalation is not expressible under the SCM state machine defined in D4.4 (Figure 2-3 of D4.4) [R6G26-D44], in which the security service lifecycle ends in a terminal RUNNING state: once a plan is deployed and operational, the only defined exit is its termination. Supporting an investigative to resolutive hand-over on top of that model would have required tearing down the entire plan, discarding its validated ontology context, the retrieved infrastructure context, and the attached CACAO linkage and resubmitting a new one, which is incompatible with a zero-touch, sub-second escalation. To enable it, the SCM OpenAPI and its execution state machine were extended with an in-place service-update capability. Concretely, the SCM now exposes an updateLinkedService operation and the state machine introduces a new UPDATING state and a SERVICE_UPDATE_REQUESTED event, so that RUNNING is no longer a terminal state. The updated lifecycle adds the following transitions on top of the D4.4 flow:

- RUNNING → UPDATING on SERVICE_UPDATE_REQUESTED: a service-update request is accepted on a running plan;
- UPDATING → RUNNING on SERVICE_RUNNING: the swap completed successfully and the plan is operational again;
- UPDATING → FAILED on EXECUTION_FAILURE: the swap failed.

While in UPDATING, the SCM terminates the current (investigative) service instance through the SCM LCM and deploys the new (resolutive) VSB in its place, returning the same plan to RUNNING. The plan, with its validated context and remediation linkage, is therefore preserved across the escalation, and the investigative loop is able to drive the transition autonomously. This extension is the enabling mechanism behind the remainder of the scenario. Acting on the update request, the platform brings up the resolutive service. Figure 5.43 shows the resolutive security service started, and Figure 5.44 shows its footprint materialised on the target infrastructure, with the resolutive S-CL stages deployed alongside the Security Functions they rely on (the AI threat-detection module and the ThingsBoard OpenC2 actuator with its MQTT fabric). The resolutive loop then performs the deep analysis the investigative loop could not.

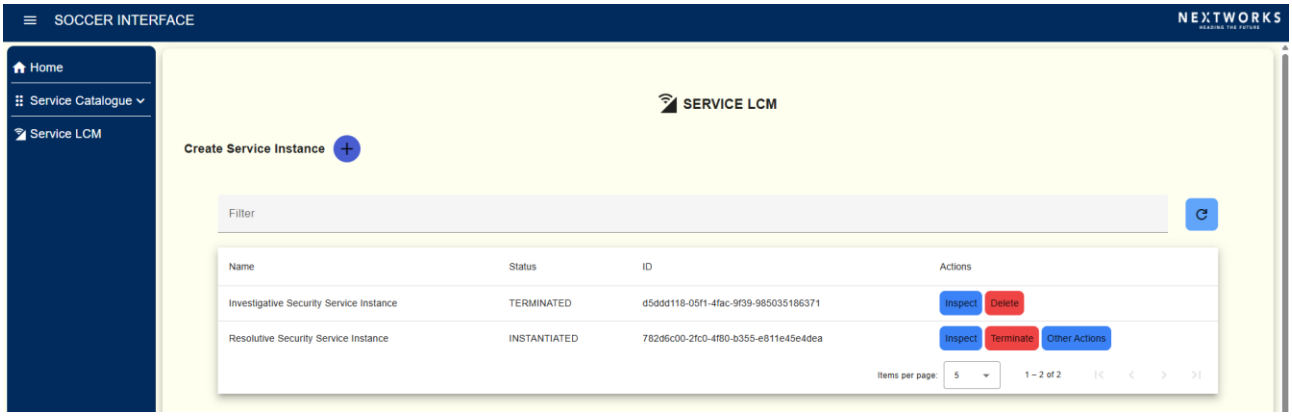

Figure 5.43: Resolutive Security Service Started

```
NAME                                                    READY   STATUS    RESTARTS   AGE
mqtt-fabric-master-5968608572-69c8c7fd7f-69j69          1/1     Running   0          102s
oc2-tb-consumer-master-2776838160-tb-admin-actuator-9dbd66n95sm   1/1   Running   0   86s
oc2-tb-consumer-master-2776838160-tb-tenant-actuator-d7cbbxf4rb   1/1   Running   0   86s
resolutive-analysis-scl-5d8c98bdd8-x6hj9                1/1     Running   0          73s
resolutive-decision-scl-5dbfc6c4f6-nw2sl                1/1     Running   0          52s
resolutive-execution-scl-75df59747c-h9zhk               1/1     Running   0          64s
threat-detection-module-7162837020-5948856cd4-pcbxk     1/1     Running   0          84s
```

Figure 5.44: Resoultive Service SFs and S-CL Stages Deployed

As shown in Figure 5.45, the resolutive analysis stage drives the PMP to deploy the network-capture toolchain (tshark and CICFlowMeter), retrieve the resulting behavioural network flow, and pass it to the AI threat-detection algorithm, which classifies the traffic as cryptomining and forwards the compound event to the decision. Figure 5.46 shows the resolutive decision stage selecting and validating the corresponding CACAO remediation playbook (the cryptomining OTA-remediation playbook reported in Figure 5.47), and forwarding it to the execution stage.

```
ubuntu@smart-office:~$ kubectl -n security-functions logs -f resolutive-analysis-scl-b4fd9598c-6dgvm
[INFO] - 2026-06-19 04:39:32,122 - MQTT Broker connected (function.py:347)
[INFO] - 2026-06-19 04:39:38,448 - Analysis Function STARTING... (function.py:362)
[INFO] - 2026-06-19 04:39:38,448 - [PMP] Contacting PMP at http://10.30.8.228:8080/health ... (function.py:99)
[INFO] - 2026-06-19 04:39:38,487 - [PMP] PMP alive - reachable at 10.30.8.228:8080 (function.py:104)
[INFO] - 2026-06-19 04:39:38,488 - [PMP] Starting tool deployment sequence... (function.py:143)
[INFO] - 2026-06-19 04:39:38,488 - [PMP] Requesting deployment of 'tshark' -> POST http://10.30.8.228:8080/api/v1/network-tool (function.py:115)
[INFO] - 2026-06-19 04:39:38,499 - [PMP] tshark accepted by PMP - toolId: tshark-eth0-1781843978, status: DEPLOYING (function.py:149)
[INFO] - 2026-06-19 04:39:38,499 - [PMP] Requesting deployment of 'cicflowmeter' -> POST http://10.30.8.228:8080/api/v1/network-tool (function.py:115)
[INFO] - 2026-06-19 04:39:38,512 - [PMP] cicflowmeter accepted by PMP - toolId: cicflowmeter-eth0-1781843978, status: DEPLOYING (function.py:155)
[INFO] - 2026-06-19 04:39:38,513 - [PMP] All tools submitted for deployment: tshark (tshark-eth0-1781843978), cicflowmeter (cicflowmeter-eth0-1781843978). Polling PMP for deployment
 status... (function.py:157)
[INFO] - 2026-06-19 04:39:39,813 - [PMP] Polling deployment status - tshark: DEPLOYING, cicflowmeter: ACTIVE (function.py:169)
[INFO] - 2026-06-19 04:39:41,033 - [PMP] Polling deployment status - tshark: ACTIVE, cicflowmeter: ACTIVE (function.py:172)
[INFO] - 2026-06-19 04:39:41,034 - [PMP] Tools deployed successfully. (function.py:176)
[INFO] - 2026-06-19 04:39:41,034 - [PMP] Kafka topic for network flows (cicflowmeter): 'pmp/cicflowmeter/network-flows' (function.py:177)
[INFO] - 2026-06-19 04:39:41,034 - [Analysis] Subscribing Kafka consumer - broker: 10.30.8.228:9092, flows topic: 'pmp/cicflowmeter/network-flows' (function.py:288)
[INFO] - 2026-06-19 04:39:41,040 - [Analysis] Loop STARTED - consuming from PMP topic: flows='pmp/cicflowmeter/network-flows' (function.py:369)
[INFO] - 2026-06-19 04:39:41,040 - [Analysis] Kafka consumer ACTIVE - ingesting PMP-produced flows from 'pmp/cicflowmeter/network-flows' (function.py:305)
[INFO] - 2026-06-19 04:40:47,899 - [PMP/cicflowmeter] Received 1 flow(s) - passing to AI analysis (function.py:322)
[INFO] - 2026-06-19 04:40:49,376 - [Analysis] Inference result: Cryptomining [src=37.59.43.131:443 dst=192.168.0.7:50396 proto=TCP flow_ts=2018-12-19T19:38:25] (function.py:220)
[INFO] - 2026-06-19 04:40:49,376 - [Analysis] Cryptomining detected from 37.59.43.131 -> 192.168.0.7, device a7b4cff0-68ff-11f1-bf96-a7939e4847df. Forwarding to decision. (function.
py:243)
[INFO] - 2026-06-19 04:40:49,380 - EVENT: Cryptomining detected from 37.59.43.131 on device a7b4cff0-68ff-11f1-bf96-a7939e4847df (function.py:263)
```

Figure 5.45: Resolutive S-CL Analysis Logs - Flow retrieved and passed to AI Algorithm

```
ubuntu@smart-office:~$ kubectl -n security-functions logs -f resolutive-decision-scl-b4d777d99-cldzz
[INFO] - 2026-06-19 04:39:51,221 - MQTT Broker connected (function.py:217)
[INFO] - 2026-06-19 04:39:56,623 - Decision Function STARTED. Listening on scl_analysis_dst (function.py:234)
[INFO] - 2026-06-19 04:40:49,382 - Received compound event: Cryptomining from 37.59.43.131, device a7b4cff0-68ff-11f1-bf96-a7939e4847df (function.py:155)
[INFO] - 2026-06-19 04:40:49,409 - Playbook 'playbook--smartbulb-ota-remediation--b2c3d4e5-f6a7-4890-9bcd-ef1234567890' validated successfully. (function.py:123)
[INFO] - 2026-06-19 04:40:49,410 - EVENT: Valid playbook found for Cryptomining. Forwarding to execution for device a7b4cff0-68ff-11f1-bf96-a7939e4847df. (function.py:89)
[INFO] - 2026-06-19 04:40:49,414 - Decision forwarded to scl_decision_dst (function.py:199)
```

Figure 5.46: Resolutive S-CL Decision Logs - Playbook selected and validated

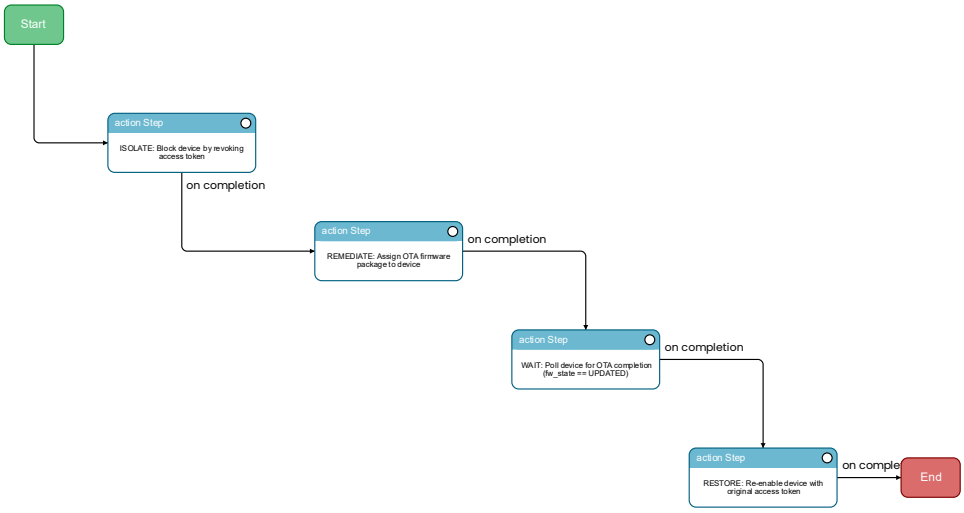

Figure 5.47: UC2 Scenario 2 Cryptomining Playbook

Finally, Figure 5.48 shows the resolutive execution stage enacting the playbook: it translates the CACAO workflow into standardised OpenC2 commands and dispatches them to the ThingsBoard OpenC2 actuator. Figure 5.49 shows the ThingsBoard OpenC2 consumer receiving and executing those commands step by step: blocking the compromised device, pushing the OTA firmware update registered in Figure 5.36, and restoring the device's credentials once remediated. This completes the validation of the escalating, two-tier response: a lightweight investigative loop detects the symptom on the IoT platform, autonomously requests an in-place security-service update through the extended SCM lifecycle manager, and a resolutive loop confirms the threat through AI-based flow analysis and remediates it via a standards-based OpenC2/CACAO actuation, all without operator intervention.

```
ubuntu@smart-office:~$ kubectl -n security-functions logs -f resolutive-execution-scl-65bb7d488b-qlg9z
[INFO] - 2026-06-19 04:39:42,421 - MQTT Broker connected (function.py:195)
[INFO] - 2026-06-19 04:39:47,539 - Execution Function STARTED. Listening on scl_decision_dst (function.py:212)
[INFO] - 2026-06-19 04:40:49,417 - Starting execution for playbook: playbook--smartbulb-ota-remediation--b2c3d4e5-f6a7-4890-9bcd-ef1234567890 (function.py:119)
[INFO] - 2026-06-19 04:40:49,418 - Runtime context keys: ['__device_id__', '__original_token__', '__ota_package_id__', '__blocked_token__'] (function.py:120)
[04:40:49] >> [System]
[04:40:49]    Initializing Context for Playbook: playbook--smartbulb-ota-remediation--b2c3d4e5-f6a7-4890-9bcd-ef1234567890
[04:40:49]    --- Applying Runtime Variable Overrides ---
[04:40:49]     >> [VariableStore]
[04:40:49]        Updated: __device_id__ = a7b4cff0-68ff-11f1-bf96-a7939e4847df
[04:40:49]        Updated: __original_token__ = qFnuYU8A4SaXaWtepDL1
[04:40:49]        Updated: __ota_package_id__ = a80dec70-68ff-11f1-bf96-a7939e4847df
[04:40:49]        Updated: __blocked_token__ = d164b8e8079307810cae096c53bccf63
[04:40:49] >> [System]
[04:40:49]    --- Runtime Variables Applied ---

--- Starting Execution of Playbook: playbook--smartbulb-ota-remediation--b2c3d4e5-f6a7-4890-9bcd-ef1234567890 ---
[04:40:49] >> [start--ota-remediation--00000000-0000-4000-8000-000000000001]
[04:40:49]    START: Start: SmartBulb OTA Remediation
[04:40:49] >> [action--block-device--00000000-0000-4000-8000-000000000002]
[04:40:49]    ACTION: ISOLATE: Block device by revoking access token
    > Resolved Inputs: {'__device_id__': 'a7b4cff0-68ff-11f1-bf96-a7939e4847df', '__blocked_token__': 'd164b8e8079307810cae096c53bccf63'}
    > Applying step_variables...
[04:40:49] >> [VariableStore]
[04:40:49]    Updated: __mqtt-topics__ = {'topic-array': ['oc2/cmd/device/tb-tenant', 'oc2/rsp']}
       > Set '__mqtt-topics__' = {'topic-array': ['oc2/cmd/device/tb-tenant', 'oc2/rsp']}
[04:40:49] >> [Command]
[04:40:49]    Executing CommandTypeEnum.OPENC2 on Agent agent-mqtt-broker--aabbccdd-1122-4333-8555-556677889900
[04:40:49]     >> [System]
[04:40:49]        Initializing new connection for Agent: MQTT Broker (OpenC2) (agent-mqtt-broker--aabbccdd-1122-4333-8555-556677889900)
[04:40:49]     >> [Agent:agent-mqtt-broker--aabbccdd-1122-4333-8555-556677889900]
[04:40:49]        Subscribing to response topics: ['oc2/rsp']
[04:40:49]     >> [OpenC2]
[04:40:49]        Resolved '__device_id__:value' to 'a7b4cff0-68ff-11f1-bf96-a7939e4847df'
[04:40:49]        Resolved '__blocked_token__:value' to 'd164b8e8079307810cae096c53bccf63'
[04:40:49]        Sending to oc2/cmd/device/tb-tenant (waiting for reply)...
[04:40:49]     >> [Agent:agent-mqtt-broker--aabbccdd-1122-4333-8555-556677889900]
[04:40:49]        MQTT Publishing to oc2/cmd/device/tb-tenant...
[04:40:49]     >> [OpenC2]
[04:40:49]        Response received from oc2/cmd/device/tb-tenant
    > Command Result: {'status': 200, 'status_text': "Set successful. Credentials updated for device 'a7b4cff0-68ff-11f1-bf96-a7939e4847df'."}
[04:40:49] >> [action--ota-update--00000000-0000-4000-8000-000000000003]
```

Figure 5.48: Resolutive S-CL Execution Logs - Playbook executed

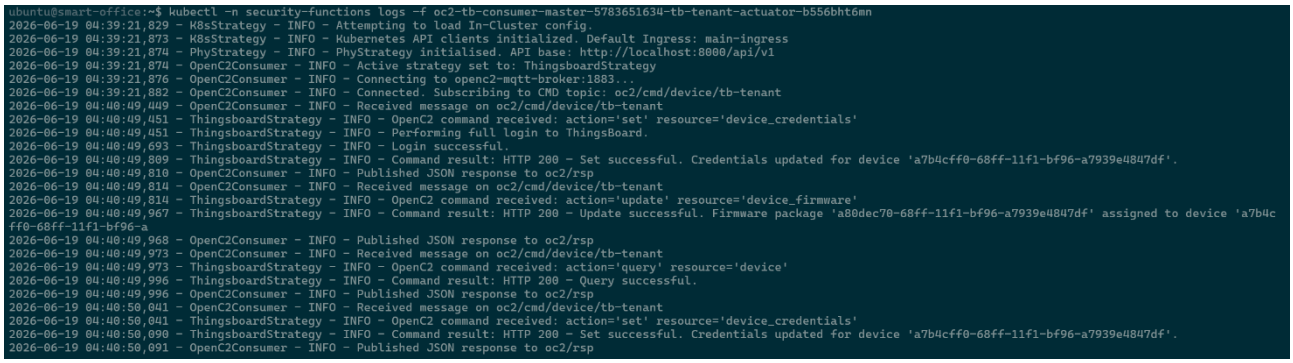

Figure 5.49: TB OpenC2 Consumer Logs - Playbook steps execution

UC2.3 Validation Outcomes

This section details the functional validation of Use Case 2 Scenario 3, where an attacker manipulates the local temperature and humidity sensor readings of a smart farm to force an erroneous irrigation behaviour, aiming to cause environmental damage and severe economic losses across a multi-tenant smart-agriculture deployment. Differently from the previous two scenarios, the validation focuses on the hierarchical coordination of multiple Security Closed Loops (S-CLs): five peripheral short loops, one per smart farm, are overseen by a single centralised long (master) loop that arbitrates their decisions and resolves conflicts. To strictly focus on the multi-loop coordination and the cross-tenant correlation, the proactive Security Service composition phase is assumed to be already completed, and the localised S-CLs and the master loop are deployed directly from the S-CL Mgmt using their onboarded descriptors. As a preliminary step, the artefacts required by the scenario are onboarded on the ZTSP. As shown in Figure 5.15, the Smart Farm Platform, consisting of 6 nodes (K3S), is registered as Platform on the S-RO. On the S-CL Mgmt Platform, the two Security Closed Loop descriptors that govern the scenario are then composed from their respective S-CLF descriptors: Figure 5.51 depicts the SHORT Farm S-CL descriptor, comprising the monitoring, analysis, decision, and execution stages of a peripheral loop, while Figure 5.52 depicts the LONG Farm S-CL descriptor, comprising the monitoring, analysis, and decision stages of the centralised master loop.

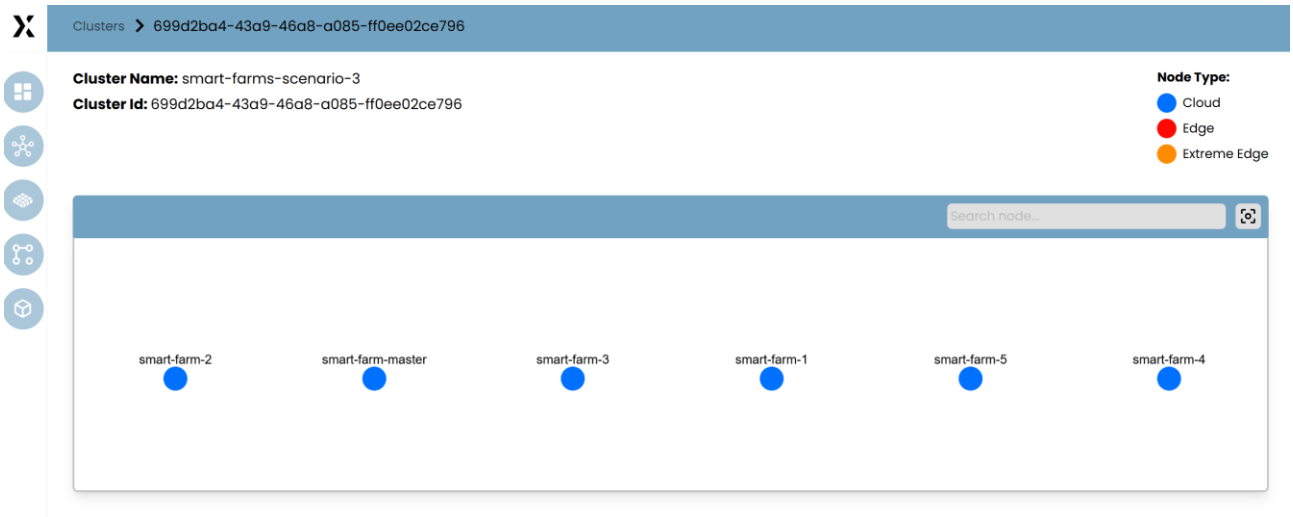

Figure 5.50: Smart Farm Platform - S-RO View

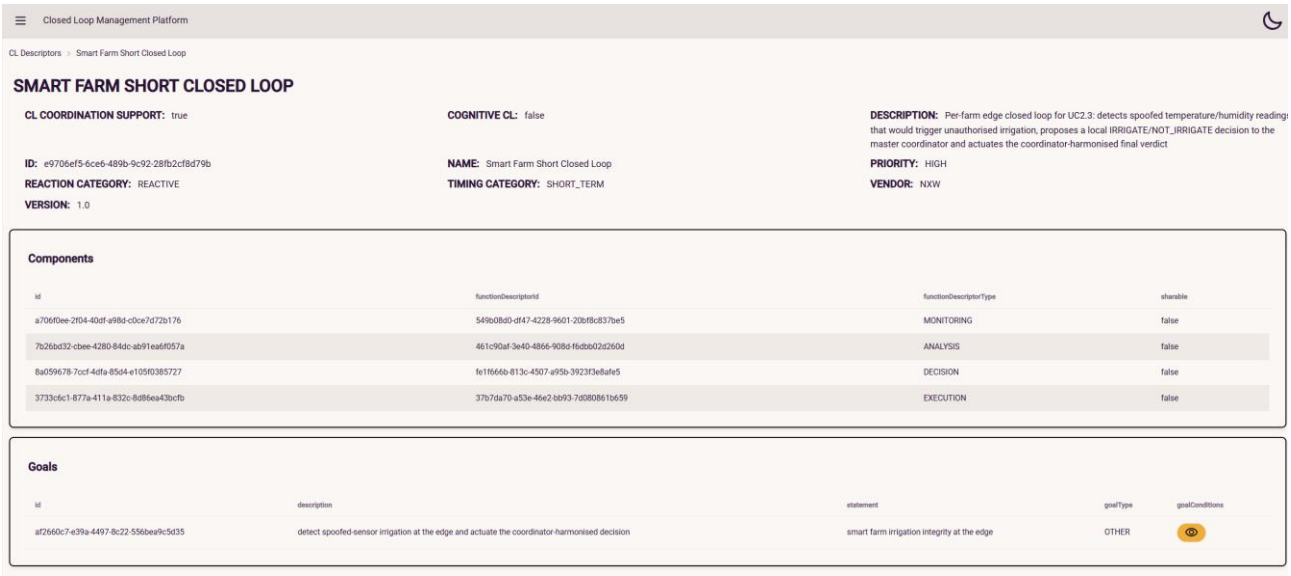

Figure 5.51: SHORT Farm S-CL Descriptor

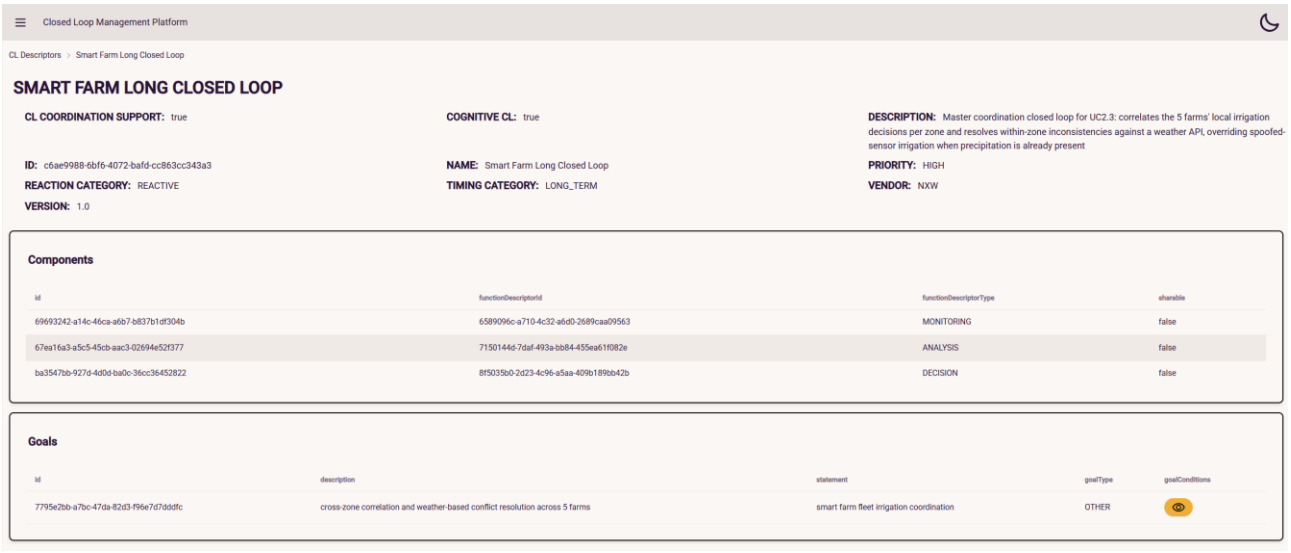

Figure 5.52: LONG Farm S-CL Descriptor

Once the S-CL descriptors and the Platform are available, the scenario infrastructure is deployed across the federated edge testbed. As depicted in Figure 5.53, the five smart farms are brought up, each running its own Smart Farm Platform pinned to the corresponding edge node. In this, Farms 1 and 2 belong to the proximity group Zone A, while Farms 3, 4, and 5 to Zone B. To complement this, a coordination bus that mediates the exchange of decisions between the peripheral loops and the master loop is deployed on the smart farm master node. The five peripheral S-CLs are then instantiated, initially without the supervising master loop, as shown in Figure 5.54: at this stage each short loop operates autonomously, scheduling its four stages on the smart farm it manages.

```
ubuntu@smart-farm-master:~$ kubectl -n smart-farm get pods -o wide
NAME                                  READY   STATUS    RESTARTS   AGE   IP           NODE                NOMINATED NODE   READINESS GATES
kafka-5f4548dc89-lnhmg                1/1     Running   0          45m   10.42.0.12   smart-farm-master   <none>           <none>
smart-farm-platform-1-9d96dbbf9-m9j8f 1/1     Running   0          42m   10.42.7.4    smart-farm-1        <none>           <none>
smart-farm-platform-2-65d88f6dd4-q4fnx 1/1     Running   0          42m   10.42.4.4    smart-farm-2        <none>           <none>
smart-farm-platform-3-65c4cc9576-pgxzz 1/1     Running   0          42m   10.42.3.4    smart-farm-3        <none>           <none>
smart-farm-platform-4-5f55ddbc9f-fvdrl 1/1     Running   0          42m   10.42.6.4    smart-farm-4        <none>           <none>
smart-farm-platform-5-68fc79f9d-d5nkn 1/1     Running   0          42m   10.42.1.4    smart-farm-5        <none>           <none>
```

Figure 5.53: Smart Farms initial Infrastructure deployment

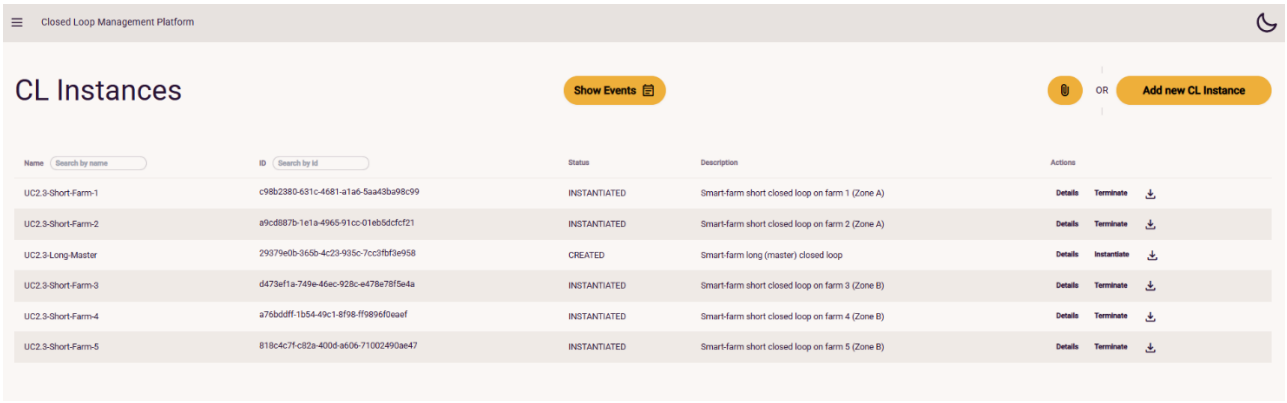

Figure 5.54: Instantiated SHORT Loops - No Coordination

To establish the nominal behaviour, the loops are first observed in the absence of any attack. Figure 5.55 reports the monitoring and analysis stages of a short loop: the monitoring function periodically grabs the temperature and humidity readings from the local Smart Farm Platform and, over a configurable observation window, forwards them to the analysis function, which aggregates them into the synthetic metrics consumed by the decision. Figure 5.56 reports the subsequent decision and execution stages: based on the aggregated metrics, the decision function computes the local irrigation verdict (IRRIGATE or NOT_IRRIGATE), and the execution function enacts it on the farm actuator. Under nominal conditions the local verdict is consistent with the real field conditions. In the example, given the low temperature and high humidity the verdict is to NOT_IRRIGATE.

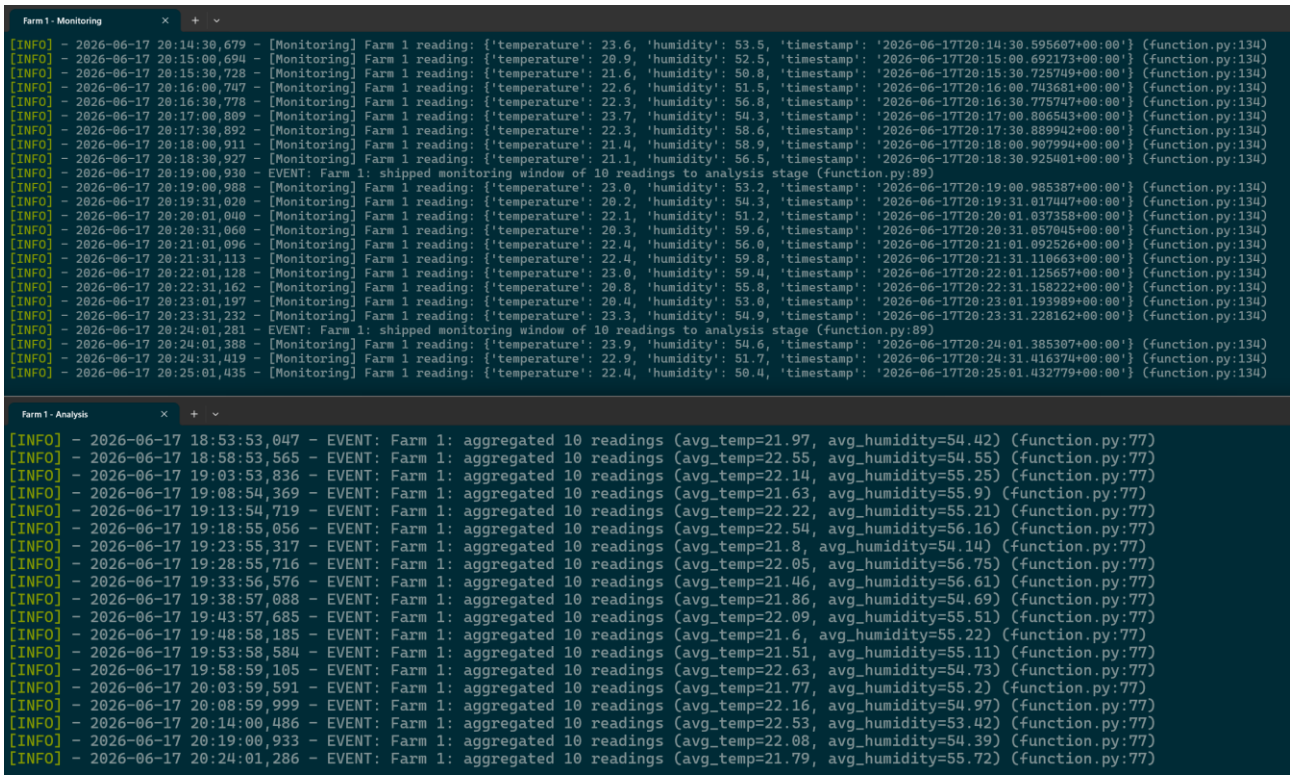

Figure 5.55: SHORT Loop Monitoring and Analysis - No Attack

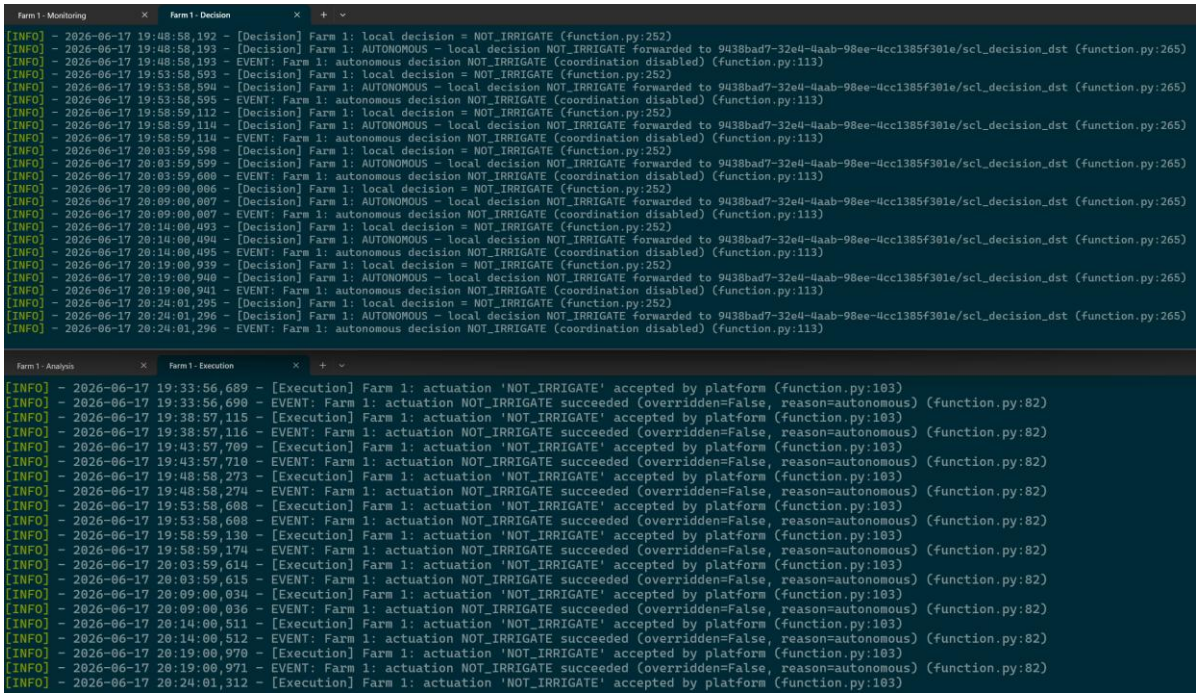

Figure 5.56: SHORT Loop Decision and Execution - No Attack

The attack is then injected on a single farm by forcing its sensors to report manipulated values and pollute the short loop. Figure 5.57 shows the monitoring and analysis stages of the targeted short loop reacting to the spoofed readings, with the aggregated metrics now reflecting the falsified field conditions. As a consequence,

Figure 5.58 shows the decision and execution stages of the same loop autonomously computing and enacting the irrigation verdict dictated by the manipulated data. This confirms that, when operating in isolation, a single peripheral loop faithfully follows its own sensors and is therefore vulnerable to a sensor-spoofing attack: the manipulated reading alone is sufficient to drive an erroneous actuation, since the loop has no means of distinguishing a genuine local microclimate event from a malicious manipulation.

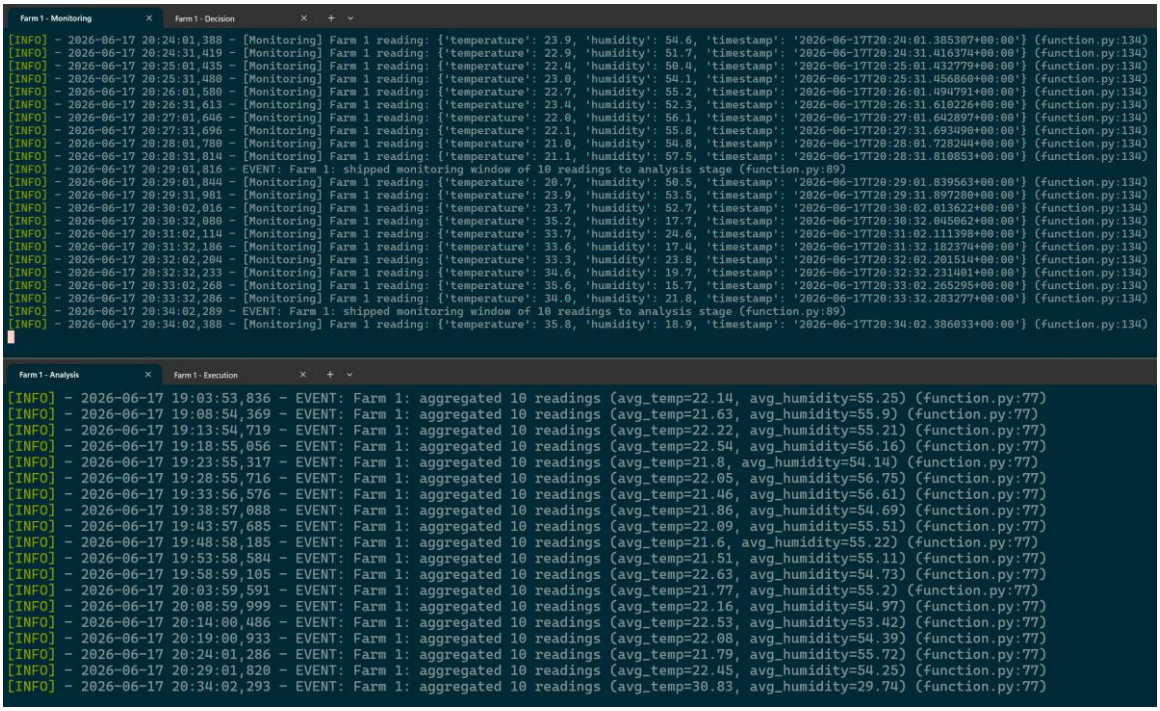

Figure 5.57: SHORT Loop Monitoring and Analysis - Attack

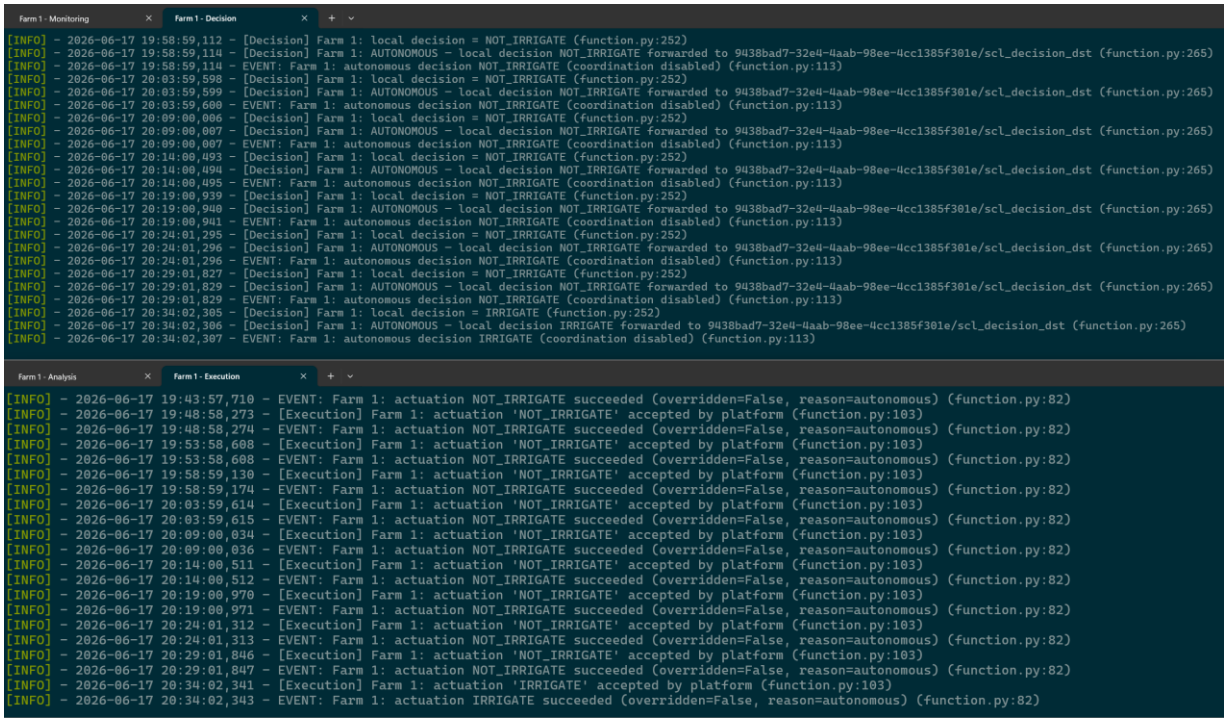

Figure 5.58: SHORT Loop Decision and Execution - Attack

The centralised master loop is then introduced to provide the cross-tenant correlation that the isolated loops lack. As shown in

Figure 5.59, the long loop is instantiated alongside the five short loops; upon start-up, it registers the peripheral loops and announces itself on the coordination bus, switching the short loops decision stages from autonomous to coordinated operation. From this point on, instead of enacting its local verdict directly, each peripheral decision stage publishes its proposed decision, enriched with the farm identifier, its geographical zone and coordinates, and a timestamp, to the coordination bus, and waits for the harmonized verdict returned by the master loop before triggering the execution stage.

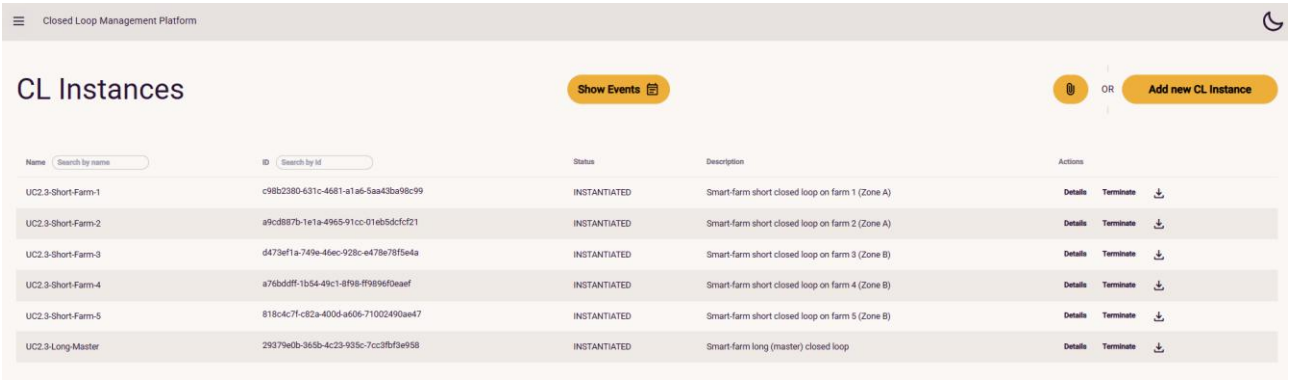

Figure 5.59: Instantiated Short Loops and Long Loop

Figure 5.60 depicts the decision stage of the attacked Farm 1 under coordination: the local proposal is sent to the coordinator, and the verdict received back mitigates the attack, overriding the erroneous local decision. The conflict-resolution logic enacted by the master loop is detailed in

Figure 5.61 and

Figure 5.62. The master monitoring stage continuously collects the decisions published by the five farms and, once a complete batch is available, passes it to the master analysis stage, which cross-references the local verdicts by geographical zone. As shown in

Figure 5.61, when the verdicts of two farms belonging to the same proximity group diverge (in this case Farm 1 disagreeing with the adjacent Farm 2 of Zone A) the analysis flags the zone as inconsistent and reports the conflicting farms to the master decision stage. reports the master decision stage resolving the inconsistency: by enriching the aggregated decisions with the live meteorological information queried from an external weather service for the geographical position of the affected farms, the master loop identifies the compromised local loop and formulates a harmonized verdict, overriding the manipulated decision and pushing it back to the targeted farm over the coordination bus for execution.

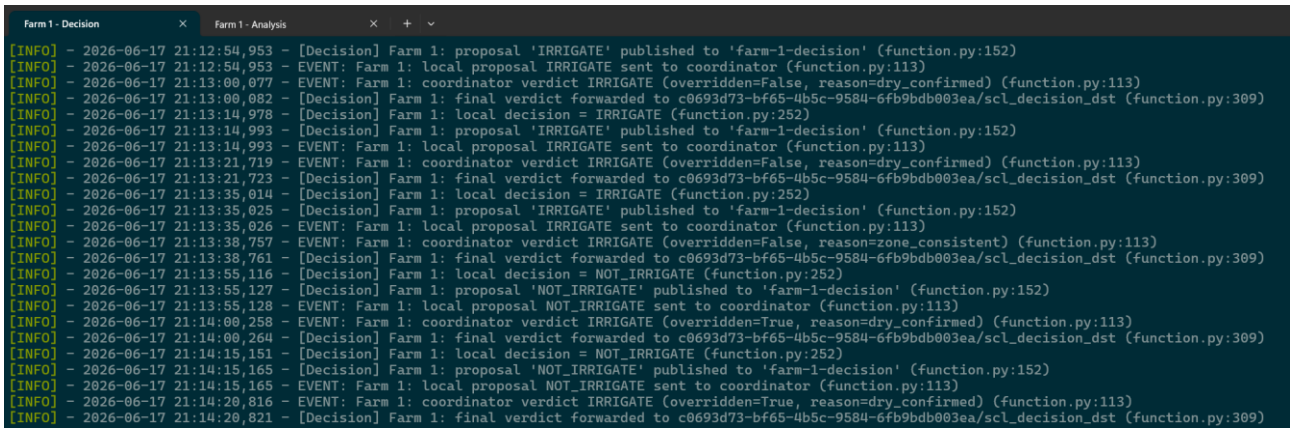

Figure 5.60: Farm 1 Decision - Attack Mitigated by Coordination

Figure 5.61: Master Loop Analysis - Inconsistency detection

Figure 5.62: Master Loop Decision - Local Farm decision ovveride

This confirms the platform's capability to coordinate five interacting closed loops concurrently and to suppress a sensor-spoofing attack through cross-tenant, weather-aware correlation, fulfilling the final multi-loop coordination objective of the Zero-Touch Security Platform. It is worth noting that the scope of this validation was deliberately confined to the Security Closed Loops and to their hierarchical coordination, rather than to the internal sophistication of the individual stages or the overall scenario setting. The peripheral loops therefore implement an intentionally lightweight monitoring–analysis–decision logic, sufficient to exercise the coordination mechanism while keeping the demonstration focused on the interaction between the local loops and the master loop. The same coordination framework is, however, agnostic to the complexity of the loops it governs: each farm could equally host a fully-fledged reactive Security Closed Loop, as validated in Scenario 1, or adopt

the escalation strategy demonstrated in Scenario 2, where the detection of a possible spoofing on a given farm dynamically triggers the on-demand composition and deployment of a dedicated resolutive loop on the affected node. In all such configurations the role of the master loop is unchanged, confirming that the validated coordination mechanism constitutes a general substrate on top of which arbitrarily rich local remediation strategies can be layered. The integration of the data flowing through Kafka is achieved through a structured semantic lifting process. The first step involved defining an ontology (Figure 5.63) that captures the conceptual model underlying the farm data: the core classes — such as ShortLoopDecision, LongLoopDecision, FarmAgent, SensorMetrics, and ClosedLoopInstance — along with their properties, establish a shared vocabulary that makes the raw Kafka payloads semantically interpretable and interoperable.

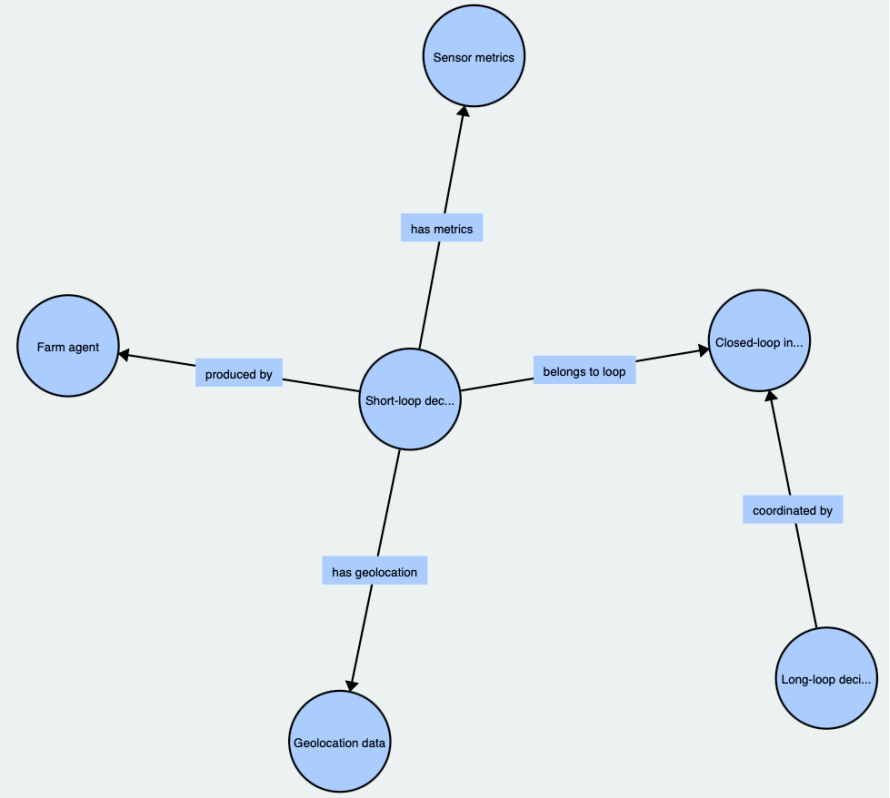

Figure 5.63: Farm data ontology

Building on this ontological foundation, two RML mappings were defined within the Data fabric (Figure 5.64).

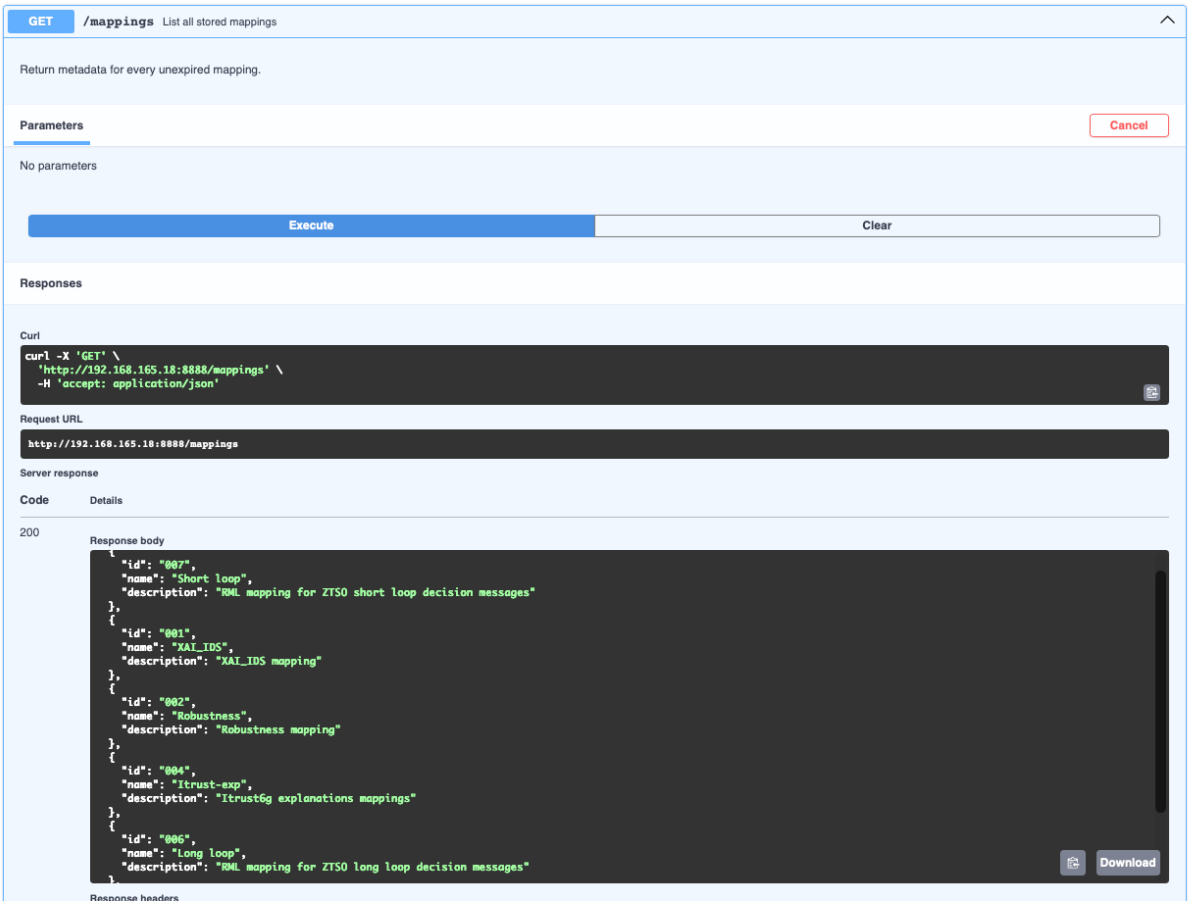

Figure 5.64: Data Fabric RML mapping storage snapshot

The short-loop mapping handles the farm-x-decision topics, lifting the local telemetry and initial irrigation decisions produced by each farm agent into structured RDF triples. The long-loop mapping targets the farm-x-coordination topics, transforming the coordinated decisions issued by the central coordinator — including override flags, conflict resolution reasons, and meteorological signals — into the corresponding ontology instances. Each Kafka topic, one short-loop and one long-loop per farm, was then associated with its respective mapping. As messages flow through the topics, the RML engine processes each payload and converts it into RDF (Figure 5.65) according to the defined mappings, ingesting the resulting triples directly into GraphDB (Figure 5.66). This approach ensures that data produced across all five farms is continuously lifted from its raw JSON form and made available in the knowledge graph as semantically rich, queryable RDF data, enabling consistent reasoning and analysis across the entire multi-farm coordination system.

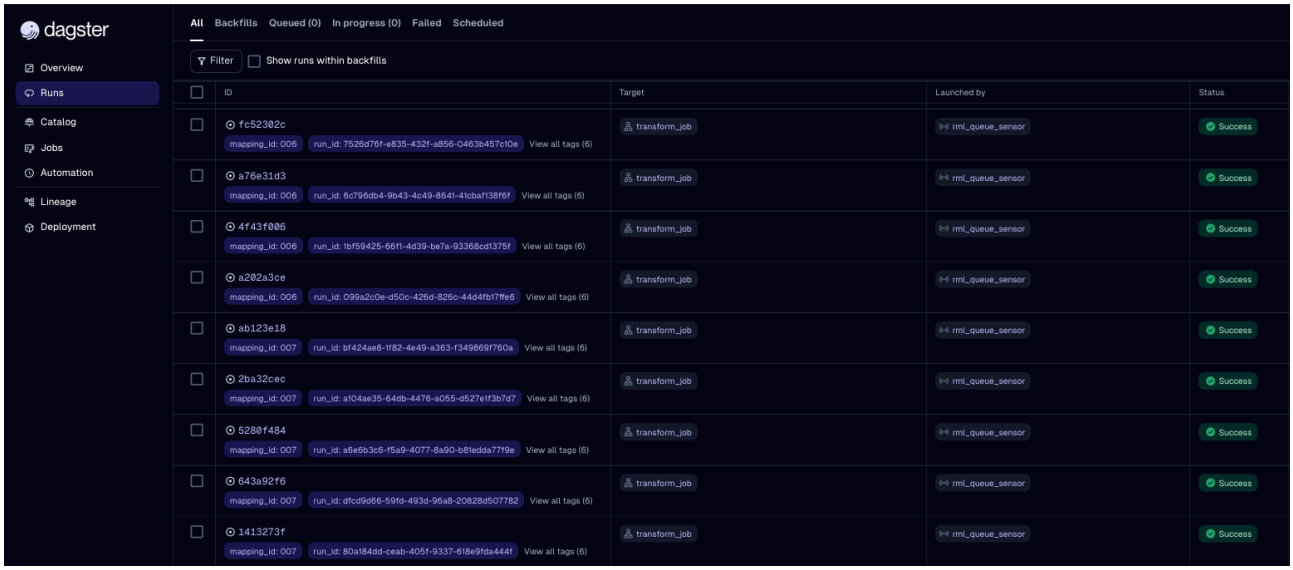

Figure 5.65: Data fabric Dagster data lifting pipelines

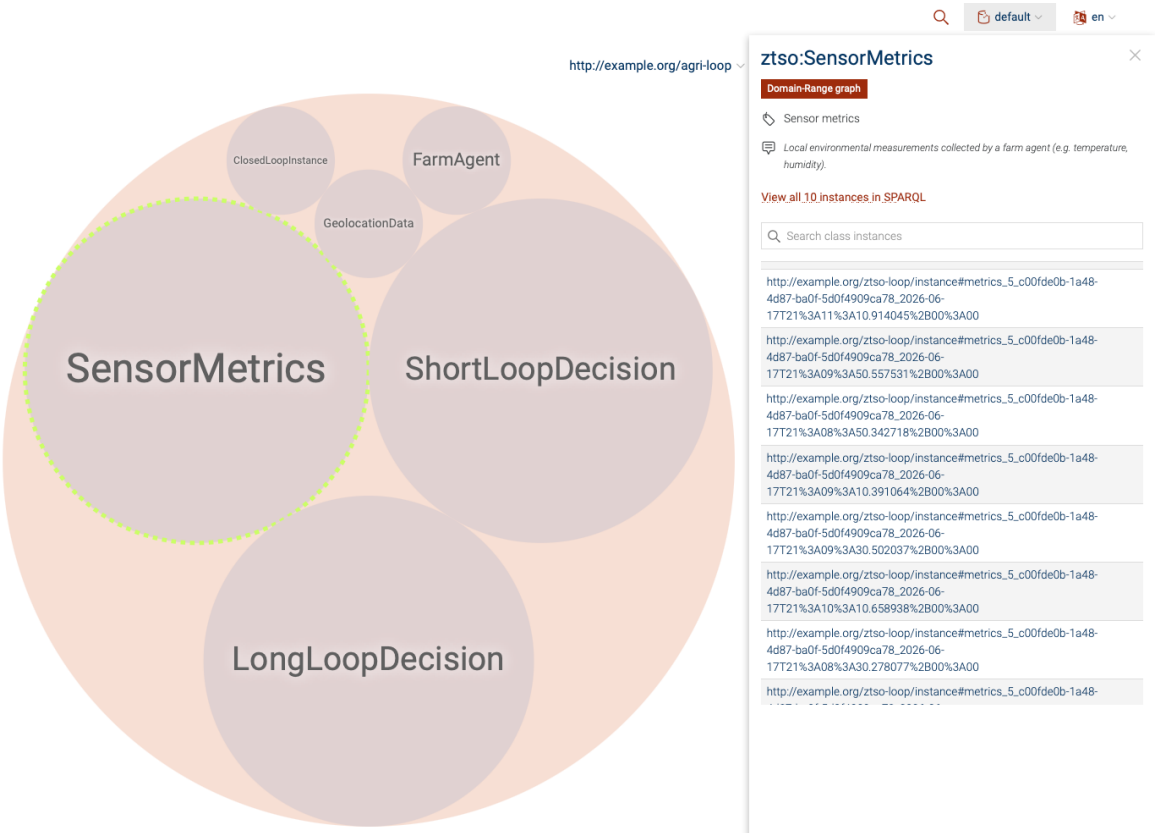

Figure 5.66: Farm data in the Data Fabric GraphDB

5.1.3 Prototype 3: NetSecaaS Gateway

This section assesses Prototype 3 from a functional validation perspective. The objective is to verify that the NetSecaaS Gateway correctly supports its intended role within the ROBUST-6G Exposure Framework, namely the controlled exposure of security capabilities towards external consumers through a northbound interface. The validation therefore focuses on the functional behaviour of the exposure path, the correct interaction with the Data Governance and Data Fabric components, and the mediation towards the orchestration side represented by ZTSO. Performance-oriented measurements, such as latency and CPU usage, are considered only as supporting evidence of

operational feasibility and are not the main focus of this prototype-level assessment, since the detailed performance validation is addressed later in the UC3 validation section.

In this scope, Prototype 3 is validated as an integrated exposure and mediation layer rather than as an isolated API endpoint. The validation confirms whether a third-party request can be received, interpreted, checked against the appropriate governance logic, routed towards the relevant platform capability, and transformed when needed into the proper internal representation for data retrieval or service triggering.

5.1.3.1 Validation Setup

The validation setup for Prototype 3 is designed to demonstrate the ROBUST-6G Exposure Layer. The execution environment integrates the complete stack of the NetSecaaS, Data Governance and Data Fabric which is triggered once a 3rd party ask to access a ROBUST-6G capability.

5.1.3.1.1 Testbed Configuration

The validation setup for Prototype 3 is distributed across two partner domains. The NetSecaaS Gateway and the data plane components are deployed in the TID testbed (TTID01 / logical area TID01), while the Security Orchestrator side is represented by the ZTSO component deployed in the Nextworks testbed (TNXW01). The setup diagram in Figure 5.67 shows the NetSecaaS block exposing a Northbound Interface (NBI), backed by the Transformation Function, and connected southbound to the Data Fabric, the Data Governance component, and the remote ZTSO instance. Two complementary KPI tests were executed on top of this deployment. First, the latency test was executed from an external client located outside the API host, so that the reported values reflect client-observed remote latency of the exposed endpoints rather than loopback-only behaviour. Second, the CPU test was executed on the same host as the API endpoints, so that CPU usage reflects the load actually produced on the machine running the NetSecaaS service during request processing. This distinction is important because it separates network-observed service responsiveness from local computational overhead, thereby covering the two most relevant operational performance dimensions of the exposure layer.

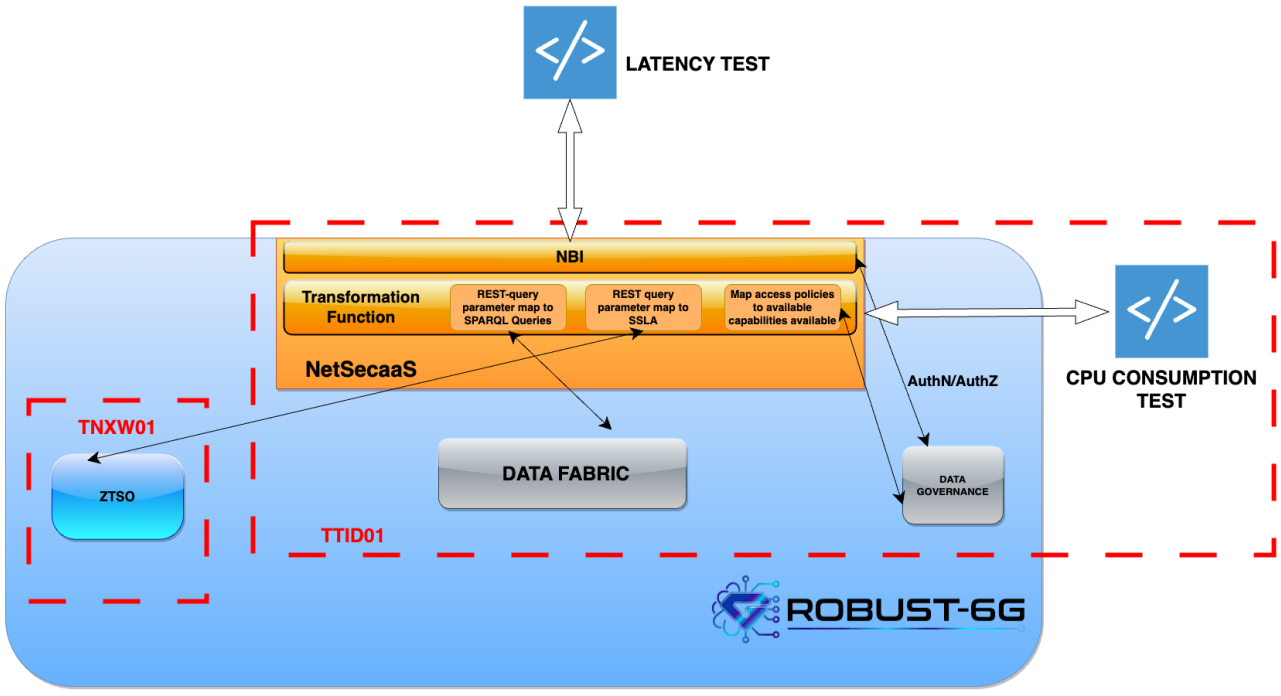

Figure 5.67:Validation setup for Prototype 3

5.1.3.1.2 Datasets

Prototype 3 does not rely on a static benchmarking dataset in the same way as AI-training or PHY-validation components.

For this prototype, the relevant validation assets are the live API exposure path, the configured endpoints, the capability metadata available through the governance plane, the semantically integrated data accessible through the Data Fabric, and the service-configuration intents that can be transformed into orchestration artefacts.

Consequently, the validation exercises functional request flows instead of dataset-driven model evaluation. The input traffic is used to trigger the exposure functions and to verify that the gateway can route, transform and return information consistently. The default endpoints used during the validation, including the root endpoint, the data endpoint and the functions endpoint, are therefore interpreted as representative access points for checking the availability and coherence of the exposed NetSecaaS functions.

5.1.3.1.3 Integration Details

The integration under test corresponds to the architecture defined for Prototype 3. At the northbound side, the NetSecaaS NBI receives HTTP requests from a third-party consumer. The Transformation Function then interprets the request and selects the appropriate internal path: data-oriented requests are mediated towards the Data Fabric, service-oriented requests can be transformed into SSLA artefacts towards the orchestration side, and exposure responses are filtered or constrained through the Data Governance logic. Authentication and authorisation are considered part of the controlled exposure process before the corresponding data retrieval or action path is completed.

This integration validates the gateway as a mediation component and not as a standalone REST front-end. The relevant functional behaviour is the ability to connect external requests with internal ROBUST-6G capabilities while preserving the intended separation between exposure, governance, data access and orchestration. The validation therefore checks that the gateway can receive a request, identify the target logical capability, invoke the appropriate backing component, and return a coherent response to the external consumer. Detailed threshold-based performance assessment is left to the UC3 validation section.

5.1.3.1.4 Inputs and Outputs

The inputs to the Prototype 3 functional validation are: (i) the base Uniform Resource Locator (URL) of the deployed NetSecaaS API; (ii) the selected endpoint list; (iii) the request parameters required to exercise the intended exposure flow; (iv) the authentication material, when needed, in the form of a bearer token or username/password credentials.

The outputs of the validation are interpreted primarily as functional evidence. They include successful responses from the selected endpoints, confirmation that the gateway can access or mediate the related platform capability, evidence that requests are routed through the expected internal path, and confirmation that the returned information is coherent with the requested operation.

5.1.3.2 Validation Outcomes

The validation outcomes confirm that Prototype 3 fulfils its intended functional role as the ROBUST-6G exposure and mediation layer for security capabilities. The prototype demonstrates that an external consumer can interact with the platform through the NetSecaaS Gateway and that the request can be mediated towards the appropriate internal function depending on its purpose. This validates the main architectural assumption of Prototype 3: third-party access to ROBUST-6G capabilities can

be centralised through a controlled gateway, rather than requiring direct exposure of internal platform components.

The first functional outcome is the validation of the northbound exposure path. The NetSecaaS interface provides a single access point through which external requests can be submitted (Figure 5.68). The successful execution of requests against the representative endpoints confirms that the gateway is reachable, that the exposed functions are accessible through the expected interface, and that the platform can return coherent responses to the requesting party.

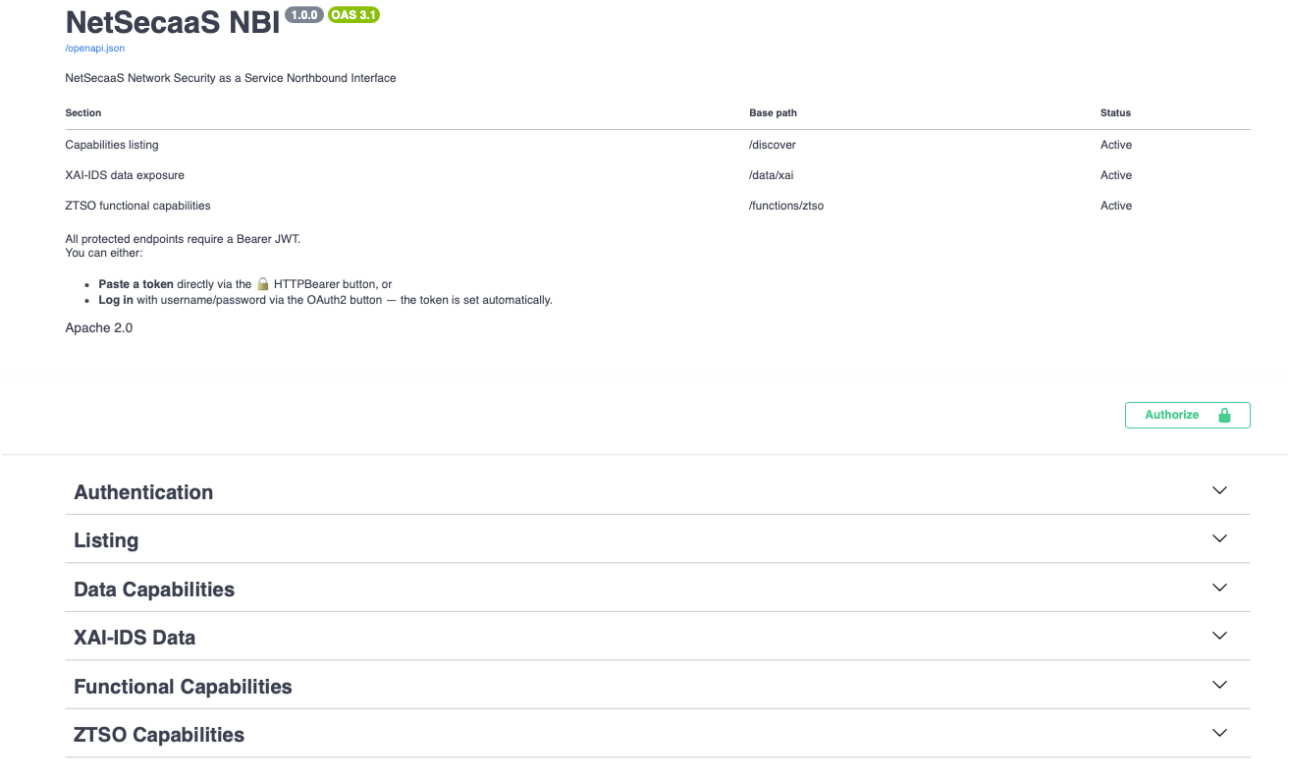

Figure 5.68: NetSecaas NBI API UI

The second functional outcome is the validation of the mediation logic. The Transformation Function supports the interpretation of incoming requests and the selection of the corresponding internal path. Depending on the request, the gateway can mediate access towards data exposed through the Data Fabric (Figure 5.69), apply governance constraints through the Data Governance component, or support the formulation of service-oriented interactions towards ZTSO (Figure 5.70).

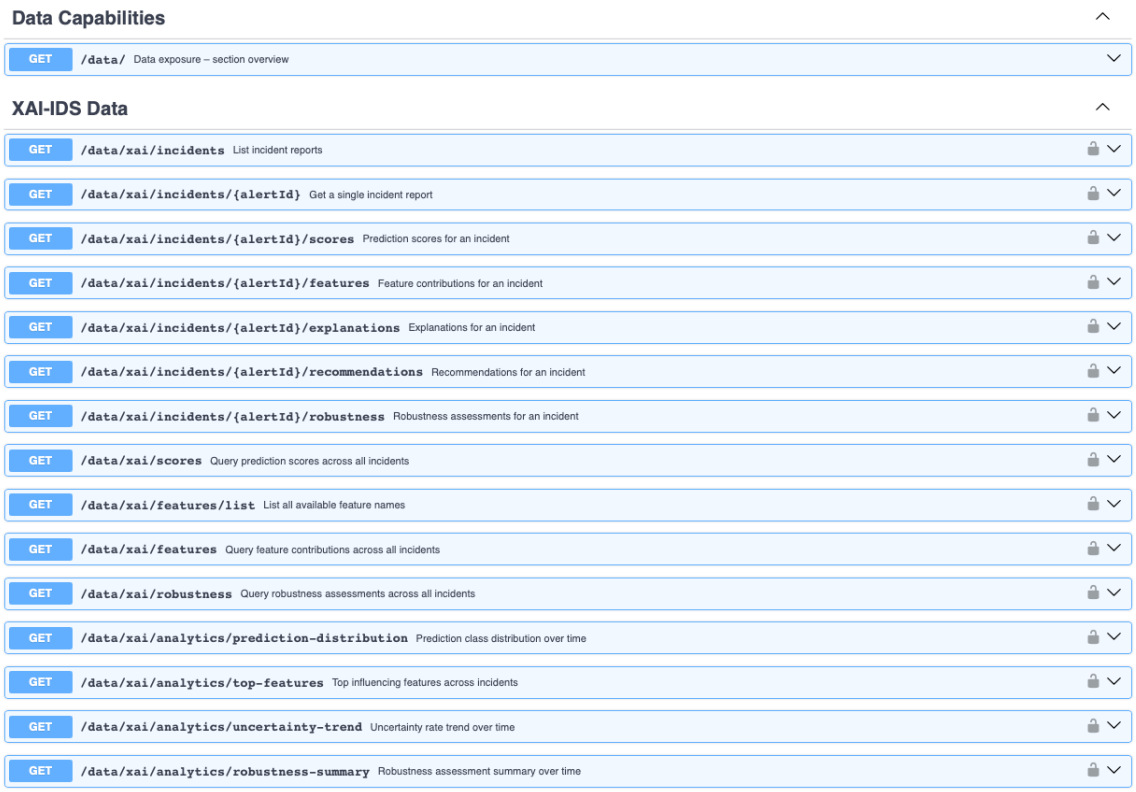

Figure 5.69:NetSecaaS API Data endpoints

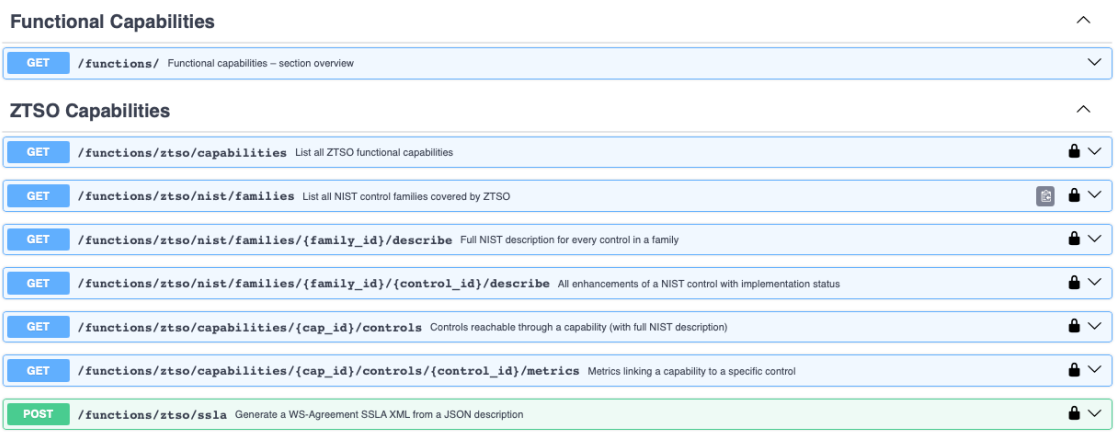

Figure 5.70:NetSecaaS API Functional endpoints

This confirms that Prototype 3 implements the expected bridge between external API consumption and internal ROBUST-6G platform capabilities.

The third functional outcome is the validation of controlled exposure. Prototype 3 does not simply publish platform functions as isolated endpoints; it exposes them through a governance-aware entry point. This is important for UC3 because the value of NetSecaaS resides in enabling third-party consumption of security capabilities while maintaining control over how data and services are accessed (Figure 5.71). The functional validation therefore confirms that exposure, governance and mediation are exercised together as part of a single workflow.

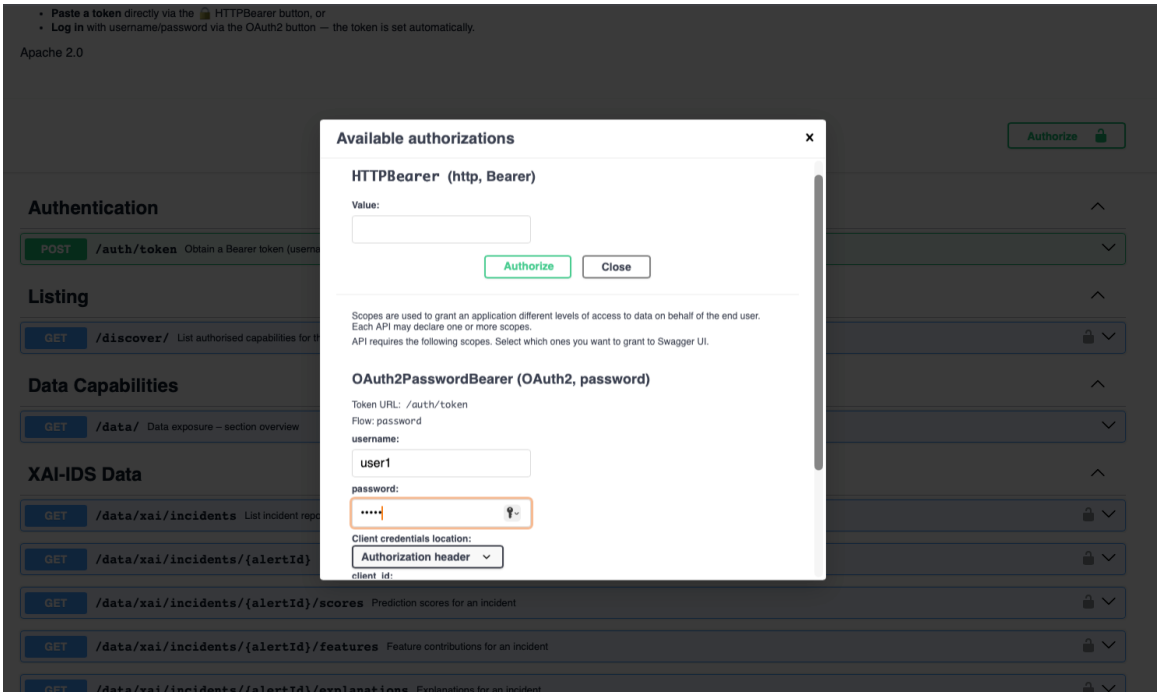

Figure 5.71: NetSecaaS AuthN/AuthZ endpoint

The fourth functional outcome concerns the CAMARA-like exposure objective. The validation supports the conclusion that the NetSecaaS framework provides a suitable mechanism for exposing ROBUST-6G capabilities through API-based access patterns. Although a precise quantitative mapping of all platform capabilities is not the focus of this subsection, the integration with Data Fabric and ZTSO demonstrates that the gateway can reach both data-oriented (Figure 5.72) and service-oriented capabilities (Figure 5.73). This provides qualitative evidence that the exposure approach is sufficiently general to cover a substantial part of the platform functionality.

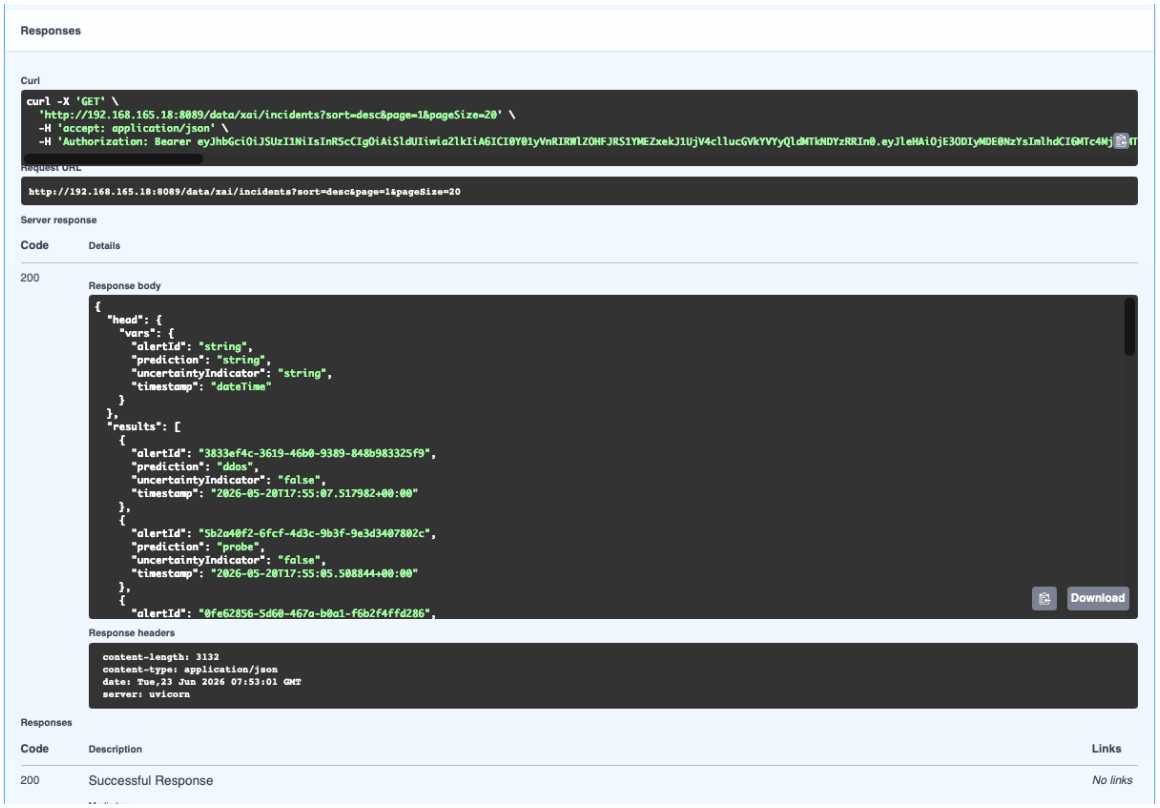

Figure 5.72:NetSecaaS API Data Endpoint response

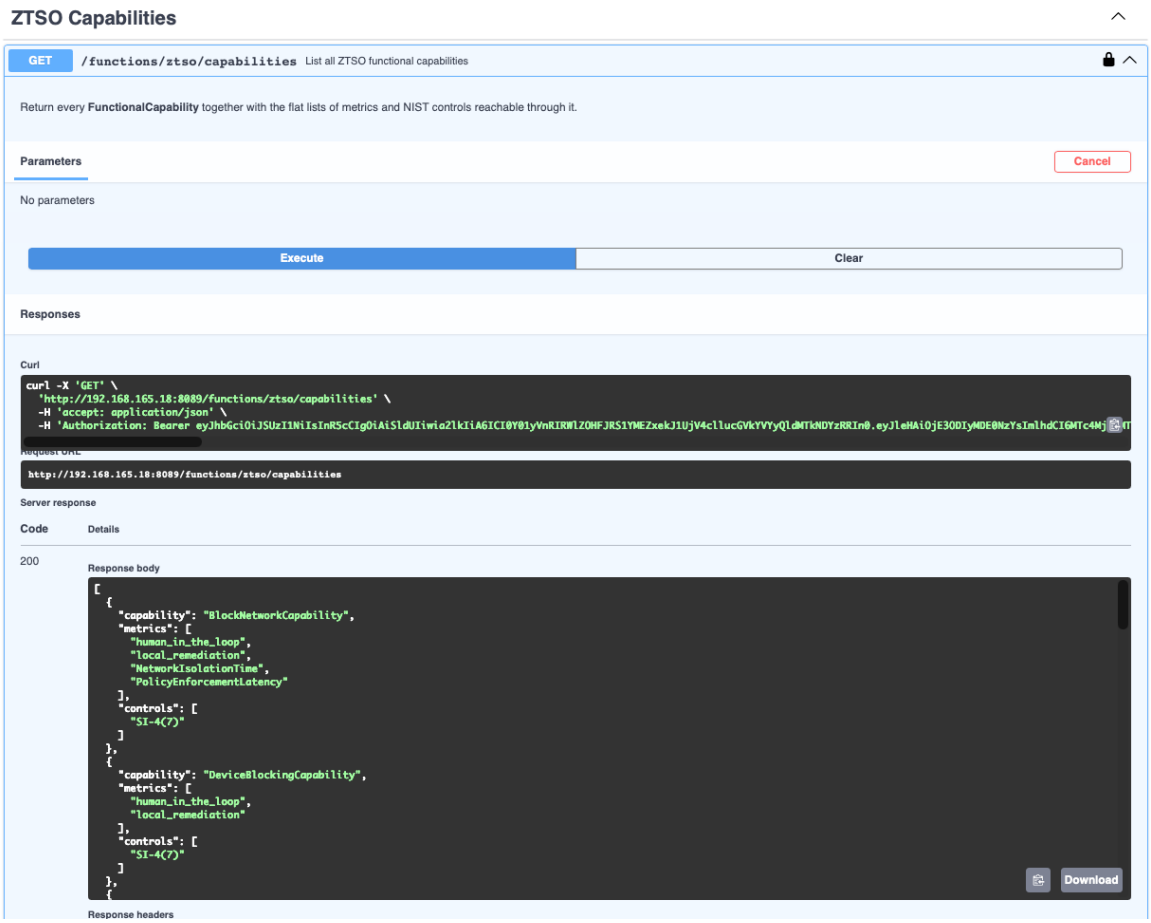

Figure 5.73:NetSecaaS API Functional endpoint response

Performance measurements collected during the prototype campaign are retained only as complementary evidence that the functional flows can be executed under realistic operating conditions. They are not used here as the main validation argument, in order to avoid overlap with the UC3 KPI validation. The prototype-level conclusion is therefore functional: NetSecaaS correctly supports controlled third-party access, request mediation, governance-aware exposure and connection to the relevant ROBUST-6G platform capabilities.

5.1.4 Prototype 4: Physical and Sensing Layer Trustworthiness

In this section, we assess the outcomes of the Physical and Sensing Layer Trustworthiness prototype demonstration, reporting on the objectives achieved, the results collected, and the limitations identified during execution. The prototype implements three complementary physical-layer security capabilities operating on real channel state information (CSI) measurements, namely: i) jamming detection and localisation by means of a generalised likelihood ratio test (GLRT) evaluated over a fine spatial grid, complemented by a windowed-limited cumulative sum (WL-CUSUM) temporal statistic; ii) spoofing detection based on the consistency of angle-of-arrival (AoA) estimates, obtained through Root-MUSIC and refined by array calibration; and iii) physical-layer secret key generation (SKG) employing Polar-CRC information reconciliation followed by privacy amplification through Davies–Meyer hashing (AES-128). With a view to maximising accessibility for consortium partners and facilitating integration with Prototype 5, the demonstrator has been deployed as a publicly reachable, browser-accessible testbed exposing both an interactive graphical front-end and a stateless representational state transfer (REST) application programming interface (API), the latter being formally specified in OpenAPI 3.1.

5.1.4.1 Validation Setup

In this section, we describe the setup used for the validation of this prototype in detail. The validation campaign was designed around two complementary access modalities served by a single deployment, so that the identical algorithmic engines could be exercised both interactively and programmatically. To this end, the demonstrator is installed on dedicated server infrastructure hosted by the ETIS Laboratory (ENSEA / CY Cergy Paris Université) and made reachable at the public domain https://robust6g-demo.etis-lab.fr/ under transport layer security (TLS). The interactive front-end permits the manual placement of network nodes on the measurement grid, the selection of attack scenarios, and the visual inspection of detection outcomes, whereas the REST interface, rooted at /api/v1/, accepts self-contained JavaScript Object Notation (JSON) requests carrying node positions and operating-point parameters and returns structured detection verdicts; the two modalities were verified to produce numerically identical results for identical scene descriptions, since both invoke the same underlying detector implementations and the same canonical operating points.

All experiments were conducted on a measured CSI dataset acquired with a 64-element uniform linear array (ULA) at a carrier frequency of 2.61 GHz. The measurement region comprises a 24 × 24 grid of candidate user positions (576 points in total) with a spatial step of 0.115 m, corresponding to one carrier wavelength, and spanning approximately $X \in [-1.44, 1.36]$ m and $Y \in [1.16, 4.03]$ m in the array reference frame; one hundred channel snapshots are available per grid position. Two dataset variants are employed: the first provides single-direction uplink observations and drives the jamming and spoofing detectors, whilst the second contains paired uplink and downlink observations and thereby furnishes the channel reciprocity exploited by the SKG pipeline. Furthermore, the validation adopts three canonical operating points, aligned between the graphical interface and the API so as to guarantee the comparability of results across access modalities: the receive signal-to-noise ratio (SNR) is set to 20 dB, 25 dB, or 30 dB for the Low, Medium, and High presets respectively, and the jammer transmit power to 7 dBm, 15 dBm, or 24 dBm correspondingly.

5.1.4.1.1 Testbed Configuration

In this section, we identify the partner testbed assets used for this prototype, how they are configured, and how the nodes are interconnected. The hosting asset is a virtualised server provisioned by the ETIS information technology service on the CY Cergy Paris Université infrastructure, running Ubuntu 26.04 LTS with eight virtual central processing unit (CPU) cores, 7.2 GiB of random-access memory (RAM), and 79 GB of storage; no graphics processing unit is required. The server is assigned the fully qualified domain name robust6g-demo.etis-lab.fr, and an Apache 2.4 reverse proxy constitutes the sole public ingress, terminating TLS on port 443 with a Let's Encrypt certificate and enforcing redirection of plain hypertext transfer protocol (HTTP) traffic to its secure counterpart. All functional services are bound exclusively to the loopback interface and are therefore reachable only through the proxy, namely: i) a TigerVNC server rendering a 1920 × 1080 virtual display on which the Tkinter-based demonstrator application is launched automatically at session start; ii) a websockify bridge translating the virtual network computing (VNC) stream into WebSocket frames consumed by the noVNC client embedded in any modern browser, thereby removing the need for partners to install client software; and iii) a uvicorn application server hosting the FastAPI implementation of the stateless REST interface on an internal port, exposed externally under the /api/v1/ path together with its interactive Swagger documentation. Each of these services is supervised by systemd with automatic restart upon termination and is enabled at boot, a configuration whose resilience was verified by a complete server reboot after which all services resumed without manual intervention.

Within the demonstrator, the emulated network comprises five logical nodes interconnected through the measured propagation environment rather than through physical links: the 64-element ULA base station (assuming the role of Bob in the SKG protocol) is positioned at the right-hand edge of the measurement grid with an inter-element spacing of 0.07 m, whilst the legitimate user equipment (Alice), the jammer, the spoofer, and the passive eavesdropper (Eve) are single-antenna terminals that may be placed at any of the 576 grid positions, each placement being snapped to the nearest dataset measurement point. The interconnection of external actors with the testbed proceeds along two paths that converge on the same engine layer: in the interactive modality, the partner's browser communicates over HTTPS with the Apache proxy, which relays the session through websockify to the VNC display hosting the graphical application; in the programmatic modality, the partner's client issues JSON requests over HTTPS to the /api/v1/ endpoints, which are forwarded by the proxy to the API service. Moreover, the API process operates on its own working copy of the codebase, its own Python virtual environment, and its own dataset symbolic links, in complete isolation from the graphical session, such that programmatic load generated by one partner cannot perturb the state observed by another partner interacting with the graphical front-end.

5.1.4.1.2 Datasets

In this section, we identify the datasets used to drive the validation of this prototype, including their source, contributing partner, and intended use. A foundational design decision for this prototype was to exercise all three security capabilities against real, measured CSI rather than against synthetically generated or purely statistical channel models. This decision was taken with a view to ensuring that the detection and key-generation algorithms confront the genuine spatial structure, correlation, and imperfections of an indoor propagation environment, since the discriminative power of angle-of-arrival (AoA) authentication and the reciprocity exploited by physical-layer key generation are both highly sensitive to the realism of the underlying channel. To this end, the prototype draws upon the publicly available "Ultra Dense Indoor MaMIMO CSI Dataset", published on IEEE DataPort by Sibren De Bast and Sofie Pollin of KU Leuven (ESAT–WaveCORE) under digital object identifier (DOI) 10.21227/nr6k-8r78, and accessible at https://ieee-dataport.org/open-access/ultra-dense-indoor-mamimo-csi-dataset.

The source acquisition was performed on the 64-antenna KU Leuven Massive multiple-input multiple-output (MaMIMO) testbed, which provides a rich and densely sampled body of measurements well suited to the spatial methods under demonstration. In this testbed, the base station is equipped with 64 antennas, each of which receives a predefined pilot signal transmitted from every user position, and from these pilots the CSI is estimated for one hundred subcarriers spaced evenly in frequency over a 20 MHz bandwidth, so that the complex-valued channel matrix for a given location spans the number of base-station antennas by the number of subcarriers. The principal distinguishing characteristic of this dataset, and the property that motivates its selection, is the exceptional density and accuracy of its spatial labelling. The user channels were collected by mounting single-antenna user-equipment terminals upon computer numerical control (CNC) XY-tables, whose antennas were advanced along a predefined zig-zag route in steps of 5 mm, yielding a positional-label accuracy of better than 1 mm and a total of 252,004 spatially labelled samples per measured topology. Such fine-grained, accurately labelled spatial sampling is precisely what permits a meaningful evaluation of AoA-based discrimination between closely spaced legitimate and adversarial positions.

The source testbed is, moreover, notable for the flexibility of its antenna deployment, which gives rise to several distinct array topologies within the same measurement campaign. These comprise a uniform rectangular array of eight-by-eight antennas, captured in both line-of-sight and non-line-of-

sight conditions by means of a metal blocker; a uniform linear array of 64 antennas arranged on a single line; and a distributed deployment in which the antennas are spread across the room in pairs. Of these, the present prototype employs the uniform linear array (ULA) topology in its line-of-sight configuration, this choice being dictated by the methods under demonstration: a linear aperture furnishes a well-defined one-dimensional angular response that is directly amenable to the Root-MUSIC AoA estimation and the spatial generalised likelihood ratio test (GLRT) at the heart of the spoofing- and jamming-detection capabilities. Furthermore, the measurement parameters of the source are of direct relevance to the prototype's physical model. The base station was configured to operate at a centre frequency of 2.61 GHz, corresponding to a wavelength of approximately 114.56 mm, and the spatial origin was defined at the centre of the array, with the positions of users and antennas measured relative to that origin and provided in three dimensions. The wavelength of roughly 0.115 m is significant in its own right, since it sets the natural spatial scale of the demonstrator: the measurement grid spacing was chosen to coincide with one wavelength, and the half-wavelength spatial-decorrelation distance underpins the security argument of the secret-key-generation (SKG) capability against a displaced eavesdropper.

From this full source acquisition, two derived datasets were prepared by the WP6 team at ENSEA, each tailored to a distinct subset of the prototype's capabilities. The first, denoted herein the AoA dataset, comprises single-direction uplink observations arranged over a 24 × 24 grid of 576 candidate user positions at a uniform spacing of one wavelength (0.115 m), with one hundred channel snapshots retained per position; it spans approximately $X \in [-1.44, 1.36]$ m and $Y \in [1.16, 4.03]$ m in the array reference frame and is used by both the jamming detector and the spoofing detector. The second, denoted the SKG dataset, comprises paired uplink and downlink observations over the same spatial region and thereby furnishes the bidirectional channel reciprocity upon which physical-layer key agreement depends; being obliged to carry both link directions, it is the larger of the two derived files. In both cases, the derivation consisted of selecting the ULA line-of-sight topology, sub-sampling the densely measured source grid onto the demonstrator's working grid and reformatting the retained measurements into the compressed array representation used by the engine layer, together with the accompanying antenna-position metadata required by the AoA estimator.

5.1.4.1.3 Integration Details

In this section, we describe the integration points between the components of this prototype as configured for the final validation, noting any changes since D6.2. The prototype is organised around a single shared engine layer, within which reside the three independent algorithmic capabilities, namely: i) the jamming detector, implementing a spatial GLRT complemented by a windowed-limited cumulative sum (WL-CUSUM) temporal statistic; ii) the spoofing detector, performing Root-MUSIC AoA estimation refined by array calibration; and iii) the SKG engine, executing Polar-CRC information reconciliation followed by privacy amplification through Davies–Meyer hashing in Advanced Encryption Standard 128-bit (AES-128) compression mode. This engine layer is exposed to external actors through two distinct but behaviourally identical integration surfaces, a design that was adopted deliberately to serve two complementary classes of consumers without duplicating the underlying algorithms.

The first surface is an interactive graphical demonstrator, intended for visual exploration, didactic presentation, and qualitative inspection of the security mechanisms by human users. The second is a stateless representational state transfer (REST) application programming interface (API), formally specified in OpenAPI 3.1 and intended for direct, programmatic, machine-to-machine integration, in particular with Prototype 5. A central integration principle, observed throughout, is that both surfaces invoke precisely the same detector implementations and the same canonical operating points, with

the consequence that a given scene description yields numerically identical results irrespective of the surface through which it is submitted; this equivalence was explicitly verified during validation and is essential to the credibility of the demonstrator, since it guarantees that a partner integrating against the API will obtain results consistent with those observed in the graphical interface and reported in the present deliverable.

The most consequential changes since deliverable D6.2 concern the mode of access and the architecture of the programmatic interface. Whereas the demonstrator described in the earlier deliverable was confined to a single local workstation and was therefore accessible only to an operator physically present at that machine, it has, for the final validation, been migrated to dedicated server infrastructure hosted by the ETIS Laboratory and made reachable at a public, fully qualified domain name secured by transport layer security (TLS). This migration permits any consortium partner to exercise the demonstrator remotely, from an ordinary web browser, without the installation of any client software, dependency, or virtual private network; the interactive desktop is streamed to the browser and the API is reachable over standard secure hypertext transfer protocol (HTTPS). A second substantial change concerns the consolidation and redesign of the programmatic interface. The earlier interface maintained server-side session state, requiring nodes to be registered in one request and a simulation to be triggered in a subsequent request, an arrangement which proved unsuitable for concurrent multi-partner use because the shared state of one consumer could be perturbed by another. To this end, the interface was redesigned as a single, stateless, versioned API rooted at the path /api/v1/, in which each request carries the complete scene description, comprising all node positions and operating-point parameters, and elicits a self-contained response; this eliminates shared session state entirely and renders each call idempotent with respect to its inputs.

The two surfaces are integrated behind a common reverse proxy that constitutes the sole public ingress to the server, terminating TLS and dispatching incoming requests either to the interactive streaming service or to the API service according to the requested path. All functional services are bound exclusively to the host’s loopback interface and are therefore unreachable except through the proxy, an arrangement that confines the externally exposed surface to a single, controlled entry point. Each service is, moreover, supervised so as to restart automatically upon unexpected termination and to resume after a complete system reboot, a property whose resilience was confirmed during validation by deliberately rebooting the host and observing the unattended restoration of all services. Finally, the programmatic API operates from an entirely separate working copy of the codebase, with its own isolated runtime environment and its own references to the measurement data, in deliberate isolation from the interactive graphical session; this isolation ensures that programmatic load generated by an integrating partner cannot disturb the state observed by a partner interacting with the graphical surface, and conversely.

5.1.4.1.4 Inputs and Outputs

In this section, we define the inputs fed into the validation activities and the expected outputs against which results are assessed. The inputs governing each validation run fall into three categories, which together constitute the complete scene description submitted to the engine layer. The first category comprises the spatial configuration of the network, namely the positions, expressed in metres within the array reference frame, of the legitimate user equipment and, as applicable to the scenario under test, of the jammer, the spoofer, and the passive eavesdropper; each such position is constrained to the measurement region and is snapped to the nearest measured grid point, so that every node corresponds to a genuine channel measurement rather than to an interpolated or synthetic one. The second category comprises the selection of active attack scenarios, permitting any combination of jamming, spoofing, and eavesdropping to be enabled independently, so that capabilities may be

exercised both in isolation and in concert; importantly, the detectors operate without privileged knowledge of which attacks are active, and therefore must themselves infer the presence or absence of a threat from the channel observations alone. The third category comprises the operating-point parameters, namely the receive signal-to-noise ratio (SNR) and the jammer transmit power, which are selected by means of three presets aligned between the graphical and programmatic surfaces: the Low, Medium, and High presets correspond respectively to SNR values of 20 dB, 25 dB, and 30 dB, and to jammer transmit powers of 7 dBm, 15 dBm, and 24 dBm.

The expected outputs, against which the collected results are assessed, are likewise organised by capability, each capability producing a structured set of quantities together with the ground-truth references necessary for their evaluation. For the jamming-detection capability, the prototype returns a binary detection verdict, the estimated location of the jammer, the peak value of the GLRT statistic together with the calibrated detection threshold, and the signal-to-interference-plus-noise ratio (SINR) and jammer-to-noise ratio observed at the legitimate user; the verdict is assessed against the known ground-truth jammer position, and the quality of localisation is judged by the proximity of the estimated position to that of the ground truth. For the spoofing-detection capability, the prototype returns a categorical verdict, distinguishing the cases of spoofing detected, spoofing not detected, and an ambiguous outcome; the absolute angular discrepancy between the estimated and the geometric angles of arrival; and aggregate accuracy statistics expressed as the median and mean absolute AoA error evaluated across the grid. These outputs are assessed against the dual expectation that a genuine user, whose claimed and physical positions coincide, is correctly accepted, whilst a spoofer transmitting from a location distinct from that which it claims is correctly rejected; the per-grid statistics additionally characterise the proportion of the measurement region over which a spoofer would be reliably detected. For the secret-key-generation capability, the prototype returns the reconciliation rate achieved between the legitimate parties, the resulting 128-bit key expressed in hexadecimal, and the corresponding bit-agreement attained both before and after the privacy-amplification hash. The expected behaviour against which these are assessed is twofold: the legitimate parties should attain a reconciliation rate approaching unity, signifying near-perfect agreement upon a common key; whilst the eavesdropper, situated beyond the half-wavelength spatial-decorrelation neighbourhood of the legitimate user, should attain a post-hash bit-agreement statistically indistinguishable from that of a random guess, thereby demonstrating that the generated key is not recoverable from a displaced observation of the channel.

5.1.4.2 Validation Outcomes

In this section, we assess the outcomes of the prototype, reporting on the objectives achieved, the results collected, and the limitations identified during execution.Prototype 4 is validated over real measurements drawn from a 64-antenna massive MIMO indoor testbed, so as to demonstrate the applicability of the physical-layer security closed loop under realistic propagation conditions. The validation is designed to highlight the operation of the closed loop under different SNR levels. Also, the three attack capabilities, i.e., jamming, spoofing and eavesdropping, may be exercised either jointly or in isolation, with the attacker and legitimate nodes placed at arbitrary positions on the measured grid. The assessments are organised around the three flows specified in Section 3.4: Flow UC1_2_01 (PHY-layer trustworthiness evaluation), Flow UC1_2_02 (mutual authentication) and Flow UC1_2_03 (fast secret key generation).

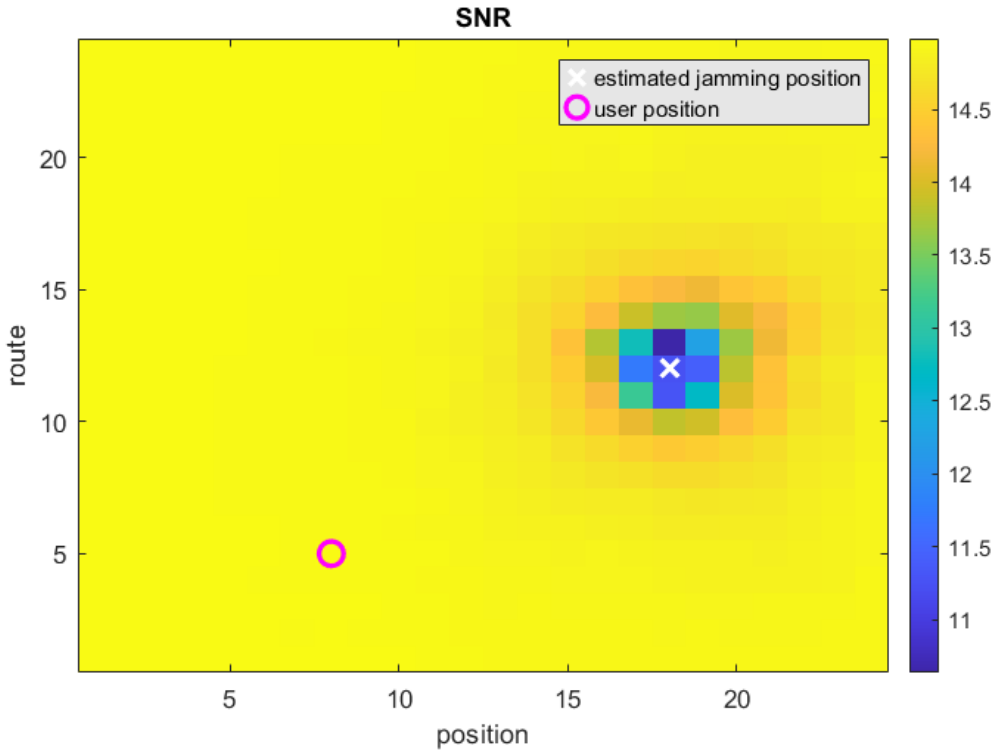

Figure 5.74: Jamming detection and localisation output over the 24 × 24 spatial grid

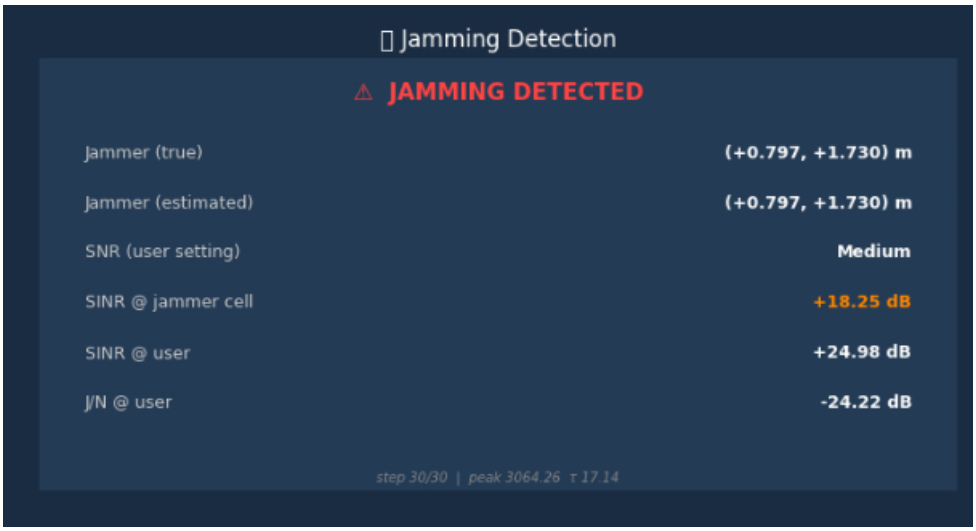

Figure 5.75: Detection decision and estimated jammer position under a medium-SNR setting

Flow UC1_2_01 — PHY-layer trustworthiness evaluation (jamming detection, localisation and mitigation). This flow, realised through CENS01 (PHY monitoring) and CENS03 (trustworthy sensing), both declares the presence of a jammer and recovers its position on the grid. The spatial output, illustrated in Figure 5.74, presents the per-cell SNR map of the monitored region: the localised depression in SNR marks the jammer's spatial footprint, and the estimated jammer position (white cross) is placed at the centre of this depression, well separated from the legitimate user position (magenta circle). The corresponding decision panel is depicted in Figure 5.75, as shown: a) the attack is flagged as detected and the estimated jammer coordinates coincide exactly with the ground-truth coordinates, and, b) beyond the jamming location, the magnitude of the induced degradation is also estimated, i.e., the signal-to-interference-plus-noise ratio (SINR) measured at both jammer and user cell. These two quantities, i.e., the estimated jammer position and the estimated SINR degradation, are precisely the outputs that close the loop: they are forwarded by the activation stage to RAN

control, which, combining the estimated degradation with the path-loss associated with the localised jammer, determines and applies a compensating transmission-power increase that restores the required link quality. The localisation-and-magnitude estimation therefore does not merely characterise the attack but directly drives the physical-layer mitigation, thereby closing the monitoring–analysis–actuation loop.

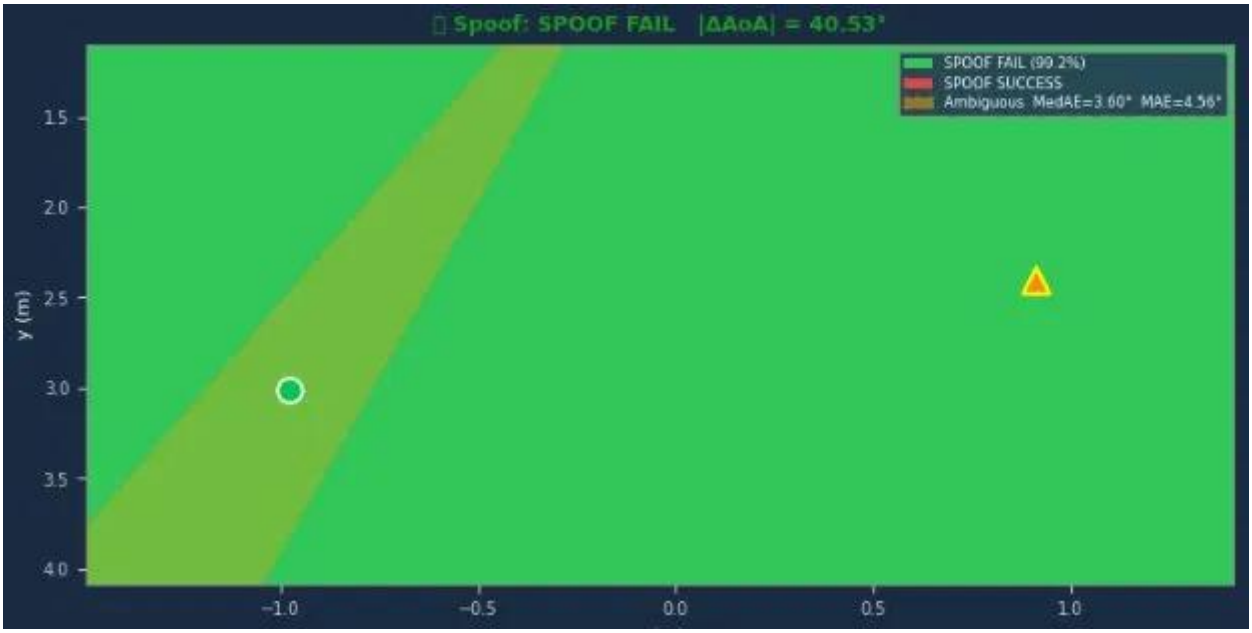

Figure 5.76: AoA-based spoof detection

Flow UC1_2_02 — Mutual authentication. This flow (CENS04, AoA-based authentication) is evaluated on the measured CSI by checking incoming transmissions against the legitimate user's AoA signature. As shown in Figure 5.76, the spoofer is separated from the legitimate user by an AoA of 40.53° and is correctly rejected (SPOOF FAIL), with a correct-rejection rate of 99.2 % and a median/mean absolute AoA error of 3.60°/4.56°. A second key outcome is the explicit identification and delineation of the ambiguous zone (the shaded band) — the angular region in which the spoofer shares the legitimate user's AoA and is therefore indistinguishable from it, so the attack cannot be detected. This constitutes the principal limitation of the authentication capability.

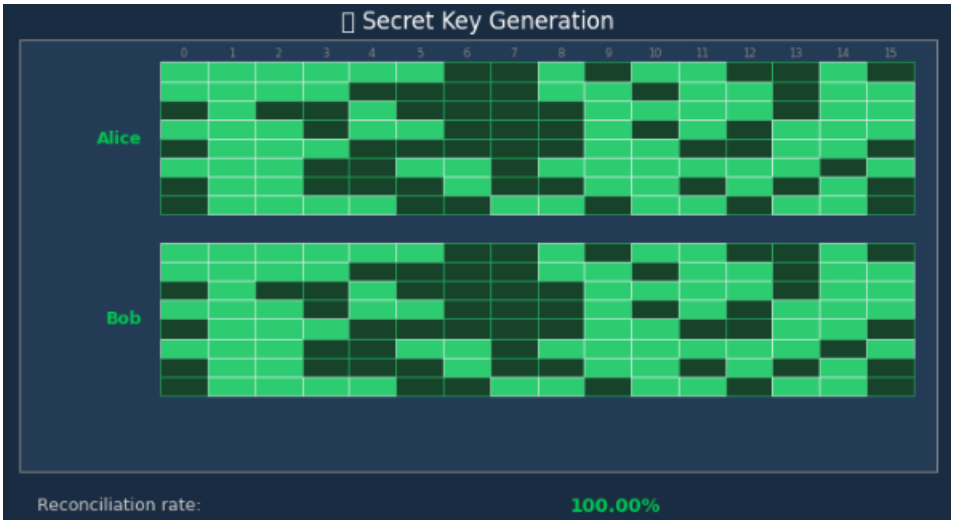

Figure 5.77: Secret key generation output

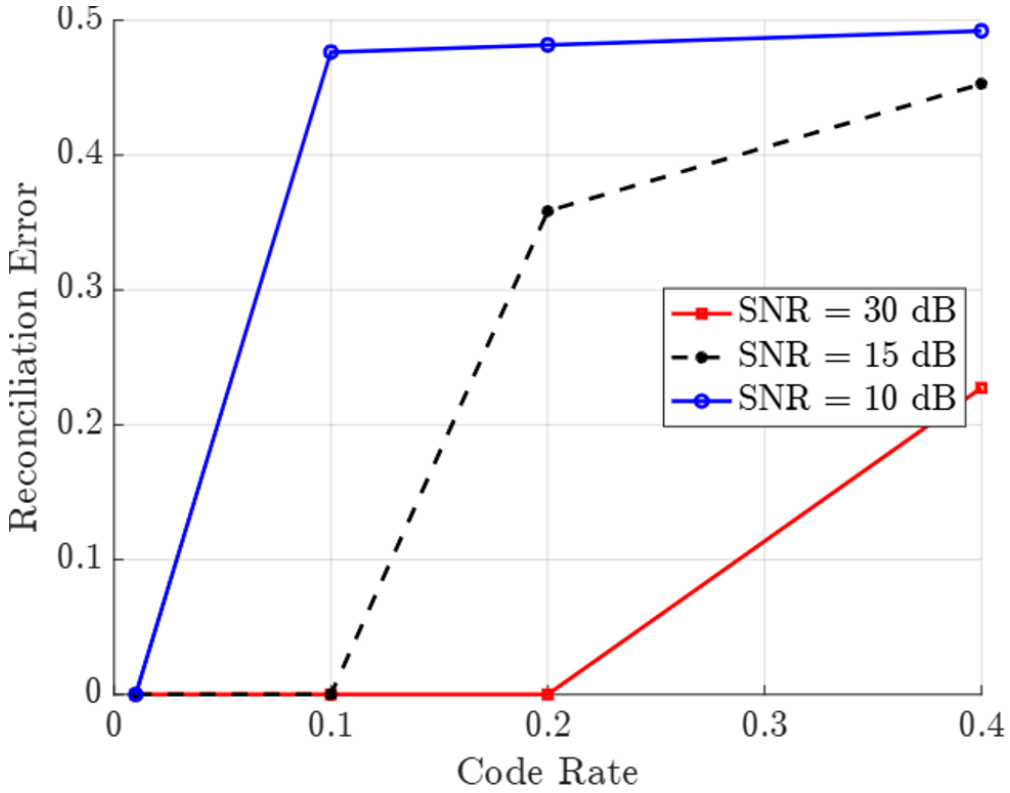

Figure 5.78: reconciliation error versus reconciliation code rate

Flow UC1_2_03 — (Fast) secret key generation. This flow (CENS05) generates a shared symmetric key between the legitimate user (Alice) and the 64-element ULA base station (Bob). Figure 5.77 shows the reconciled key material at the two parties to be bit-for-bit identical, at a 100% reconciliation rate, sustained across all tested geometries and operating points. Figure 5.78 characterises the operating envelope, reporting the reconciliation error against the channel-code rate for receive-SNR levels of 10, 15 and 30 dB: the error remains negligible up to a code rate of about 0.2 at 30 dB and departs from zero progressively earlier as the SNR decreases (already beyond 0.1 at 10 dB), confirming that the demonstrator operates within the error-free SNR–code-rate regime. Secrecy against a displaced eavesdropper is enforced by the privacy-amplification stage rather than assumed from channel geometry, and is stress-tested in the dedicated WP5 benchmark (D5.3, Challenge 3), which evaluates key recovery when the eavesdropper is located as close as one wavelength from the legitimate node — the regime in which half-wavelength decorrelation can no longer be taken for granted indoors.

Overall, the three security objectives pursued by Prototype 4, i.e., jamming detection with localisation and mitigation, AoA-based authentication, and physical-layer key agreement, were achieved on real measured CSI, the authentication objective being bounded by the ambiguous AoA zone, whose extent the evaluation explicitly characterises. These outcomes are assessed against the quantitative use-case KPI targets in Section 5.2. Finally, a deeper, quantitative evaluation of these mechanisms has been carried out and demonstrated within WP5: the AoA-based authentication and the secret key generation are treated in detail in Deliverable D5.3 as Challenge 2 and Challenge 3 respectively, which formulate and assess the corresponding open research problems beyond the scope of the present demonstrator.

5.1.5 Prototype 5: Master Prototype

The validation setup for Prototype 5 (Master Prototype) is explicitly designed to demonstrate the cross-layer integration capabilities of the ROBUST-6G architecture. Unlike individual prototypes that focus on specific domains, this setup acts as the overarching integration hub, combining the functionalities of Prototypes 1, 2, 3, and 4 into a unified, interoperable security ecosystem across a federated testbed infrastructure. While this section reports results of the integration between Prototype 1, 2, and 4, the integration between prototype 2 and 3 is completely reported in Prototype 3 where the exposure of the ZTSP functionalities through NetSecaaS is detailed.

5.1.5.1 Testbed Configuration

The integrated testbed for the Master Prototype comprises the following core components operating in concert:

- NetSecaaS Gateway (Prototype 3): Hosted in TID testbed, serves as the northbound Exposure Framework to translate third-party requests into actionable Security Service Level Agreements (SSLAs).
- Zero-Touch Security Platform (ZTSP - Prototype 2): Hosted in NXW testbed, acts as the central orchestration engine (ZTSO, S-RO, S-CL Manager) responsible for semantic reasoning, service composition, and lifecycle management.
- Global Model Repository (GMR - Prototype 1): Hosted in UMU testbed, it provides the trustworthy, centralised registry from which AI/ML threat detection algorithms are retrieved for use in the ZTSP.
- Physical Layer Security Closed Loop (Prototype 4): Hosted in the hardware lab at ENSEA, it represents the physical-layer execution environment and hardware-level security mechanisms (operating on real measured CSI from the massive-MIMO testbed) that are mapped semantically into the orchestrator for automated actuation

5.1.5.1.1 Prototype 2 and Prototype 4 Integration

The PHY-layer integration workflow validates how the physical-layer security capabilities delivered by Prototype 4, hosted in the ENSEA/ETIS hardware laboratory, are made available to the Zero-Touch Security Platform as orchestrable security assets. Differently from the AI/ML workflow, where the models are software artefacts dynamically retrieved from the GMR, the PHY capabilities are produced by a radio-hardware demonstrator operating on a measured-CSI dataset (a 24×24 spatial grid acquired with a 64-element uniform linear array at 2.61 GHz). The demonstrator exposes three independent detection and key-agreement engines through a REST API (live at https://robust6g-demo.etis-lab.fr/api/v1/docs), summarised in Table 5.1. The integration therefore does not consist of deploying a new function, but of semantically describing these existing capabilities and binding them to their physical environment, so that the ZTSO can reason about them and orchestrate a closed loop around them.

Table 5.1: PHY Demonstrator capabilities and accessible methods

Capability	Method	API Endpoint
Jamming detection	Spatial GLRT + temporal WL-CUSUM	POST /jamming/detect
Spoofing detection	Root-MUSIC Angle-of-Arrival + jammer-mitigation calibration	POST /spoofing/detect
Secret-key generation	Polar-CRC reconciliation + Davies-Meyer/AES-128	POST /skg/generate

As a first step, the demonstrator and its capabilities are registered in the ZTSO Knowledge Graph through the ontology-manager API (POST /kg/entity, POST /kg/relationship, POST /kg/property). As depicted in Figure 5.79, the demonstrator is modelled as a *SecurityApplication*

(PHYLayerDemonstrator) exposing the three functional capabilities reported in Prototype 4. These capabilities are wired to the activities they enable: *JammingDetection*, *SpoofingDetection* and *KeyGeneration*, each measured by its own individual metrics (SINR and jamming peak-score, ΔAoA, reconciliation rate and key-match) and linked to the PHY attack techniques it counters, namely *PHYLayerJamming* and *SignalReplicaSpoofing*, which in turn raise the corresponding jamming and spoofing alerts.

Figure 5.79: PHY Demonstrator modelling in the KG

The physical environment is then registered in the ZTSO Security Catalogue through the catalog-manager API and linked to the demonstrator infrastructure. As shown in Figure 5.80, the ENSEA Lab is catalogued as a target environment of physical type, holding the demonstrator API endpoint as its access point. Modelling it as a physical environment, rather than as a cloud-native Kubernetes or Docker tier, reflects the fact that the underlying mechanisms run on real radio hardware and cannot be instantiated on demand by the orchestrator.

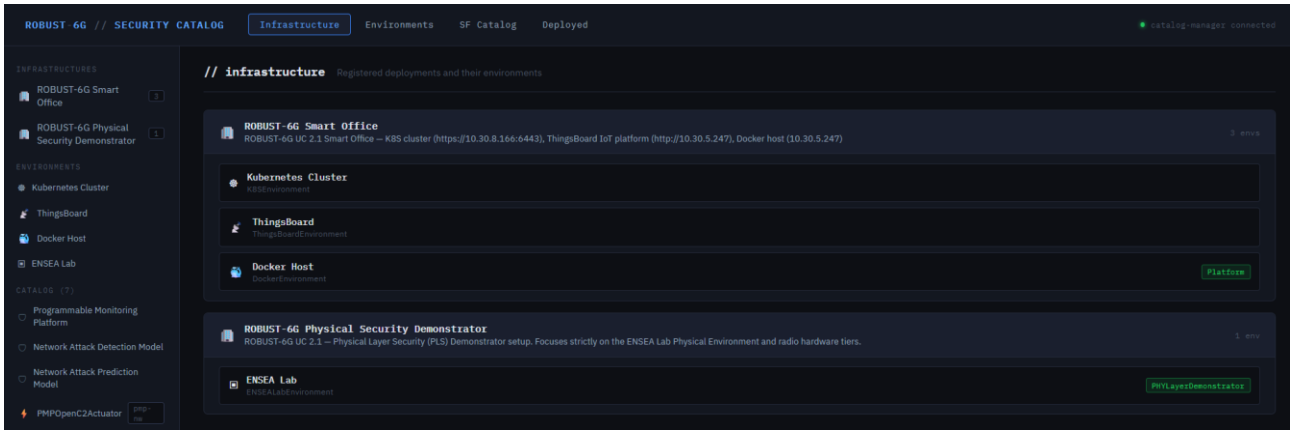

Figure 5.80: PHY Demonstrator Environment in the Catalogue

Finally, the demonstrator itself is onboarded as a Security Function bound to that environment, again via the catalog-manager and registered as already deployed on the ENSEA Lab. As shown in Figure 5.81, the PHYLayerDemonstrator Security Application is catalogued together with its OpenAPI manual descriptor marked as suitable for, and resident in, the ENSEA Lab physical environment. Unlike the AI models, which are onboarded as deployable S-RO artefacts, the PHY demonstrator is

catalogued as an existing, hardware-resident capability provider: the ZTSP consumes it in place rather than orchestrating its deployment. A central design choice of the PHY integration is that the demonstrator is not driven directly by the closed-loop stages, but through a purpose-built OpenC2 actuator. On top of the demonstrator's three REST endpoints, an actuator strategy (PhyDemonstratorStrategy) is implemented that exposes the underlying engines as a small set of standardised OpenC2 commands, expressed as *action* / *target-resource* pairs: detect / jamming (GLRT-based jamming detection), detect / spoofing (Angle-of-Arrival spoofing detection), and start / skg (secret-key generation, i.e. key rotation). When an OpenC2 command addressed to the phy target service is received, the actuator's dispatcher validates the target, maps the (action, resource) pair to the corresponding engine, builds the appliance-specific request body from the generic command arguments (the user/jammer/spoofer/eavesdropper positions and the operating point), applies the appropriate call semantics and normalises the heterogeneous responses (jamming alarm, peak score and user SINR; spoofing verdict and ΔAoA; SKG reconciliation percentage and Alice–Bob key match) into a uniform OpenC2 result of status code, status text and structured data. The significance of this is that it decouples the orchestration logic from the appliance: because every demonstrator-specific detail (the exact endpoint paths, the request payloads, the timeouts, the response field names) is confined to the actuator, the four stages of the closed loop are not tailored one-to-one to the PHY demonstrator and its APIs. They are general, reusable closed-loop functions that emit abstract OpenC2 commands (e.g. start / skg) and consume normalised results, with no hard-coded knowledge of the demonstrator's REST contract; evolving the underlying PHY appliance, or replacing it with a different physical-layer security provider, therefore requires changing only the actuator strategy, while the loop, its descriptors and its orchestration remain untouched. This is the PHY Orchestration using OpenC2 approach: the same standardised, profile-based actuation model already applied to the IoT tier in Use Case 2 (ThingsBoard OpenC2 actuator), now extended down to the physical and radio layer.

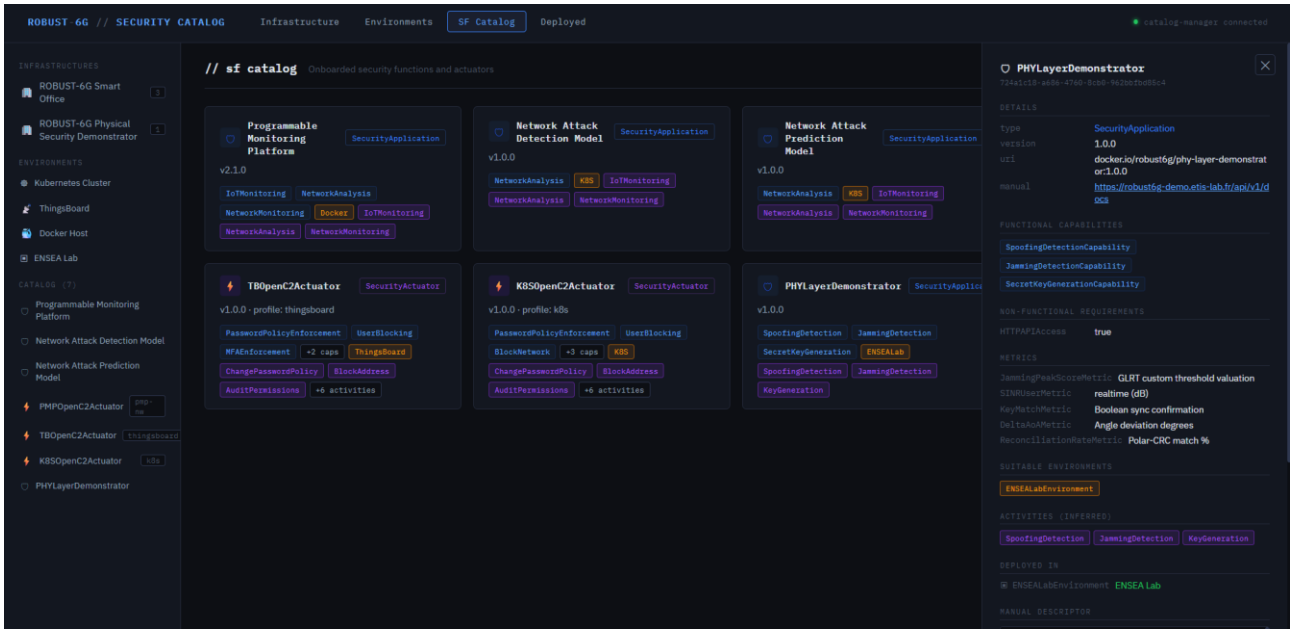

Figure 5.81 - PHY Demonstrator Security Functions in the Catalogue

With the demonstrator described in the Knowledge Graph (Figure 5.79), catalogued against its physical environment (Figure 5.80 and Figure 5.81), and fronted by the OpenC2 PHY actuator loaded in the Security Functions Catalogue (Figure 5.81), the orchestration of a physical-layer security loop becomes possible without any appliance-specific coupling. The four stages of the *PHY Jamming &*

Spoofing Mitigation Closed Loop (monitoring, analysis, decision and execution) are onboarded as orchestrable templates on the Security Resource Orchestrator and, at runtime, drive the demonstrator through OpenC2 commands dispatched over the shared MQTT broker: the loop monitors for jamming and spoofing, confirms the alarm, and elects a RAN/network mitigation together with a secret-key rotation, while the OpenC2 PHY actuator translates the abstract detect/jamming, detect/spoofing and start/skg commands into the concrete demonstrator calls and returns normalised verdicts to the loop. At the end of this workflow the physical-layer security capabilities are fully integrated into the ZTSP, semantically described, catalogued against their hardware environment, and actuated through a standardised OpenC2 profile rather than a bespoke interface, and are therefore available, alongside the GMR-trained AI models, to the Zero-Touch Security Orchestrator for SSLA-driven security service composition and the automated deployment of PHY-layer Security Closed Loops. As shown in

Figure 5.82, the *PHY Jamming & Spoofing Mitigation Closed Loop* descriptor is composed on the S-CL Manager from its four stage descriptors (monitoring, analysis, decision and execution), declaring the loop's goal: confirm a jamming or spoofing alarm and remediate it via RAN/network mitigation together with a secret-key rotation.

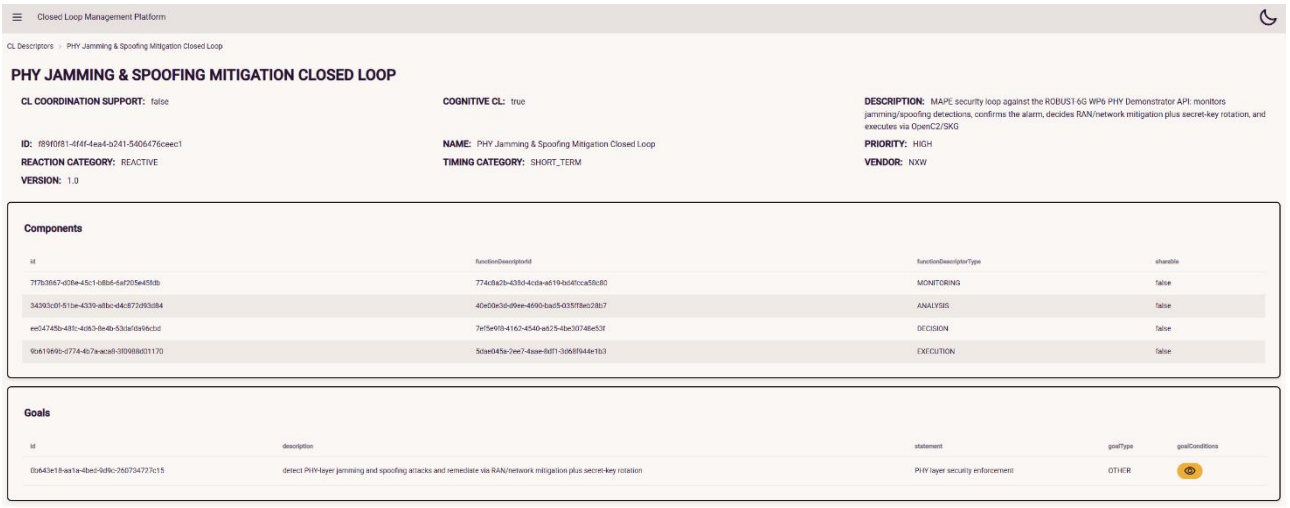

Figure 5.82: PHY S-CL Descriptor

Figure 5.83 shows the resulting PHY-layer security service, in which the S-CL is bound together with the PHY OpenC2 actuator (and its supporting MQTT fabric) into a single deployable security service. Upon submission, the service is instantiated by the lifecycle manager: Figure 5.84 reports the PHY security service reaching the INSTANTIATED state.

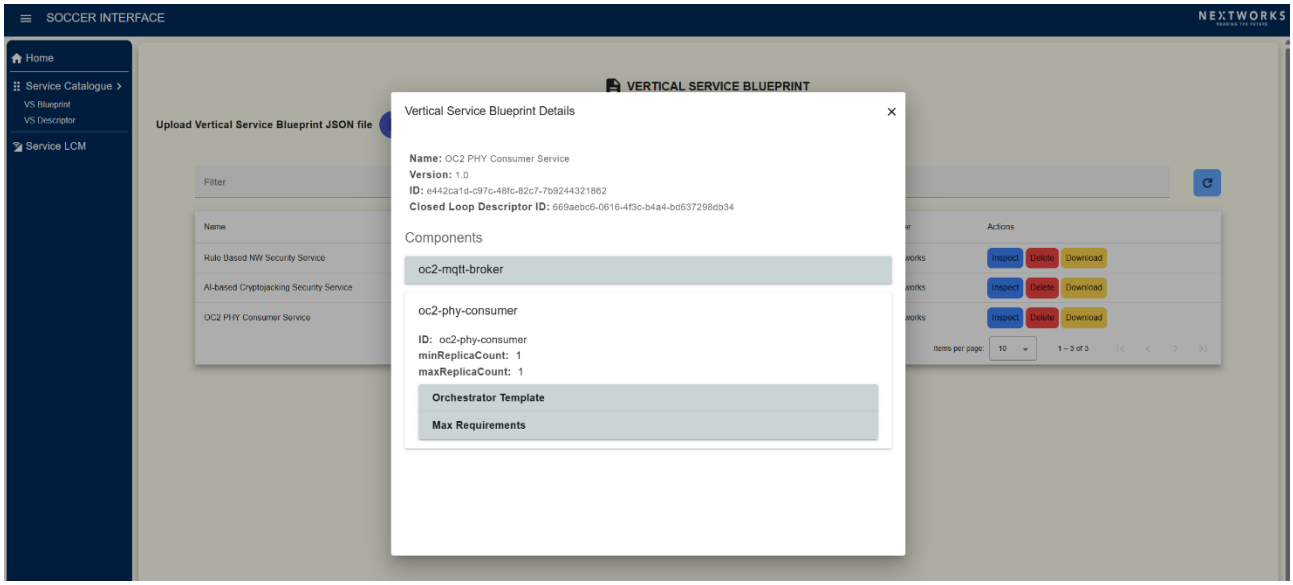

Figure 5.83: PHY Layer Security Service – PHY S-CL and PHY OpenC2 Acutator

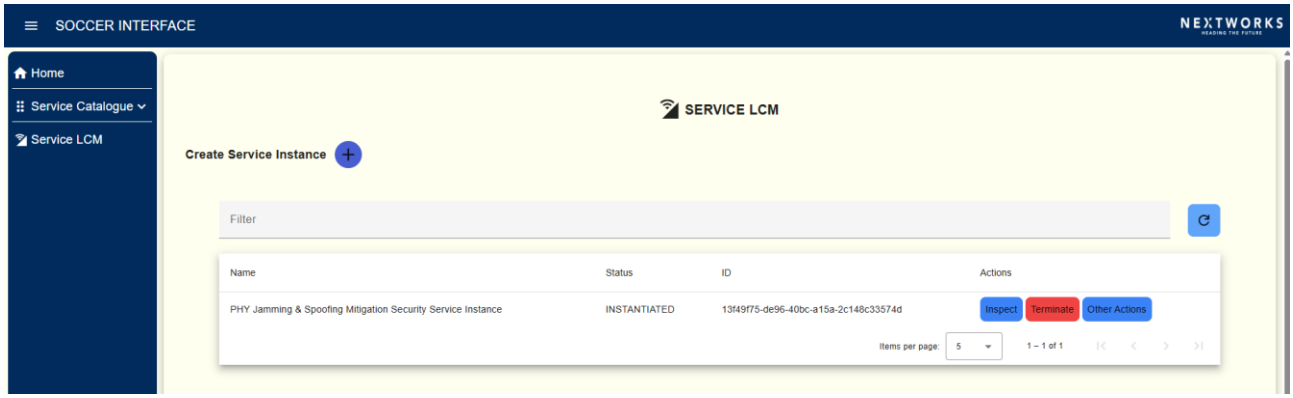

Figure 5.84: SHY SSe Instantiated

Figure 5.85 shows the instantiated service materialised on the target infrastructure, where the four S-CL stage pods are deployed alongside the OpenC2 PHY consumer and the shared MQTT broker, the security functions and closed-loop stages that together realise the complete Security Service.

```
ubuntu@smart-office:~$ kubectl -n security-functions get pods
NAME                                                READY   STATUS    RESTARTS   AGE
mqtt-fabric-master-2094438380-59645f796d-pc2ss      1/1     Running   0          66s
oc2-phy-consumer-0828155766-phy-actuator-85d4d7558-t8hrj   1/1     Running   0          52s
ubuntu@smart-office:~$ kubectl -n closed-loop-functions get pods
NAME                                  READY   STATUS    RESTARTS   AGE
analysis-scl-phy-6749bf5bd6-nzmpn     1/1     Running   0          39s
decision-scl-phy-967578cf-9zgs4       1/1     Running   0          19s
execution-scl-phy-58b6d4b95b-92whl    1/1     Running   0          30s
monitoring-scl-phy-6b9f7d757d-2mdg5   1/1     Running   0          50s
```

Figure 5.85: SFs and S-CL Stages deployed

The four remaining figures capture the loop executing against the live demonstrator through the dedicated OpenC2 consumer. Figure 5.86 shows the monitoring stage logs, where the stage periodically issues the detect/jamming and detect/spoofing OpenC2 commands and forwards the raw detection results (alarm, peak score and SINR; verdict and ΔAoA) to the analysis stage. Figure 5.87 shows the analysis stage confirming the alarm by testing those detections against its configured thresholds (the jamming peak-score ratio and the ΔAoA floor) and emitting a consolidated jam/spoof verdict. Figure 5.88 shows the decision stage electing the remediation: a RAN/network mitigation together with a secret-key rotation carrying the user and eavesdropper positions required for the

rekeying. Finally, Figure 5.89 shows the execution stage enacting that decision through the OpenC2 PHY actuator: as depicted in

Figure 5.90, the OpenC2 actuator dispatches the mitigation actions and the start/skg command, and confirms the outcome, namely a secret-key rotation committed once the Polar-CRC reconciliation exceeds the required quality and the Alice–Bob keys match.

```
[INFO] - 2026-06-19 07:32:43,437 - OpenC2 MQTT connected (function.py:100)
[INFO] - 2026-06-19 07:32:43,440 - MQTT Broker connected (function.py:229)
[INFO] - 2026-06-19 07:32:52,890 - Waiting 30s for the Analysis stage to come online... (function.py:181)
[INFO] - 2026-06-19 07:32:52,891 - Monitoring Loop STARTED (function.py:241)
[INFO] - 2026-06-19 07:33:22,892 - Sending OpenC2 detect commands... (function.py:184)
[INFO] - 2026-06-19 07:33:22,893 - Sending 'detect/jamming' to openc2-phy-consumer (function.py:112)
[INFO] - 2026-06-19 07:33:23,069 - Sending 'detect/spoofing' to openc2-phy-consumer (function.py:112)
[INFO] - 2026-06-19 07:33:23,523 - jamming  : alarm=True, peak=2874.88, SINR_user=+24.85 dB, confidence=Very High (function.py:147)
[INFO] - 2026-06-19 07:33:23,523 - spoofing : verdict=SPOOF_FAIL, dAoA=23.29° (clean MedAE 3.60°) (function.py:152)
[INFO] - 2026-06-19 07:33:23,524 - EVENT: monitoring detected jamming, spoofing, passed now to the analysis stage (function.py:168)
```

Figure 5.86: S-CL Monitoring Stage Logs

```
[INFO] - 2026-06-19 07:32:54,413 - MQTT Broker connected (function.py:125)
[INFO] - 2026-06-19 07:33:01,975 - Analysis Function STARTED. Listening on scl_monitoring_dst (function.py:183)
[INFO] - 2026-06-19 07:33:23,532 - Received monitoring report; analysing jamming/spoofing alarms (function.py:140)
[INFO] - 2026-06-19 07:33:23,533 - GLRT estimate at (-0.287, +2.894); peak 2874.88 vs threshold 17.14 (function.py:73)
[INFO] - 2026-06-19 07:33:23,533 - [analyse:jam] confirmed=True (function.py:83)
[INFO] - 2026-06-19 07:33:23,533 - [analyse:spoof] dAoA=23.29° vs clean MedAE 3.60° -> confirmed=True (function.py:92)
[INFO] - 2026-06-19 07:33:23,534 - EVENT: confirmed jamming, spoofing, pass it to the decision stage (function.py:106)
[INFO] - 2026-06-19 07:33:23,537 - Published analysis to scl_analysis_dst: {'user_xy': [-0.7, 3.6], 'jam_confirmed': True, 'spoof_confirmed': True} (function.py:161)
[INFO] - 2026-06-19 07:33:55,812 - Received monitoring report; analysing jamming/spoofing alarms (function.py:140)
```

Figure 5.87: S-CL Analysis Stage Logs

```
[INFO] - 2026-06-19 07:33:12,619 - MQTT Broker connected (function.py:102)
[INFO] - 2026-06-19 07:33:20,135 - Decision Function STARTED. Listening on scl_analysis_dst (function.py:154)
[INFO] - 2026-06-19 07:33:23,538 - Received Analysis verdict: jam_confirmed=True, spoof_confirmed=True (function.py:116)
[INFO] - 2026-06-19 07:33:23,539 - EVENT: planned mitigation actions [notify_RAN(handover_or_beamsteer), drop_session(suspected_spoofer)], passing them to the executi
on stage (function.py:83)
[INFO] - 2026-06-19 07:33:23,543 - Forwarded decision to Execution: {'decision': 'EXECUTE', 'actions': ['notify_RAN(handover_or_beamsteer)', 'drop_session(suspected_s
poofer)'], 'rotate_key': True, 'user_xy': [-0.7, 3.6]} (function.py:132)
```

Figure 5.88: S-CL Decision Stage Logs

```
[INFO] - 2026-06-19 07:33:03,811 - OpenC2 MQTT connected (function.py:94)
[INFO] - 2026-06-19 07:33:03,816 - MQTT Broker connected (function.py:172)
[INFO] - 2026-06-19 07:33:11,054 - Execution Function STARTED. Listening on scl_decision_dst (function.py:211)
[INFO] - 2026-06-19 07:33:23,544 - Applying mitigation actions: notify_RAN(handover_or_beamsteer), drop_session(suspected_spoofer) (function.py:140)
[INFO] - 2026-06-19 07:33:23,544 - EVENT: applied mitigation actions [notify_RAN(handover_or_beamsteer), drop_session(suspected_spoofer)] (function.py:127)
[INFO] - 2026-06-19 07:33:23,547 - Rotating secret key via OpenC2 (please wait, cold call may take 25-40s)... (function.py:145)
[INFO] - 2026-06-19 07:33:23,547 - Sending 'start/skg' to openc2-phy-consumer (function.py:106)
[INFO] - 2026-06-19 07:33:23,548 - reconciliation_pct=0.00%, alice_bob_match=False (function.py:158)
[INFO] - 2026-06-19 07:33:23,548 - EVENT: key rotation result below threshold (reconciliation_pct=0.00%, alice_bob_match=False); deferring commit (function.py:127)
```

Figure 5.89: S-CL Execution Stage Logs

```
2026-06-19 07:32:40,980 - OpenC2Consumer - INFO - Connecting to openc2-mqtt-broker:1883...
2026-06-19 07:32:40,985 - OpenC2Consumer - INFO - Connected. Subscribing to CMD topic: oc2/cmd/device/phy-consumer
2026-06-19 07:33:22,895 - OpenC2Consumer - INFO - Received message on oc2/cmd/device/phy-consumer
2026-06-19 07:33:22,896 - PhyStrategy - INFO - OpenC2 command received: action='detect' resource='jamming'
2026-06-19 07:33:23,067 - OpenC2Consumer - INFO - Published JSON response to oc2/rsp
2026-06-19 07:33:23,070 - OpenC2Consumer - INFO - Received message on oc2/cmd/device/phy-consumer
2026-06-19 07:33:23,071 - PhyStrategy - INFO - OpenC2 command received: action='detect' resource='spoofing'
2026-06-19 07:33:23,521 - OpenC2Consumer - INFO - Published JSON response to oc2/rsp
2026-06-19 07:33:23,549 - OpenC2Consumer - INFO - Received message on oc2/cmd/device/phy-consumer
2026-06-19 07:33:23,549 - PhyStrategy - INFO - OpenC2 command received: action='start' resource='skg'
2026-06-19 07:33:55,653 - OpenC2Consumer - INFO - Published JSON response to oc2/rsp
```

Figure 5.90: OpenC2 Phyisical Layer consumer logs

5.1.5.1.2 Prototype 2 and Prototype 1 Integration

The AI/ML integration workflow validates the end-to-end path through which a decentralised-federated-learning (DFL) threat-detection model produced in Use Case 1 Scenario 1 is retrieved from the Global Model Repository (GMR, Prototype 1, hosted at UMU) and made available to the Zero-Touch Security Platform as an orchestrable Security Function. As a first step, the administrator interacts with the GMR via its REST API, depicted in Figure 5.91, to navigate the trained models without any prior knowledge of their internal identifiers. The administrator first lists the registered training runs (GET /api/robust/scenarios) to locate the DFL scenario trained on the TON-IoT dataset, and then browses the produced checkpoints in a metric-aware fashion (GET /api/robust/models?metric=Test/Accuracy), which ranks the per-participant models by their measured test accuracy instead of by file name. From this ranking, the administrator selects the best three models, one per federation node, and downloads their artefacts together with the associated metrics (GET /api/robust/models?scenario_name=…&participant_id=… and GET /api/robust/metrics/{scenario_name}/). The measured performance that drives this selection is reported in Table 5.2.

Table 5.2: Best three DFL TON-IoT threat-detection models retrieved from the GMR

Model (node)	Accuracy	Precision	Recall	F1-Score
P1	93.72%	93.94%	93.72%	93.69%
P0	93.25%	94.46%	93.25%	93.72%
P3	93.17%	93.50%	93.17%	93.06%

Once the models are retrieved, their security semantics are registered in the ZTSO Knowledge Graph through the ontology-manager API (POST /kg/entity, POST /kg/relationship, POST /kg/property). As depicted in Figure 5.92, the DFL TON-IoT model is modelled as a *SecurityApplication* exposing a *NetworkThreatDetectionCapability* (and the shared *NetworkAnalysisCapability*), wired to the *NetworkThreatDetection* activity that it is suited for, to the abstract performance metric types it is measured by (accuracy, precision, recall, F1-score), and to the set of TON-IoT attack techniques it covers, each technique, in turn, raising the corresponding AI threat-detection alert. This semantic description is what later allows the ZTSO reasoner to infer the model as a candidate Security Function whenever an activity requiring network threat detection is requested.

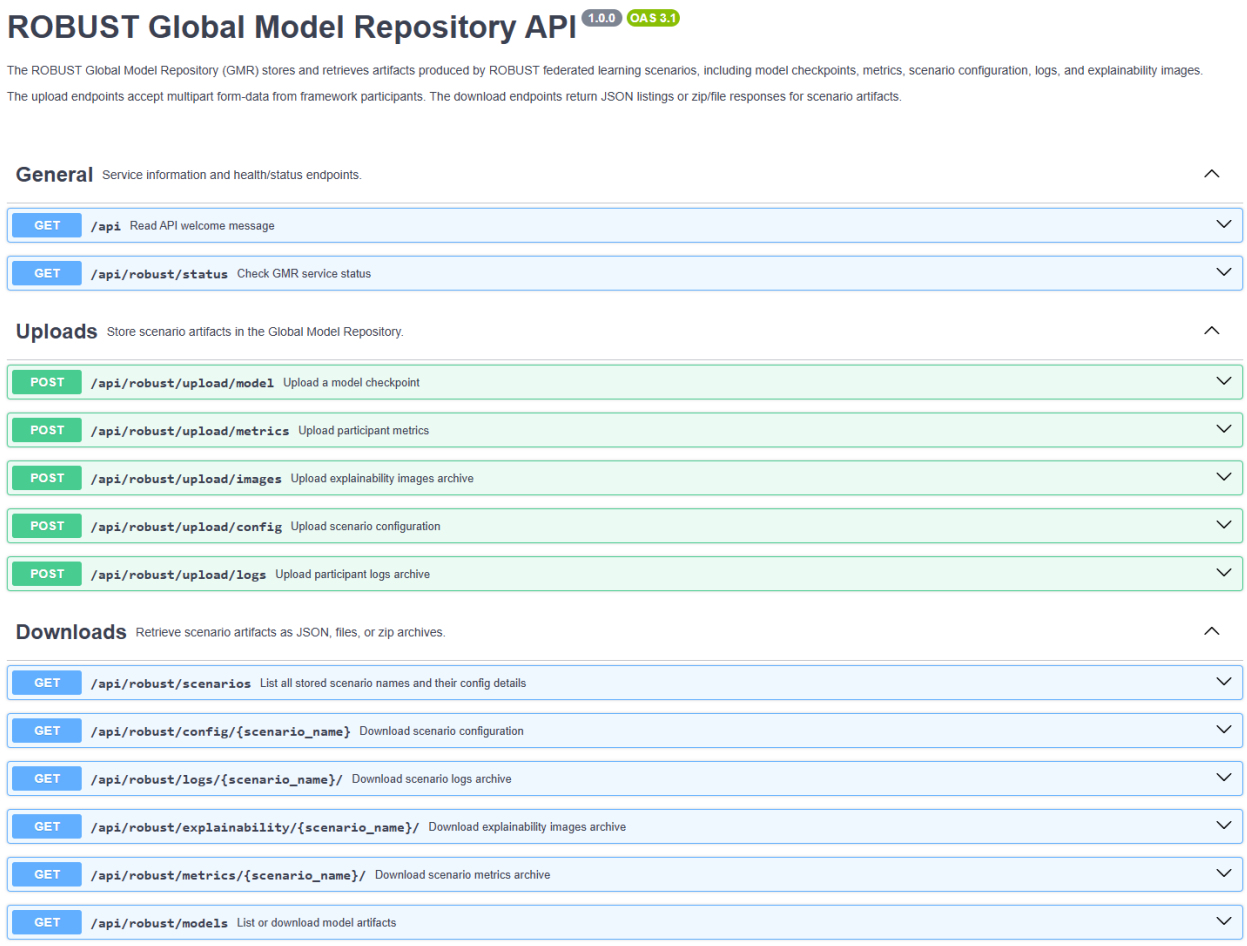

Figure 5.91: GMR OpenAPI Specification

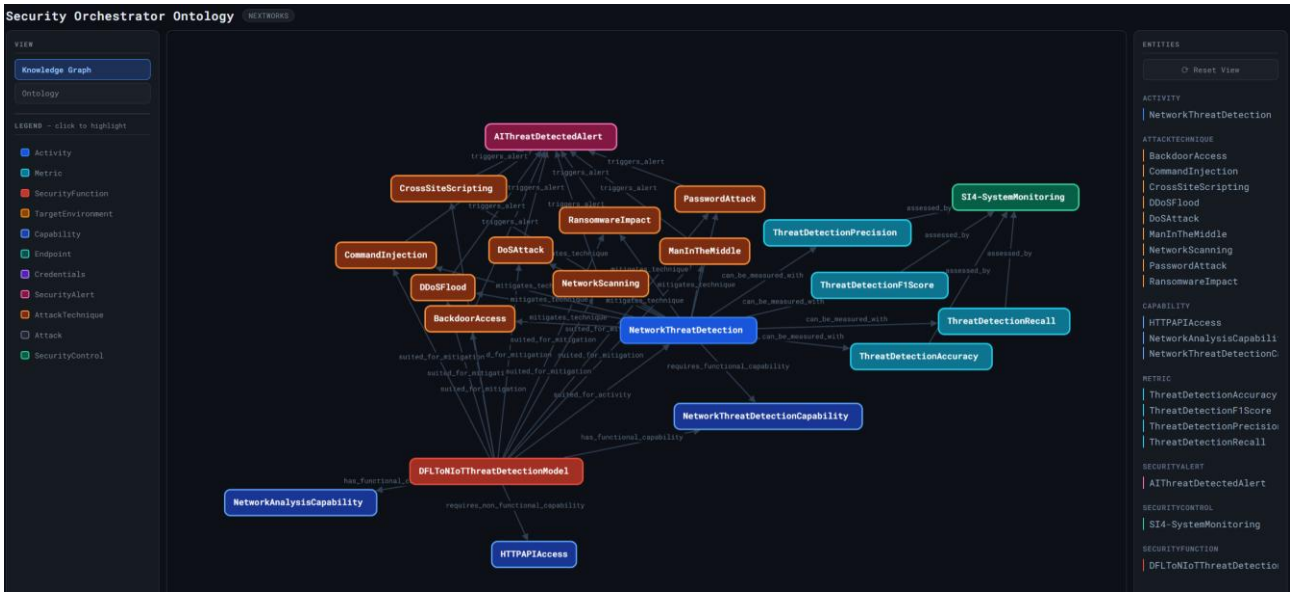

Figure 5.92: DFL AI Algorithms modelling in the KG

In parallel, each selected model is turned into a deployable artefact and onboarded on the Security Resource Orchestrator (S-RO). At a high level, the model checkpoint is wrapped in a BentoML inference service, exposing a /predict endpoint that consumes the 42 TON-IoT flow features and returns the ten-class verdict, containerised into an image, and packaged as a Helm chart. The chart is then uploaded to the S-RO service-template catalogue (POST /api/v1/service-template-catalogue/templates), which registers it as an orchestrable template and returns the artefact reference used by the subsequent steps. Figure 5.93 shows the resulting DFL algorithm artefact onboarded and available in the S-RO.

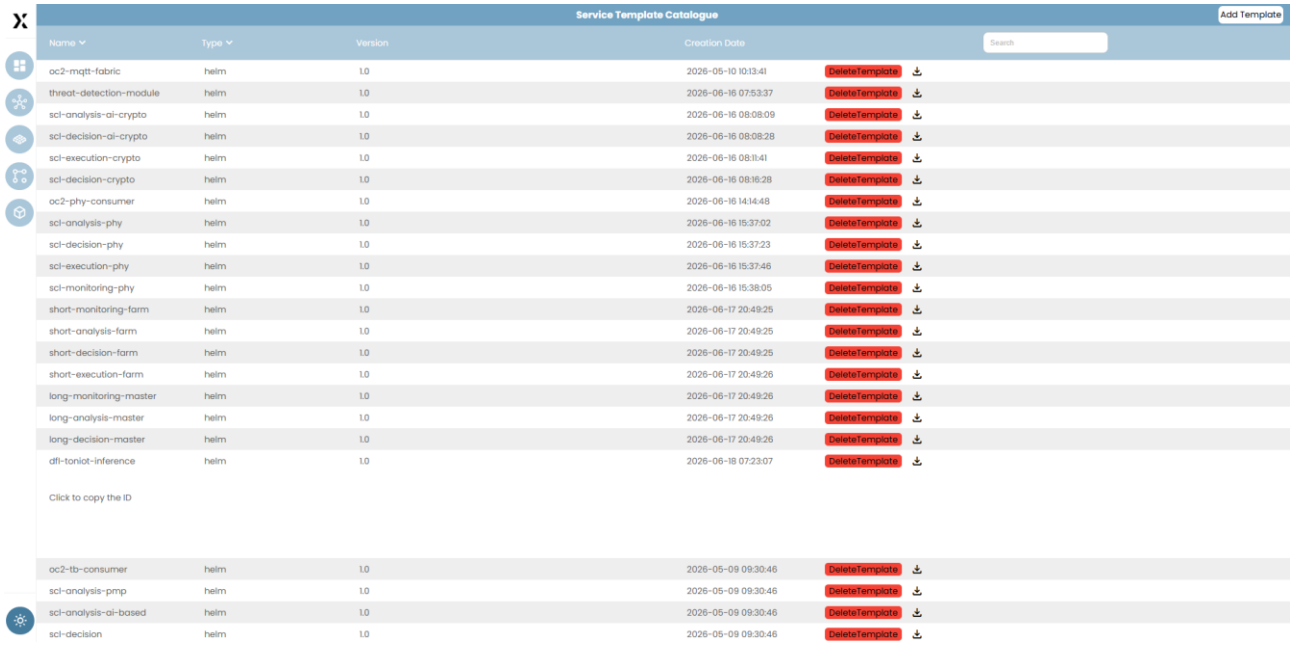

Figure 5.93: DFL Algorithm artefact in the S-RO

Finally, with the deployable artefact in place, the administrator loads the three models into the ZTSO Security Catalogue through the catalog-manager API (POST /catalog/security-function). As shown in Figure 5.94, each model is registered as a distinct Security Function named after its originating node, carrying its functional capabilities and its measured metrics from Table 5.2, and bound to the S-RO template uploaded in the previous step via the sroTemplateName/sroTemplateVersion

reference. This binding is what closes the loop between the semantic description held in the Knowledge Graph, the measurable performance recorded in the catalogue, and the concrete, deployable artefact held in the S-RO.

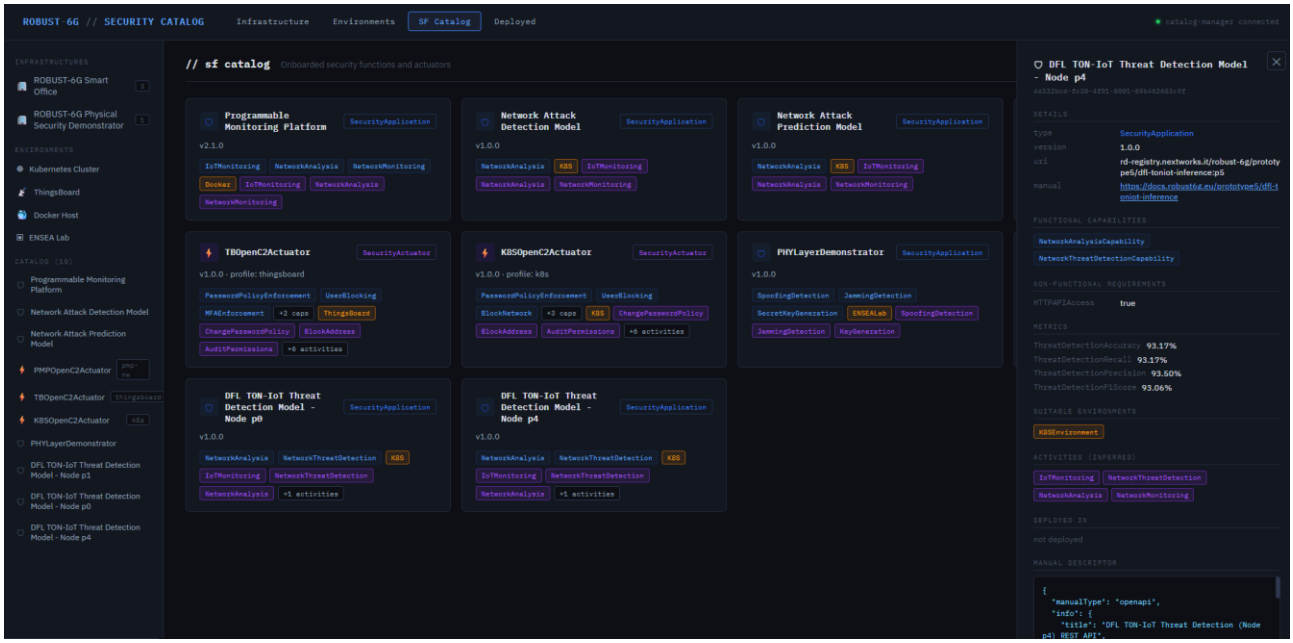

Figure 5.94: DFL AI Algorithms in the catalogue - Best 3

At the end of this workflow the three GMR-trained models are fully integrated into the ZTSP: semantically described in the Knowledge Graph, quantitatively characterised and catalogued in the ZTSO Security Catalogue, and backed by orchestrable artefacts in the S-RO. They are therefore immediately available to the Zero-Touch Security Orchestrator for SSLA-driven security service composition and for the automated deployment of AI-driven Security Closed Loops.

5.1.5.1.3 Prototype 2 and Prototype 3 Integration

The third workflow of the Master Prototype validation demonstrates the integration between Prototype 2 (Multi-Layer Zero-Touch Defender) and Prototype 3 (NetSecaaS Gateway), establishing an end-to-end path through which a third-party consumer can request a complex, orchestrated security service without directly composing a raw SSLA artefact. In Prototype 2, security services are triggered through SSLAs ingested, validated, and parsed by the ZTSO Policy Manager before driving Security Function selection, IRP generation, and Security Closed Loop deployment. Writing a well-formed SSLA, however, requires detailed knowledge of the ZTSO Ontology, the available Security Functions, the target-environment identifiers, and the applicable security-activity vocabulary, a level of expertise that cannot reasonably be expected from a third-party consumer or vertical application developer. Prototype 3 removes this barrier by exposing a parameterized REST endpoint (Figure 5.95 and Figure 5.96) that accepts a minimal, high-level description of the security intent (the target resource or environment, the desired protection activity, and optionally constraints such as performance bounds or applicable NIST SP 800-53 Rev. 5 control families) and transforms it into a complete, schema-valid SSLA. What makes this transformation possible is the information that the Ontology Manager of the ZTSO publishes about the security model, and the way the exposure layer consumes it. The Ontology Manager maintains the ROBUST-6G security ontology and exposes it over REST through two complementary surfaces already introduced in D4.4 [R6G26-D44]: the TBox Management API, which manages the ontology schema (the classes and properties describing Security Functions, capabilities, activities, metrics, controls and target environments), and the ABox

/ Knowledge-Graph Management API (the /kg/entity, /kg/relationship, /kg/property and graph-query operations) through which the concrete security individuals and their relationships are asserted and retrieved, with the reasoner inferring, for each activity, the Security Functions suited to perform it. For the exposure integration this surface was extended with a dedicated activities query endpoint (/activities) that returns, in a single consumer-oriented document, every security Activity together with the FunctionalCapabilities it requires and the Metrics by which it is measured, each metric now enriched with its type and admissible range (operator and operand bounds) and anchored to the corresponding NIST control and control-enhancement. This is the missing piece that lets an external party know not only which activities and metrics exist, but also what values are admissible for them when expressing a service-level objective. On the NetSecaaS side, the Transformation Function consumes this /activities output and bridges it to a standardised control vocabulary by means of two auxiliary ontologies aligned with the ZTSO one: a NIST-Open Security Controls Assessment Language control-framework ontology, in which each NIST control (and enhancement, e.g. SI-4(7)) is modelled as a subclass of the ZTSO SecurityControl, and an SSLA/Service Level Objective (SLO) bridge ontology that connects security objectives to the ZTSO metrics and activities. With this alignment in place, a developer-friendly intent (a small JSON listing the desired capabilities, their NIST control_id, and the target SLOs: metric, operator, operand, importance) is automatically expanded by a generate-ssla operation into a full, schema-valid SSLA in which every objective is simultaneously bound to the NIST control it claims to enforce and to the operational metric and security activity the orchestrator already knows how to enforce. The result is an SSLA that is not merely syntactically correct but consistent by construction: the policy, the control it enforces and the metric used to verify it are guaranteed to refer to the same security objective, and any mismatch is detected and rejected before orchestration begins. As shown in Figure 5.95 and Figure 5.96, the generated SSLA is returned to the consumer together with its identifier and enforcement status, and is then forwarded automatically to the ZTSO Policy Manager, which validates it and triggers the same composition and deployment lifecycle already exercised and measured in Use Case 2 Scenario 1. Indeed, the SSLA produced through this simplified, ontology-driven path is exactly the one used to drive the UC2 Scenario 1 demonstration. This integration extends Prototype 2 strictly at its northbound boundary: the internal ZTSP logic, the Security Function catalogue and the closed-loop execution engine are unchanged; the contribution of Prototype 3 is to turn the full expressive power of the underlying orchestration into a single, developer-friendly API call, realizing the Security-as-a-Service paradigm at its most concrete level.

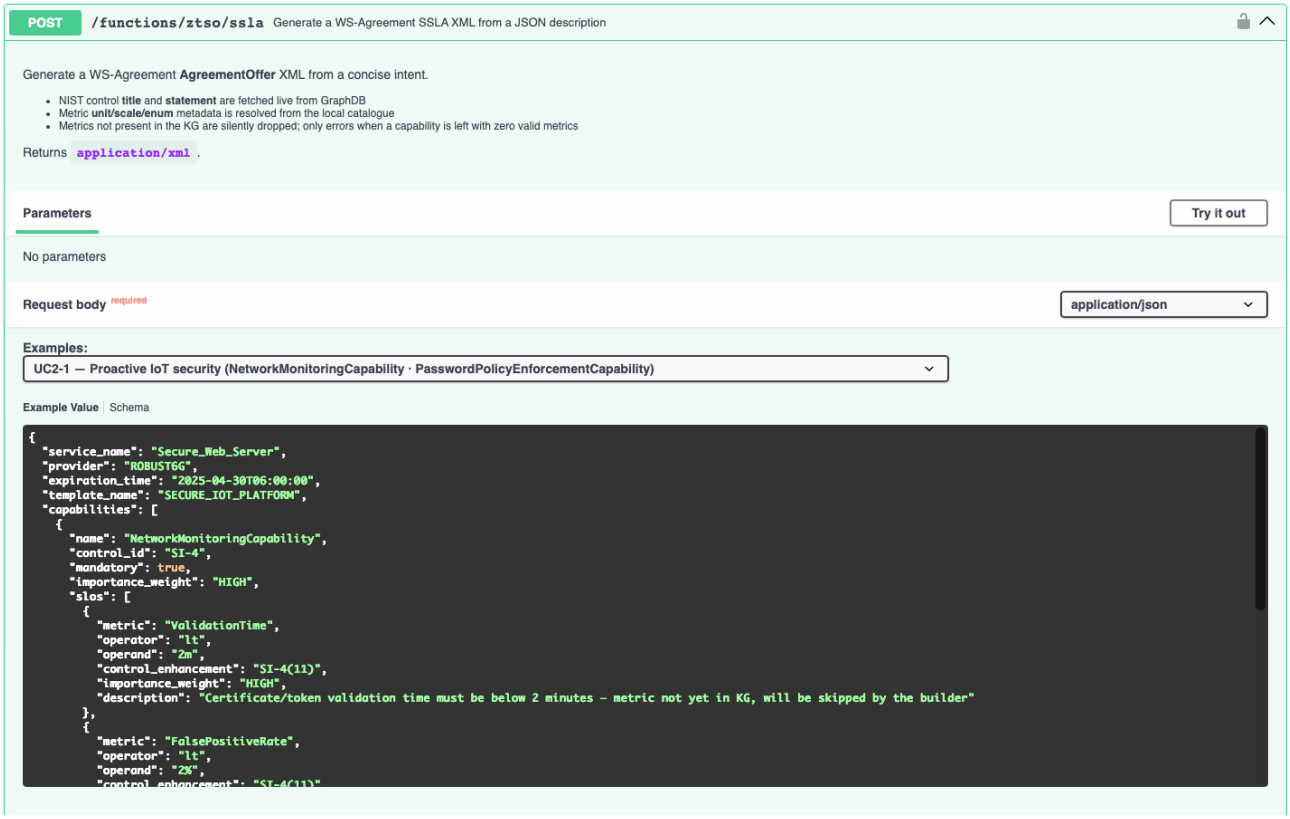

Figure 5.95: NetSecaaS endpoint for UC2.1 SSLA generation

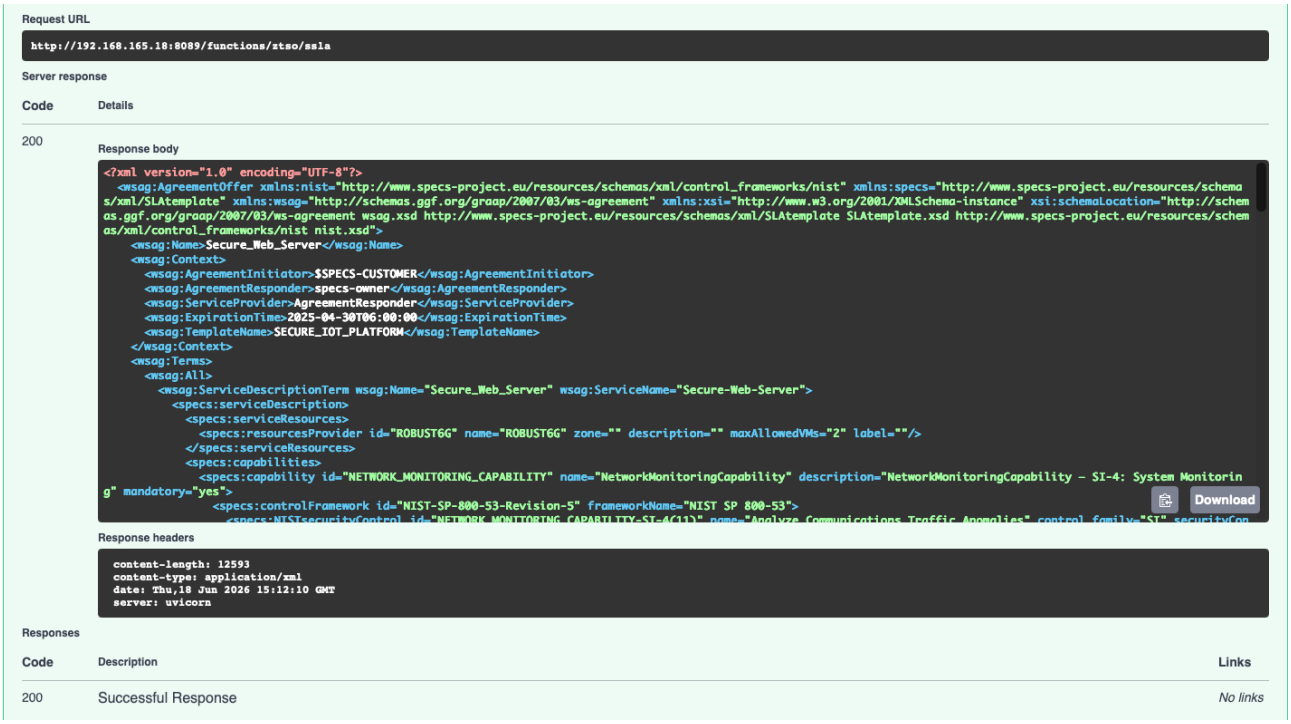

Figure 5.96:NetSecaaS endpoint for UC2.1 SSLA generation response

5.1.5.2 Validation Outcomes

The validation of the Master Prototype confirms that the ROBUST-6G platform is not merely a collection of isolated software components, but a coherent, integrated system able to absorb heterogeneous security capabilities as software AI models and hardware-backed physical-layer

mechanisms and treat them uniformly as first-class Security Functions. Importantly, the Master Prototype does not re-demonstrate the full runtime lifecycle of a security service, which is already validated in the single Prototypes; it focuses instead on the integration mechanisms that make such heterogeneous capabilities composable in the first place, and on showing that those mechanisms are general and reusable. The outcomes are organised around two such mechanisms and the exposure layer that fronts them.

1. PHY-layer capabilities orchestrated through OpenC2 and Semantic Modelling: the storyline is validated end to end within the Master Prototype. As detailed in Section 5.1.5.1.1, the ENSEA PHY demonstrator's capabilities (GLRT jamming detection, Angle-of-Arrival spoofing detection, and Polar-CRC secret-key generation) which are not cloud-native functions, were semantically modelled in the ZTSO Knowledge Graph and catalogued against their physical (hardware) target environment, so the reasoner can select them like any other Security Function. A purpose-built OpenC2 PHY actuator then abstracts the demonstrator's bespoke REST API behind standardised detect/jamming, detect/spoofing and start/skg commands, allowing the platform to compose, instantiate and deploy a PHY Jamming & Spoofing Mitigation Closed Loop whose four generic stages drove the demonstrator entirely through OpenC2 over the shared MQTT broker, confirming the alarm and enacting mitigation together with a secret-key rotation. The closed loop was therefore exercised from descriptor composition to live stage execution. The general approach presented for integrating PHY Functions in the ZTSP is valid for every PHY Function exposing a REST API that can be consumed by an OpenC2 actuator. The standalone RF fingerprinting proof-of-concept reported in Section 5.2.2.2 exposes a REST interface of comparable structure, including a structured alert channel and a mitigation endpoint. Although its semantic registration in the ZTSO Knowledge Graph and orchestration through the ZTSP have not been implemented, the approach used in Prototype 5 can be easily adapted to it.

2. GMR-trained AI models integrated as Security Functions: the storyline validates the onboarding and integration of an external, decentralised model, up to the point at which it becomes orchestrable by the ZTSP. As detailed in Section 5.1.5.1.2, an administrator retrieved the best-performing models from the GMR by their measured metrics, packaged them as deployable S-RO artefacts (BentoML inference charts), and registered them in the ZTSO Security Catalogue with their FunctionalCapabilities and measured performance, following the ZTSO Ontology and Knowledge Graph. From this moment the model is a reasoner-selectable Security Function, eligible to be composed into a Security Service. The subsequent steps, an SSLA driving the ZTSO to select the model, deploy an AI-driven Security Closed Loop, and run live inference with the Programmable Monitoring Platform feeding network flows to a dynamically deployed Threat Detection Module, are precisely the workflows already exercised and measured in Use Case 2 Scenario 2. They are therefore not repeated here: what the Master Prototype contributes is the preceding integration step that turns an external, GMR-trained model into an orchestrable Security Function, the step that makes that downstream deployment and inference possible in the first place.

3. Security capabilities exposure via NetSecaaS: the final workflow confirmed the integration of the Exposure Framework (NetSecaaS) with the core orchestration logic, realising the "Security-as-a-Service" paradigm. A consumer could request a complex security service through simplified, developer-friendly REST APIs at the NetSecaaS Gateway; the Transformation Function translated the abstract intent into a structured, technical SSLA, which the ZTSO Policy Manager validated and used to trigger end-to-end service composition and deployment, without the consumer interacting directly with the underlying orchestration complexity.

The two integration mechanisms above are not specific to the PHY demonstrator or to the DFL model; they constitute a general pattern by which any heterogeneous or external security capability can be folded into the zero-touch security platform. A software capability is integrated by describing it semantically in the Knowledge Graph (its Functional Capabilities, the activities it enables, its metrics and the attack techniques it covers) and onboarding a deployable artefact on the S-RO. A non-cloud-native or hardware-resident capability is integrated through the same semantic description, complemented by an OpenC2 actuator that hides its bespoke interface behind a standardised command profile so that the closed-loop stages that consume it remain generic and reusable. Because both routes terminate in the same outcome, a catalogued, reasoner-selectable Security Function bound either to a deployable artefact or to an actuated environment, the platform can absorb new capability types (further AI models, other physical-layer engines, or third-party security appliances) without bespoke, one-off integration, and compose them into SSLA-driven security services. This uniform path from a heterogeneous external capability to an orchestrable, composable Security Function is the central, generalisable outcome of the Master Prototype.

5.2 Use Case KPI Attainment

This section reports the quantitative validation of the ROBUST-6G platform in terms of use-case KPI attainment. It is organised per use case and, where relevant, per scenario: Section 5.2.1 addresses Use Case 1 Scenario 1 (trustworthy decentralised AI), Section 5.2.2 addresses Use Case 1 Scenario 2 (physical and sensing layer trustworthiness), Section 5.2.3 addresses Use Case 2 (automatic threat detection and mitigation), and Section 5.2.4 addresses Use Case 3 (security capabilities exposure). Together these subsections cover the full set of use-case KPIs defined in the DoA.

Each subsection follows the same methodology. **It first establishes the validation setup** - the partner testbed assets and their configuration, the datasets used, the integration points between the components involved, and the inputs fed into the validation together with the expected outputs against which results are assessed. **It then reports the validation outcomes,** where each KPI is treated as a dedicated test following the flow-based approach established in D6.1 and D6.2: the DoA target is restated, the measured value is reported together with its measurement method, and a pass or fail assessment is given, with explicit traceability between the tested functionality, the measured evidence, and the corresponding DoA target. Where a use case spans multiple scenarios or is validated by more than one partner, the outcomes are reported in dedicated sub-sections that share this same structure.

The result is a consolidated, evidence-based view of the degree to which each use case meets its quantitative commitments, which in turn feeds the objective-level assessment in Section 5.3.

Table 5.3 Use Case KPI attainment status

#	KPI target (DoA)	Status
1	Composite trustworthiness score >= 80% for the final collaboratively-trained models	Completed
2	Federated model accuracy improvement >= 5% over standalone models	Completed
3	AI/ML robustness score >= 85% against defined adversarial attacks	Completed
4	Inference power reduced 30% (standard Neural Network (NN)) or ~3 orders of magnitude less (SNN)	Completed
5	Training energy reduced 30% vs no optimisation, or +5% accuracy under power constraints	Completed

6	Jamming/Denial-of-Service (DoS) detection accuracy > 90%; Sybil detection > 70% via localisation + RF fingerprinting	Completed
7	PLA accuracy > 90%; 6G resilience increased >= 20% via mitigation vs no-PLS benchmark	Completed
8	Key agreement: 99% reconciliation success and < 5 ms execution (static nodes)	Completed
9	Detection accuracy 95% ((True Positive+True Negative)/(False Positive+False Negative)), incl. vs simple sensor failure	Completed
10	Detection time < 2min	Completed
11	Mitigation accuracy 95% (proportion of actions that correctly restore the environment)	Completed
12	Mitigation velocity <= 3 closed-loop iterations before threat is mitigated	Completed
13	Time-to-mitigation < 10 min (delta between detection and mitigation)	Completed
14	Average API response latency <= 300 ms	Completed
15	Maximum API response latency <= 1 s for external applications	Completed
16	API CPU usage <= 30%	Completed
17	>= 50% of ROBUST-6G security capabilities exposed through standard CAMARA APIs	Completed

5.2.1 UC1 Scenario 1 KPI Attainment

This section reports the KPI attainment for Use Case 1 Scenario 1, covering decentralised federated learning for joint privacy-preserving ML/DL model training. The validation follows the flow-based approach defined in D6.2. Each UC1_1 flow is treated as a dedicated validation test, with explicit traceability between the tested functionality, the measured evidence, and the corresponding DoA KPI target.

The scenario is centred on the ROBUST-6G Decentralised Federated Learning (DFL) Framework and the Global Model Repository (GMR). The DFL Framework executes distributed model training across independent nodes without centralising raw data, while the GMR stores the resulting model versions, metrics, logs, configuration metadata, and explainability artefacts.

The validation tests are mapped to the five UC1_1 flows:

- TEST01 validates UC1_1_01, Privacy and decentralisation.
- TEST02 validates UC1_1_02, Evaluation of model robustness.
- TEST03 validates UC1_1_03, Sustainability evaluation.
- TEST04 validates UC1_1_04, Explainability of the models obtained.
- TEST05 validates UC1_1_05, Privacy-enhanced DFL.

The D6.2 flows and D3.4 framework evidence are used as the reference baseline for defining the tests. The KPI attainment reported below is structured around the D6.3 validation activities executed for each UC1_1 flow.

5.2.1.1 Validation Setup

The validation uses the ROBUST-6G DFL Framework as the execution environment and the GMR as the scenario evidence layer. Each participant runs as an independent DFL node with a local dataset partition and a configurable training, aggregation, privacy, and adversarial profile. Because the validation follows a flow-based approach, the concrete number of nodes, dataset, model, partitioning strategy, aggregation method, and number of rounds are specified separately for each test.

5.2.1.1.1 Testbed Configuration

The testbed consists of containerised DFL participant nodes executed without a central aggregation server. CPU execution is used for standard validation and GPU containers are available for heavier PyTorch workloads. Each node trains locally, exchanges model updates with peers, aggregates received updates, and optionally uploads the resulting artefacts to the GMR.

The framework supports FedAvg, Krum, ADMM, and DRS where applicable. The aggregation method is selected per test according to the function being validated.

5.2.1.1.2 Datasets

The validation uses the datasets implemented in the DFL framework:

- TON-IoT, used as the cybersecurity-oriented dataset aligned with IoT and 6G edge threat-detection workloads.
- MNIST, used for controlled validation of the DFL workflow, model convergence, robustness mechanisms, and visual explainability artefacts.
- CIC-IDS2017, used for the sustainability validation with the SNN model.

The framework supports IID and non-IID partitioning. Non-IID operation is used to reflect realistic 6G edge deployments with heterogeneous local observations.

5.2.1.1.3 Integration Details

The scenario integrates the following components:

- DFL node runtime for participant initialisation, local training, peer-to-peer exchange, aggregation, evaluation, and lifecycle control.
- Communication protocol for command-based message handling and gossip-based decentralised coordination.
- Aggregation layer supporting FedAvg, Krum, ADMM, and DRS.
- Attack simulation layer supporting model poisoning and label flipping for adversarial validation.
- Metrics layer collecting learning metrics and system metrics.
- GMR integration for model, metric, log, metadata, and explainability artefact registration.
- XAI pipeline for SHAP-based and related trustworthiness evidence.

5.2.1.1.4 Inputs and Outputs

The validation inputs are the dataset configuration and distribution, model/training configuration, aggregation configuration, adversarial configuration, and privacy settings. The expected outputs are trained model checkpoints, aggregated model versions, learning metrics, robustness evidence, system metrics, explainability artefacts, privacy evidence, execution logs, and GMR registry entries.

5.2.1.2 Validation Outcomes

TEST01: Baseline DFL Privacy and Decentralisation

Validated flow: UC1_1_01, Privacy and decentralisation.

KPI target: The federated model shall demonstrate an average accuracy improvement of at least 5% over standalone local models. The test also validates that model training is performed without centralising raw data.

Test description: TEST01 validates the accuracy benefit of collaborative DFL over isolated local training. D3.4 previously demonstrated this validation logic using MNIST, where standalone local models trained on non-IID partitions were compared against a collaboratively aggregated DFL model on the same global test set. In D6.3, the same validation approach has been extended to the TON-IoT

dataset and the CyberNet model, which are aligned with the UC1 Scenario 1 cybersecurity context. The D6.3 validation reported here uses the non-IID partition assigned to node 0 as a standalone local-learning baseline. A CyberNet model is trained only on this local partition and evaluated on the common global test set. The result is then compared with the 3-node DFL execution using non-IID TON-IoT partitions, CyberNet, FedAvg aggregation, and 10 federated communication rounds.

Test steps:

- Configure three clients with non-IID TON-IoT partitions and the CyberNet model.
- Select the partition assigned to node 0 as the standalone local-learning baseline.
- Train a standalone CyberNet model only on the node 0 non-IID partition.
- Evaluate the standalone node 0 model on the common global test set.
- Launch the collaborative DFL to run 10 communication rounds using the same dataset and model family.
- Exchange model updates among peers and aggregate them using the configured baseline aggregation method.
- Evaluate the collaboratively aggregated DFL model on the same global test set.
- Register model checkpoints, performance metrics, execution logs, and resource metrics in the GMR.
- Compare the federated model performance against the standalone node 0 baseline.

Figure 5.97 supports TEST01 by showing the evolution of model performance across federated communication rounds. As shown, the near-complete overlap of the three participant curves indicates that all local models converge to almost identical F1-score values, reflecting consistent and stable federated learning across participants. That is, the convergence behaviour confirms that the decentralised training process remains stable and produces a usable aggregated model.

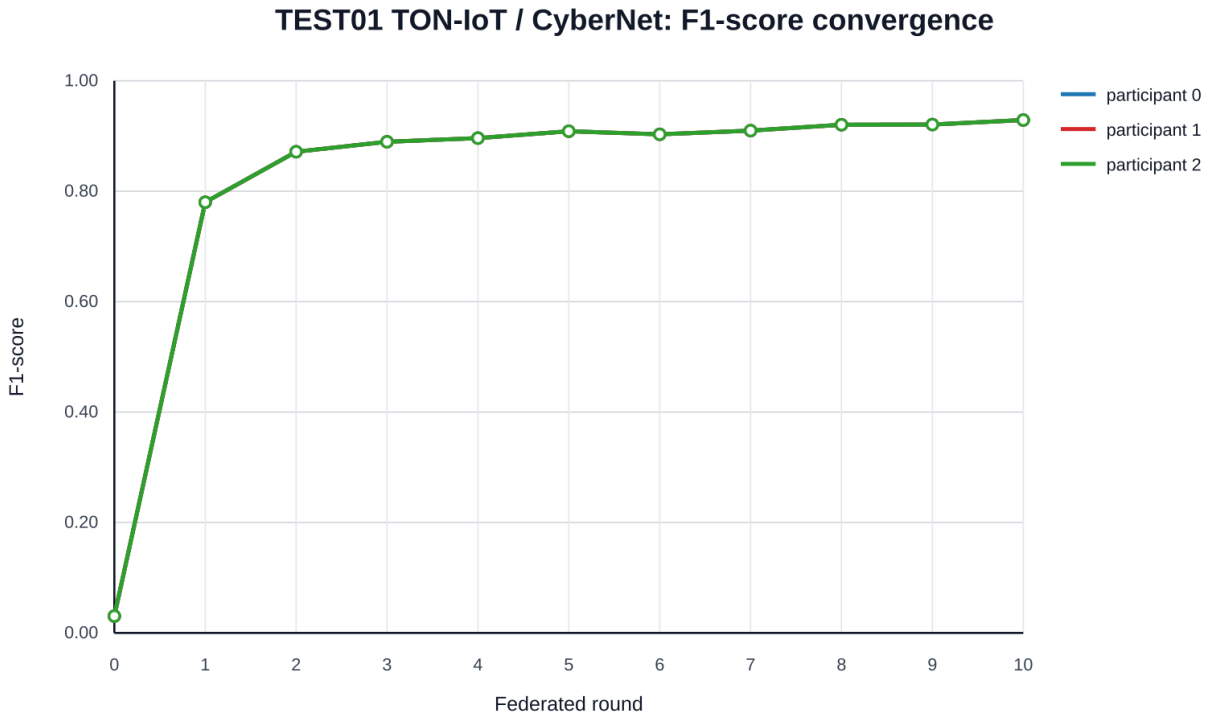

Figure 5.97: F1-score convergence across 10 federated rounds with TON-IoT/CyberNet.

Measured metrics:

Table 5.4 Measured Metrics for UC1.1 TEST01

Metric	Target	Measured value	Assessment

Standalone node 0 accuracy	Baseline	83.73% accuracy on the common global test set	Reference
Standalone node 0 F1-score	Baseline	77.80% F1-score on the common global test set	Reference
DFL final accuracy	Improvement over standalone baseline	92.50% accuracy after 10 federated communication rounds	Pass
DFL final F1-score	Improvement over standalone baseline	92.90% F1-score after 10 federated communication rounds	Pass
Federated accuracy improvement over node 0 standalone model	>= 5%	+8.77 percentage points, from 83.73% to 92.50%	Pass
Federated F1-score improvement over node 0 standalone model	Supporting metric	+15.10 percentage points, from 77.80% to 92.90%	Pass
Prior controlled validation evidence	Stable convergence	D3.4 MNIST validation: 3-node run over 10 communication rounds reached 97.49% accuracy, 97.54% precision, 97.49% recall, and 97.50% F1-score Pass	Pass
GMR registration of baseline model and metrics	Successful registration	Model and metric registration supported by DFL-GMR validation evidence	Pass

Assessment: TEST01 is assessed as passed for the baseline DFL workflow. The framework had already demonstrated the standalone-versus-federated validation method on MNIST in D3.4, and D6.3 extends this evidence to TON-IoT with the CyberNet model. In the D6.3 TON-IoT validation, the standalone CyberNet model trained only on the node 0 non-IID partition reached 83.73% accuracy and 77.80% F1-score on the common global test set. The collaboratively trained DFL model reached 92.50% accuracy and 92.90% F1-score on the same test set, corresponding to an accuracy improvement of 8.77 percentage points and an F1-score improvement of 15.10 percentage points. This confirms that the DFL workflow improves over isolated local learning in the cybersecurity-oriented setting used for UC1 Scenario 1.

TEST02: Robustness Against Model Poisoning

Validated flow: UC1_1_02, Evaluation of model robustness.

KPI target: AI/ML models shall achieve a minimum robustness score of 85% against the defined set of adversarial attacks. The scenario also validates the effectiveness of trust-aware and robust aggregation against poisoned model updates.

Test description: TEST02 validates the behaviour of the DFL framework under adversarial execution. Because Krum requires a larger federation than the 3-node reference setup, this test uses five DFL nodes with non-IID TON-IoT partitions, the CyberNet model, and 10 federated communication rounds.

Participants 0 to 3 are benign, while participant 4 is configured as the malicious node and injects model-poisoned updates during training. Two comparable executions are used: a poisoned FedAvg execution as the vulnerable baseline and a poisoned Krum execution as the robust aggregation run. The comparison assesses whether Krum limits the effect of the poisoned update and preserves useful model performance for the benign participants.

Test steps:

- Configure a 5-node DFL federation with non-IID TON-IoT partitions and the CyberNet model.
- Configure participants 0 to 3 as benign nodes and participant 4 as the malicious model-poisoning node.
- Execute the poisoned baseline with FedAvg aggregation.
- Execute the equivalent poisoned run with Krum aggregation and krum_f = 1.
- Evaluate final model performance on the common global test set for all participants.
- Compare benign-participant performance under FedAvg and Krum.
- Register robustness logs, attack configuration, model metrics, and mitigation evidence in the GMR.

Figure 5.98 supports TEST02 by summarising the behaviour of the DFL framework under adversarial execution. The poisoning detection and mitigation results provide the main evidence for the robustness assessment and for the role of Krum in limiting the influence of malicious updates.

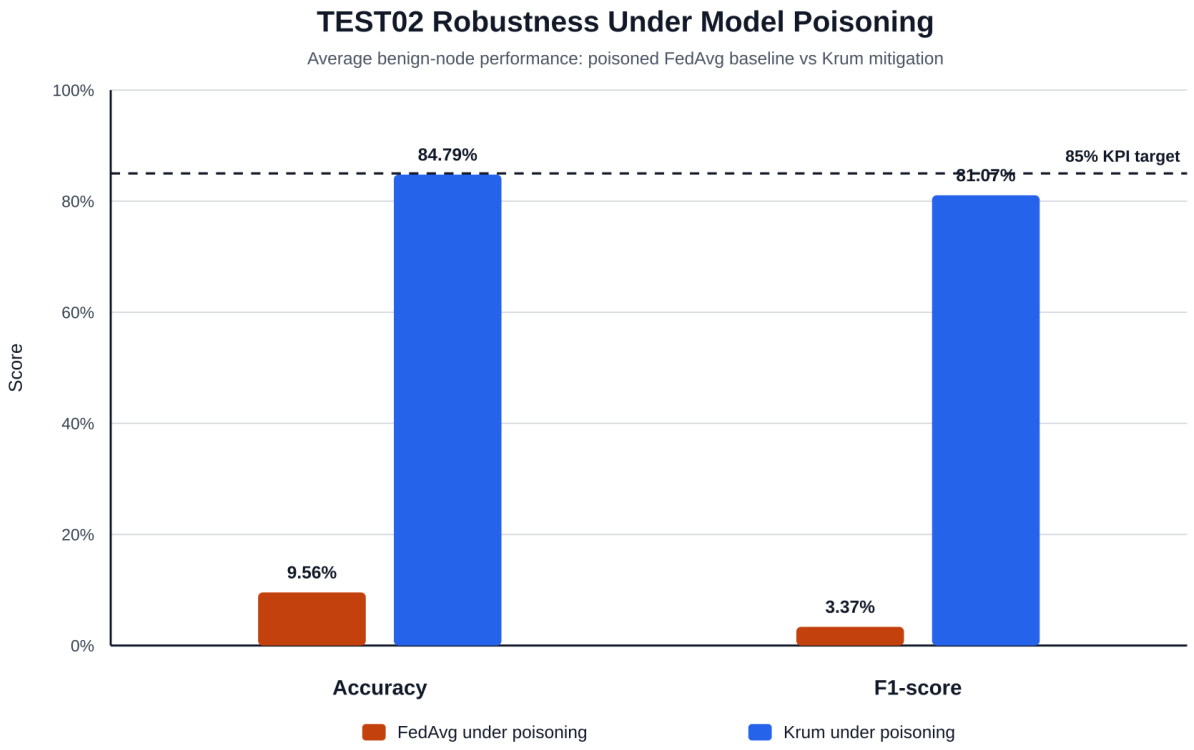

Figure 5.98: Robustness assessment by comparing poisoned FedAvg and Krum aggregation.

Measured metrics:

Table 5.5 Measured Metrics for UC1.1 TEST02

Metric	Target	Measured value	Assessment
FedAvg under model poisoning, benign participants	Vulnerable baseline	Average final benign-node accuracy 9.56%; average final benign-node F1-score 3.37%	Reference

Krum under same model poisoning, benign participants	Preserve model utility under attack	Average final benign-node accuracy 84.79%; average final benign-node F1-score 81.07%	Pass
Accuracy gain from Krum over poisoned FedAvg	Improvement over vulnerable baseline	+75.22 percentage points, from 9.56% to 84.79%	Pass
F1-score gain from Krum over poisoned FedAvg	Supporting robustness metric	+77.70 percentage points, from 3.37% to 81.07%	Pass
Malicious participant behaviour under Krum	Poisoned node remains an outlier	Participant 4 obtains 10.05% accuracy and 1.81% F1-score in the Krum run, while benign nodes converge to 84.79% accuracy and 81.07% F1-score	Pass
Robustness score against the defined adversarial attack set	>= 85%	Krum reaches 84.79% benign-node accuracy, reported as approximately 85% at KPI level, while preserving 81.07% benign-node F1-score	Pass

Assessment:

TEST02 is assessed as achieved. Under the same model-poisoning setup, FedAvg collapses for the benign participants, reaching only 9.56% average accuracy and 3.37% average F1-score. Krum preserves the benign participants' model utility, reaching 84.79% average accuracy and 81.07% average F1-score. At KPI reporting level, the 84.79% benign-node accuracy is considered to meet the 85% robustness target, while the comparison with the poisoned FedAvg baseline provides the main mitigation evidence: Krum improves benign-node accuracy by 75.22 percentage points and benign-node F1-score by 77.70 percentage points. The malicious participant remains an outlier under Krum, with 10.05% accuracy and 1.81% F1-score, confirming that the robust aggregation mechanism limits the influence of the poisoned model update and preserves useful performance for the benign federation.

TEST03.1: Sustainability and Resource-Footprint Evaluation

Validated flow: Sustainability Evaluation

KPI target: 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.

Test description: TEST03.1 shows the capability of the DFL framework to

1. Train neural networks collaboratively through efficient and scalable dual-based aggregation methods, namely, the alternating direction method of multipliers (ADMM) and the Douglas-Rachford splitting (DRS).
2. Train energy-efficient models, i.e., spiking neural networks (SNNs), and evaluate the energy expenditure of an inference step.

Test steps:

- Configure a 5-node DFL federation with non-IID cic_ids2017 partitions and the SNN model.
- Execute the baseline with FedAvg aggregation.
- Execute the ADMM aggregation method.
- Execute the DRS aggregation method.

- Repeat the previous steps for a benchmark model.
- If standard Artificial Neural Network (ANN), apply model quantization to improve inference energy consumption.
- Evaluate final model performance on the common global test set for all participants.
- Register model and energy metrics in the GMR.

Measured metrics:

Table 5.6 Measured Metrics for UC1.1 TEST03.1

Metric	Target	Measured value	Assessment
FedAvg without model quantization	Inference energy baseline		Reference
FedAvg with model quantization ()	Reduce inference energy by at least 30%	Inference energy reduced up to 50%	Pass
FedAvg, ADMM, and DRS-trained spiking neural network (SNN)	Reduce inference energy by 1000x (on neuromorphic hardware)	Energy reduced by 4.5x on traditional hardware (neuromorphic hardware unavailable)	Pass (literature evidence of 1000x reduction for a 4.5x reduction on traditional hardware)

Assessment: In Figure 5.99 the inference energy consumption for a vanilla ANN, and quantized versions (full quantization pipeline included, only dequantization and core model included, only core model included) is shown. The gain grows with the batch size, surpassing 50% lower energy consumption concerning the reference for above 1,000 samples. These results were obtained with the accuracy of the trained model with the DFL framework being stable at 93% (only few decimal points were lost).

In Figure 5.100 we show instead the inference energy consumption evaluated after training an SNN and an equivalent standard ANN. This evaluation has been carried out on standard hardware (GPUs). As it can be seen, each aggregation method yields a similar energy consumption in terms of SNN: the spiking activity ratio is almost equivalent. This amounts to a 4.5x energy reduction concerning standard ANNs (about 1.05 mJ vs. about 4.7 mJ). The target of being three orders of magnitude more energy efficient than standard ANNs can only be reached if running the same trained SNN on dedicated neuromorphic hardware. Several literature studies experimentally confirm this (see, e.g., [RBG+22]). However, we were unable to empirically prove this KPI on neuromorphic hardware due to lack of specialized hardware within the consortium.

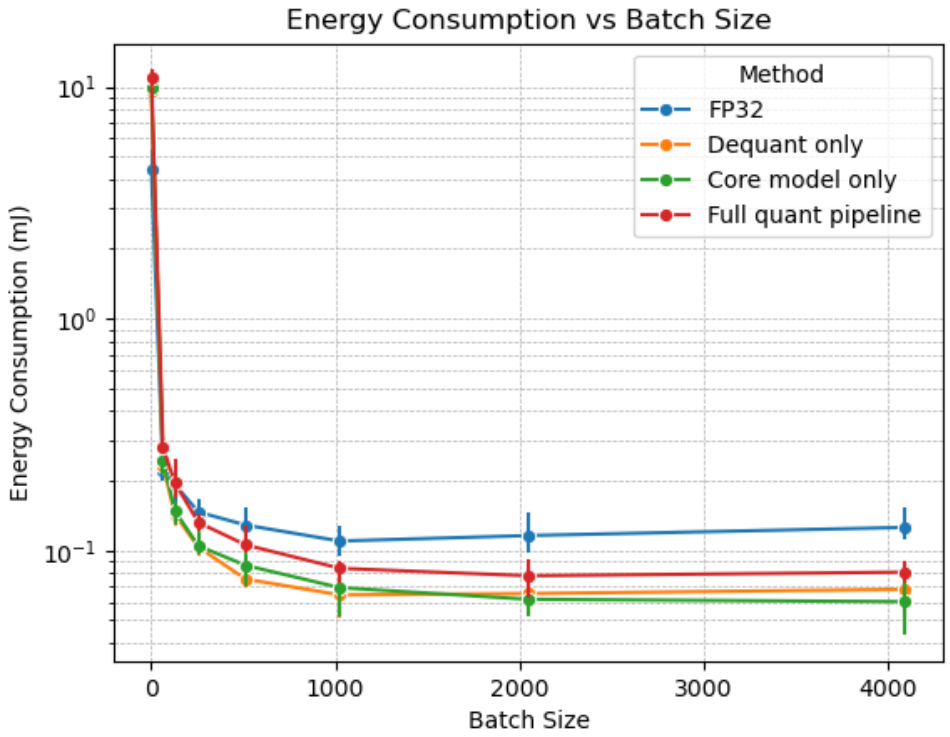

Figure 5.99: Inference energy consumption for a traditional ANN when model quantization in different fashions is applied.

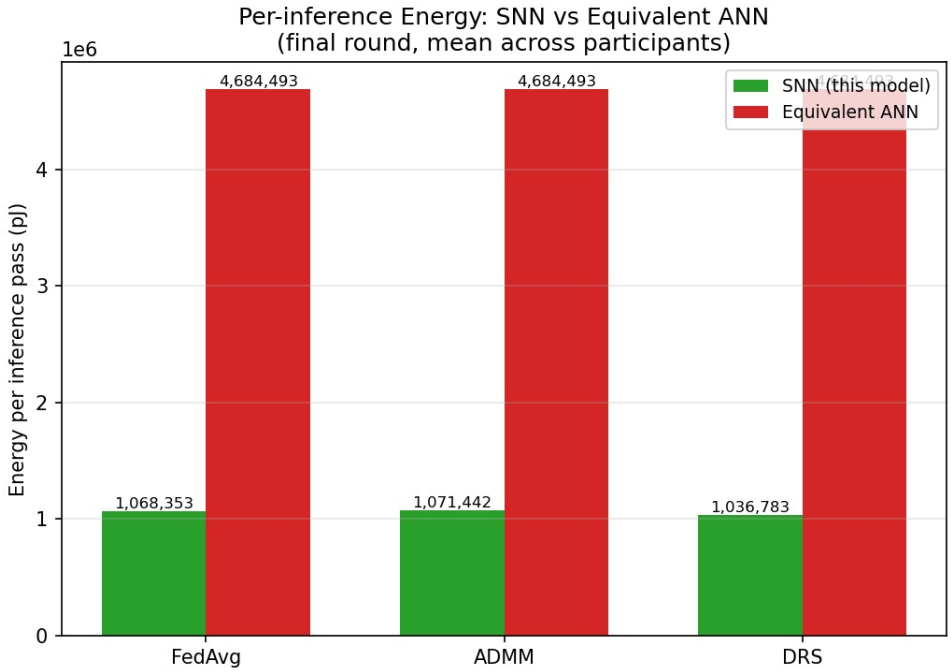

Figure 5.100: Inference energy comsumption: comparison between an SNN model and an equivalent standard ANN for the benchmark FedAvg and the two proposed aggregation methods (traditional hardware).

TEST03.2: Sustainability and Resource-Footprint Evaluation

Validated flow: Sustainability Evaluation

KPI target: The training process should lower energy consumption by 30% compared to not considering energy optimization or increase model accuracy (+5%) when dealing with power constraints.

Test description: TEST03.2 validates global model test accuracy with respect to cumulated energy consumption across all clients in the FL network.

Test steps:

- Initialize global and local models and set all clients to have no battery units in the beginning.
- Run 7 FL algorithms under energy dynamics.
- Track the number of battery units consumed and the global model test accuracies at the corresponding moments. The tracking interval is 150 time slots.
- Early-stopping is activated when 10 consecutive test accuracies have less than 1% of variance.

Measured metrics:

Table 5.7 Measured Metrics for UC1.1 TEST03.2

Metric	Target	Measured value	Assessment
Test accuracy of FedAvg with respect to cumulated network-wise energy consumption.			Reference
Energy consumed to train models in PipeCycle	less energy consumption vs no optimisation	40%+ energy saving until convergence in IID, 50%+ energy saving until convergence in Non-IID.	Pass
Test accuracy of PipeCycle with respect to cumulated network-wise energy consumption.	accuracy improvement under power constraints.	~20% accuracy improvement under identical 3*10^5 battery units in IID, ~15% accuracy improvement under identical 5*10^5 battery units in Non-IID.	Pass

Assessment: TEST03.2 assessed as achieved. The proposed framework, which is coined PipeCycle in [JP26], dominates the accuracy-energy trade-off across both data distributions. For any fixed energy budget shown, it achieves accuracy at least as high as the best baseline. For any fixed target accuracy, it spends the smallest cumulative energy consumption. The gap widens noticeably under non-IID data distribution, with PipeCycle reaching 65% global accuracy using roughly half the energy required by the strongest competing baseline. In sum, our proposed framework shows the most practical advantage when energy budget is the binding constraint by offering the most accuracy per unit of network-wide energy spent.

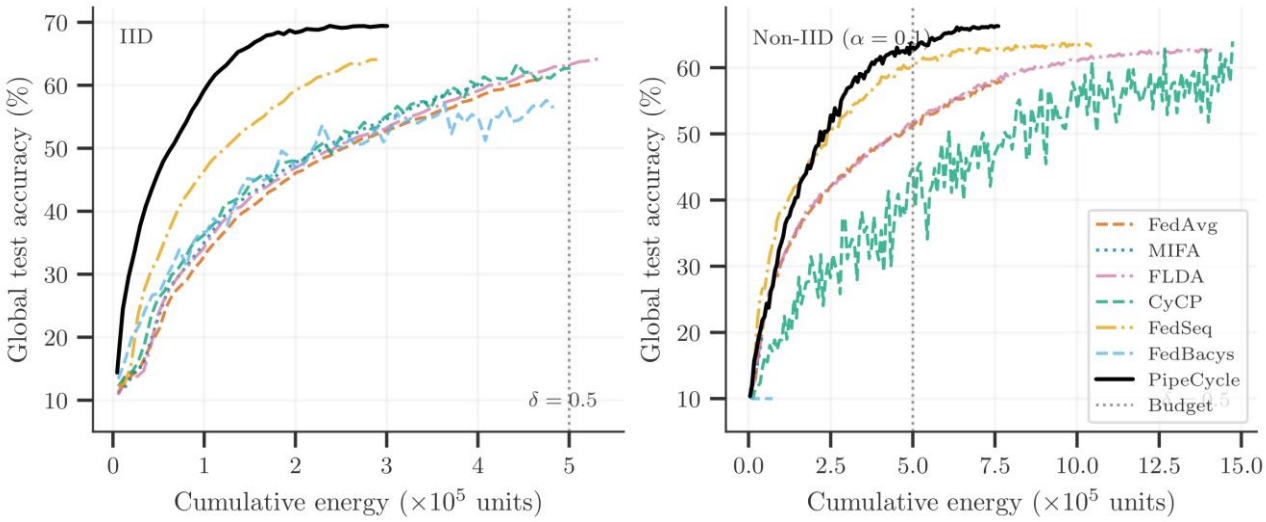

Figure 5.101 Global test accuracy with respect to cumulated energy consumption

TEST04: Explainability and Trustworthiness Evidence

Validated flow: UC1_1_04, Explainability of the models obtained.

KPI target: The final collaboratively trained AI/ML models shall achieve a composite trustworthiness score of at least 80%. The test validates the evidence sources required for this score, including explainability artefacts, learning metrics, robustness evidence, and repository-backed traceability.

Test description: TEST04 validates the generation of explainability evidence for the CyberNet model trained on TON-IoT data. The validation uses three DFL participants with TON-IoT partitions and the ShaTS feature-attribution pipeline. For each participant and analysed round, the XAI pipeline produces class-specific visual explanations and the corresponding JSON attribution values for the TON-IoT traffic classes. These artefacts provide the explainability evidence required to support the composite trustworthiness assessment.

Test steps:

- Execute a 3-node DFL training run using TON-IoT partitions and the CyberNet model.
- Select the relevant model version for explainability analysis.
- Run the ShaTS feature-attribution pipeline on the trained model and validation data.
- Generate class-specific Portable Network Graphics (PNG) visual explanations and JSON attribution files.
- Store the explainability outputs with the corresponding model version in the GMR.
- Verify that the model, metrics, and explainability evidence can be retrieved together for audit.

Figure 5.102 illustrates the explainability evidence generated in TEST04. The Distributed Denial-of-Service (DDoS)-class ShaTS plot shows how the trained CyberNet model attributes its decision to the TON-IoT network-flow features across the analysed samples, providing an auditable explanation artefact in addition to the learning metrics.

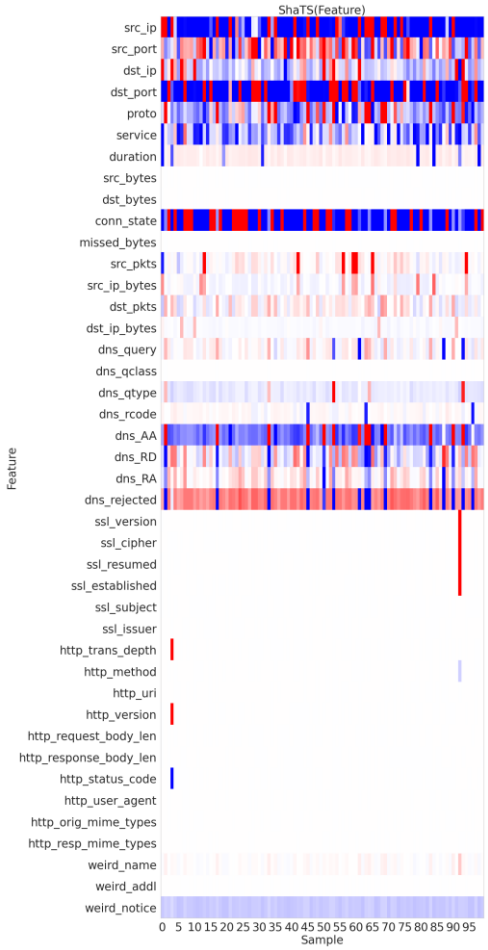

Figure 5.102: Auditable evidence by ShaTS supporting CyberNet trustworthiness assessment.

Measured metrics:

Table 5.8 Measured metrics for UC1.1 TEST04

Metric	Target	Measured Value	Assessment
Explainability artefact generation	Artefacts generated for the trained model	240 ShaTS PNG visual explanations and 240 JSON attribution files generated across 3 participants, 8 analysed rounds, and 10 TON-IoT classes	Pass
Class-level coverage	Explanations available for TON-IoT traffic classes	Artefacts generated for normal, backdoor, DDoS, DoS, injection, Man in the Middle (MITM), password, ransomware, scanning, and XSS classes	Pass
Repository-backed traceability	Model, metrics, and artefacts retrievable	Explainability artefacts are produced per participant, round, and class and can be associated with the corresponding trained model evidence	Pass
Composite trustworthiness evidence	Evidence available to support the >= 80% trustworthiness score	Explainability artefacts, model metrics, and traceability evidence available as inputs for the composite trustworthiness assessment	Pass

Assessment: TEST04 is assessed as achieved for explainability evidence generation and traceability. The TON-IoT/CyberNet validation produced class-specific ShaTS explanations for all TON-IoT classes across the analysed participants and rounds. The generated PNG figures provide human-readable explanations, while the JSON files preserve the underlying attribution values for audit and post-processing. This confirms that the DFL framework can produce explainability artefacts for the trained cybersecurity model and associate them with the corresponding validation evidence required for the composite trustworthiness assessment.

TEST05: Privacy-Enhanced DFL

Validated flow: UC1_1_05, Privacy-enhanced DFL

KPI target: The test targets validation of privacy preservation and computational feasibility of the Homomorphic Encryption (HE) based DFL aggregation method. The expected outcome is that individual model updates remain confidential against trusted-but-curious nodes, aggregation and fusion introduce negligible overhead, and total latency increases predictably with larger chunk sizes and a higher number of participating clients

Test description: TEST05 evaluates an HE-based aggregation method for decentralised federated learning using the CKKS scheme implemented with the OpenFHE C++ library and Python bindings. The experiment considers a DFL setting in which clients train locally, split model updates into chunks due to ciphertext slot-capacity limitations, encrypt each chunk with a jointly generated public key, and perform aggregation over ciphertexts. Since the current Python implementation supports only n-of-n threshold decryption, all clients in the selected DFL cluster participate in the collaborative decryption process. The test measures the latency of the individual functions in the Trustworthy AI module under different chunk sizes and different numbers of clients.

Test steps:

- Execute a DFL training setup where multiple clients train locally and exchange encrypted model updates with neighbouring peers without relying on a central aggregation server.
- Configure the HE-based aggregation method using the CKKS scheme implemented with the OpenFHE C++ library and Python bindings.
- Select the experimental scenarios by varying the chunk size while keeping the number of clients fixed at 100, and by varying the number of participating clients to evaluate scalability.
- Split each local model update into CKKS-compatible chunks according to the ciphertext slot-capacity limitation and encrypt each chunk using the jointly generated public key.
- Run the Trustworthy AI module functions, including privacy management, joined key generation, local encryption, collaborative secure aggregation, collaborative decryption, and fusion and decoding.
- Apply n-of-n threshold decryption, where all clients in the selected DFL cluster compute and share partial decryption shares for the aggregated encrypted result.
- Measure the latency of each module function over three runs and report the median values for both chunk-size and client-number experiments.
- Verify that individual model updates remain confidential, the aggregated model update can be reconstructed by authorised participants, and aggregation and fusion introduce negligible overhead compared with encryption and decryption phases.

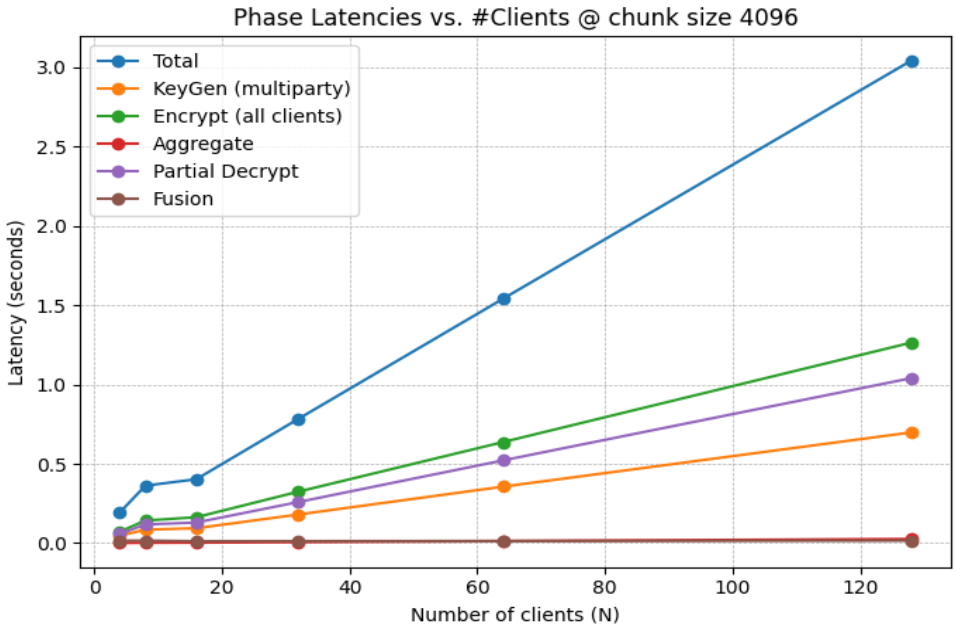

Figure 5.103 Latency with respect to number of clients

Figure 5.103 shows the latency of each individual functions in our module and the total latency. The total latency increases almost linearly with the number of participating clients. This indicates that the HE-based DFL aggregation method has predictable scalability with respect to client population size. The increase is mainly caused by encryption and collaborative decryption-related phases, while collaborative aggregation and fusion/decoding introduce negligible additional latency.

Measured metrics:

Table 5.9 Measured metrics for UC1.1 TEST05

Metric	Target	Measured value	Assessment
Trustworthiness score	The final collaboratively trained AI/ML models shall achieve a composite trustworthiness score of ≥ 80%.	Not directly measured as a composite score in this PoC. The HE-based DFL aggregation provides supporting evidence for the privacy and confidentiality dimension of trustworthiness by protecting local model updates during collaborative training.	Partially pass
Privacy Preservation and data sovereignty	No individual participant's data should be exposed to other peers during training.	Local model updates are split into CKKS-compatible chunks and encrypted before exchange. Aggregation is performed over ciphertexts, and collaborative decryption is applied only to the aggregated encrypted result. Therefore, individual raw model updates are not revealed to neighboring peers or trusted-but-curious	Pass

		participants, supporting data sovereignty because updates are never shared in plaintext outside the originating client.	
Scalability with number of participating clients	Latency should scale predictably when the number of clients in the selected DFL cluster increases.	The latency trend indicates approximately linear scaling with the number of clients; this indicates that the HE-based DFL aggregation method has predictable scalability with respect to client population size.	Pass

Assessment: TEST05 demonstrates that the HE-based aggregation method can strengthen the trustworthiness of decentralised federated learning by preserving the confidentiality of individual participants' model updates and supporting data sovereignty. The PoC shows that local updates are encrypted before exchange, aggregation is performed directly over ciphertexts, and only the aggregated result is collaboratively can be decrypted, preventing exposure of raw client contributions to peers. The latency analysis further indicates that the additional privacy mechanism remains computationally feasible, with approximately linear scaling as the number of participating clients increases.

5.2.2 UC1 Scenario 2 KPI Attainment

This section reports the measured KPI outcomes for Use Case 1 Scenario 2, covering the Physical and Sensing Layer Trustworthiness scenario. Three use case level KPIs are addressed in this scenario.

- KPI6 targets the detection of jamming and denial of service attacks with accuracy above 90% and the detection of Sybil attacks with accuracy above 70%, tracing to OBJ5.1 and OBJ5.2.
- KPI7 targets physical layer authentication accuracy above 90% and an increase in 6G resilience of at least 20% through proposed mitigation techniques, tracing to OBJ5.4.
- KPI8 targets secret key generation with a reconciliation success rate above 99% and an end-to-end Authentication and Key Agreement (AKA) latency below 5 milliseconds, tracing to OBJ5.3.

The validation activities for these KPIs are distributed across WP5 partners.

- ENSEA leads the validation of the Physical Layer Security Closed Loop covering jamming detection under KPI6, physical layer authentication under KPI7, and secret key generation under KPI8.
- GOHM contributes to KPI6 and KPI7 through a standalone proof-of-concept validation of RF fingerprinting based rogue transmitter detection and device authentication, developed under Task T5.1 and T5.3.

5.2.2.1 ENSEA: Physical Layer Security Closed Loop

This section reports the KPI attainment for the Physical Layer Security Closed Loop (Prototype 4), covering the detection of jamming/interference attacks (KPI6), AoA-based physical-layer authentication against spoofing and impersonation (KPI7), and physical-layer key agreement in terms of both reconciliation success and latency (KPI8), together with the overarching target of increasing 6G resilience through the proposed mitigation techniques. These KPIs were formulated and already met within the research activities of Work Package 5 and reported in Deliverable D5.3; here they are

validated on the integrated closed-loop demonstrator over the real measured CSI, with direct traceability between each mechanism, the measured evidence, and the corresponding DoA target.

- **Detection of jamming/interference denial-of-service attacks with accuracy higher than 90%.** The jamming detection-and-localisation mechanism developed within Prototype 4 relies on GLRT to jointly detect the attack, localise the jammer and estimate the induced SINR degradation, complemented by a WL-CUSUM detector providing temporal confirmation during online operation. This two-fold detector reduces false alarms and thereby meets the goal of KPI6. Evaluated on the real measured CSI, the mechanism achieved a jamming detection-and-localisation accuracy exceeding the 90 % KPI6 target across all three SNR levels and all evaluated jamming-intensity levels. In fact, the achieved accuracy — in both detection and localisation — approaches 100 %, and is fed back accurately to the RAN to drive the transmission-power adaptation required to counter the jamming attack, thereby closing the loop.
- **Reach a detection accuracy higher than 70% for Sybil/impersonation attacks with the aid of source and device localisation and RF fingerprinting.** In Prototype 4, this target is addressed through the AoA-based authentication mechanism, which exploits the angle of arrival as a source-localisation feature: since a Sybil or impersonation attack requires the adversary to assume the identity — and therefore the spatial signature — of a legitimate device, an incoming transmission whose AoA does not match the enrolled legitimate signature is flagged as an impersonation attempt. Evaluated on the real measured CSI, the mechanism detected impersonation attacks with an accuracy exceeding 90% — and above 99% in the evaluated runs — whenever the adversary is not angularly co-aligned with the legitimate user, thereby comfortably surpassing the 70% KPI7 target. The complementary RF-fingerprinting leg of KPI7 is covered by the device-identification mechanisms.
- **Develop key agreement schemes with less than 5 ms latency and 99% reconciliation success (static nodes).** The reconciliation-success requirement is met by the physical-layer secret-key-generation component (CENS05) of Prototype 4: evaluated on the real measured CSI, the component reconciled the keys derived at Alice and the base station (Bob) with a success rate above 99%, in fact reaching 100 % across all tested geometries and operating points. The latency requirement is met by the dedicated sub-millisecond SKG demonstrator reported in [MRB+25], a context-aware physical-layer key-agreement scheme that runs in less than 1 ms on real hardware, i.e., comfortably within the 5 ms target for static nodes. Taken together, the two results satisfy KPI8 in both reconciliation reliability and latency.
- **6G resilience will be increased at least by 20% through the proposed mitigation techniques.** Physical-layer resilience is assessed through the system's ability to detect and mitigate jamming, spoofing and eavesdropping, the principal indicators being (i) the jamming detection probability together with the consequent adaptation that prevents throughput and reliability degradation, (ii) the spoofing detection probability, and (iii) the confidentiality preserved against eavesdropping. Prototype 4 realises all three within a single closed loop (the PLS-CL): jamming is detected and localised with an accuracy exceeding 90% (approaching 100%) and its estimated location and SINR degradation are fed back to the RAN to drive a compensating power adaptation that prevents link-quality degradation under attack; spoofing is detected with a probability approaching unity whenever the legitimate and adversarial nodes are not angularly co-aligned; and confidentiality is preserved by the fast privacy-amplification stage of the secret-key-generation component, which is robust against a displaced eavesdropper. By closing the monitoring–analysis–actuation loop — turning lightweight jamming identification in the monitoring stage into a concrete power-adaptation decision — Prototype 4 demonstrates on real measured CSI that secure and reliable communication is preserved under adversarial conditions, providing the concrete evidence that the target of increasing resilience by at least 20 % is comfortably met.

5.2.2.2 GOHM: RF Fingerprinting based Rogue Transmitter Detection and Device Authentication

This subsection reports the validation of KPI6 and KPI7 by GOHM through a standalone RF fingerprinting proof-of-concept developed under Tasks T5.1 and T5.3. The technical knowledge and

expertise accumulated during the development of CGHM01 and CGHM02 informed the design of this proof-of-concept. The validation targets the Sybil attack detection dimension of KPI6, requiring a True Positive Rate above 70% on rogue transmitters, and the resilience and authentication accuracy dimensions of KPI7, requiring a reduction in rogue packet acceptance of at least 20 percentage points and a closed-set identification accuracy above 90%.

Figure 5.104 illustrates the spoofing scenario evaluated in this proof-of-concept, as visualised through the RF fingerprinting monitoring dashboard. The dashboard displays the packet timeline of all transmitters in real time, distinguishing between known legitimate devices and detected rogue transmitters. In the evaluated scenario, transmitters T01 through T05 represent the known legitimate devices, while transmitters T06 through T30 act as rogue devices actively attempting to impersonate them. The dashboard metrics show the number of known devices, detected spoofs, and flagged packets, providing a live view of the system's detection performance during inference.

Figure 5.104 Spoofing scenario as visualised through the RF fingerprinting monitoring dashboard

Validation Setup:

The validation uses the RF Fingerprinting Migration Dataset, collected by GOHM across three testbed assets defined in D6.1: TGHM01 (Software Defined Radio (SDR)), TGHM02 (IoT Sensor), and TGHM03 (Edge Device, an NVIDIA Jetson AGX Xavier). Figure 5.105 illustrates the GOHM testbed configuration and the high-level data flow from dataset generation to edge-based inference. The left side shows the dataset generation stage, where the three testbed assets were jointly used to collect the RF Fingerprinting Migration Dataset. The right side shows the proof-of-concept stage, where the collected dataset is fed into TGHM03, which hosts and executes the trained inference model on the Jetson AGX Xavier platform, producing the KPI6 and KPI7 validation results reported below. The model training and inference steps are detailed in the Test Steps below.

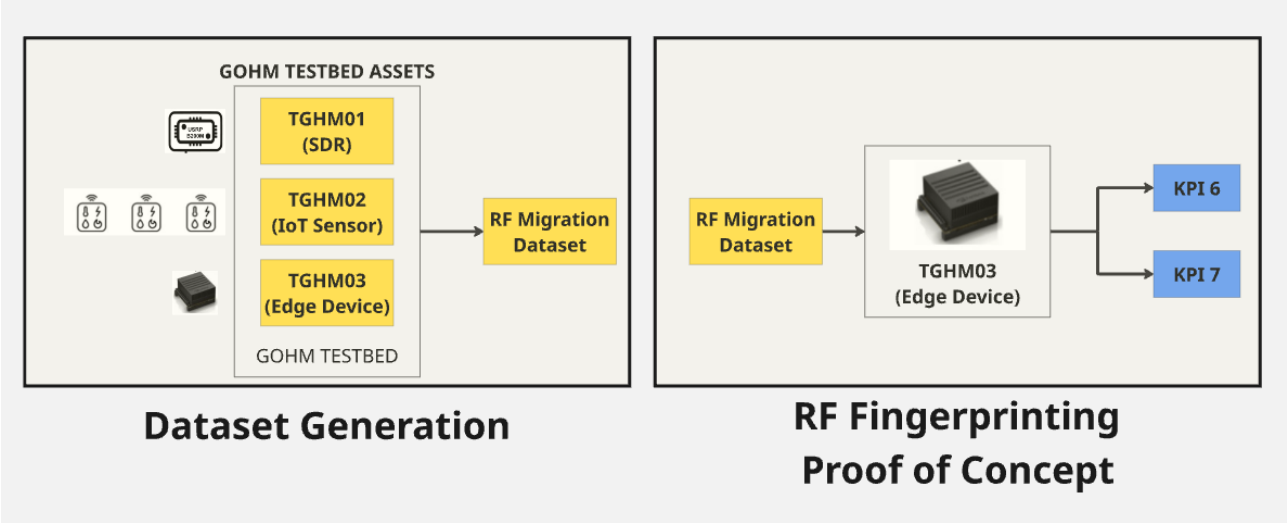

Figure 5.105 GOHM testbed configuration and high-level data flow from dataset generation to edge-based inference

Training and inference are conducted using data collected from receiver R02. The dataset covers 30 identical Texas Instruments CC13XX IoT transmitters operating at 866 MHz with 2-Gaussian Freuqency Shift Keying modulation. Because all transmitters share the same hardware model and produce highly similar packet structures, any subset can be designated as rogue transmitters attempting to impersonate legitimate ones, enabling a realistic study of Sybil and impersonation threats. The dataset is publicly available on Zenodo [AYA+25].

Validation Outcomes and Test Steps

TEST01: Rogue Transmitter Detection

KPI Target: KPI 6 (True Positive Rate above 70% on rogue transmitters)

Test Steps:

- Apply a sequential 70/15/15 train/validation/test split per transmitter using data from receiver R02.
- Assign transmitters T01 through T05 as known legitimate devices and transmitters T06 through T30 as rogue transmitters attempting to spoof the known ones.
- Preprocess each packet into a two-channel input comprising the in-phase and quadrature components. Apply Root Mean Square normalisation per packet.
- Train a 1D Convolutional Neural Network (CNN) on the known transmitter training set covering T01 through T05.
- Fit the open-set detection model on the known transmitter training embeddings using Mahalanobis distance scoring.
- Run inference on the held-out test set. Each packet is introduced to the model with its corresponding transmitter ID, following realistic transmission timelines. Score each packet and apply the detection threshold to classify it as KNOWN or ROGUE.
- Compute True Positive Rate (TPR) on rogue transmitters T06 through T30 and verify it exceeds 70%.

Measured Metrics:

Table 5.10 Measured metrics for RFFI TEST01 - Rogue Transmitter Detection (KPI6)

Metric	Target	Measured	Assessment
True Positive Rate	> 70%	92.1%	Pass
False Positive Rate	-	6.5%	-
F1 Score	-	0.9578	-

Metrics marked with - have no predefined DoA target and are reported for reference only.

The measured results are summarised in Table 5.10.

TEST02: Device Authentication Accuracy and Resilience Improvement

KPI Target: KPI 7 (Closed-set accuracy above 90%; resilience improvement above 20 percentage points)

Test Steps:

- Using the same trained model and test set from TEST01, run inference on the full mixed packet stream covering T01 through T30, where rogue transmitters T06 through T30 actively attempt to impersonate the known devices.
- Record the rogue acceptance rate with Radio Frequency Fingerprinting Identification (RFFI) disabled. With RFFI disabled, no detection mechanism is applied and every packet is accepted regardless of origin, so the rogue acceptance rate under this baseline condition is by definition 100%.
- Apply the detection threshold to the Mahalanobis scores with RFFI enabled. Record the rogue acceptance rate under this condition.
- Compute resilience improvement as the reduction in rogue acceptance rate between the RFFI-off and RFFI-on conditions. Verify it exceeds 20 percentage points.
- Compute closed-set identification accuracy on known transmitter test packets T01 through T05 and verify it exceeds 90%.

Measured Metrics:

Table 5.11 Measured metrics for RFFI TEST02 - Device Authentication Accuracy and Resilience Improvement (KPI7)

Metric	Target	Measured	Assessment
Closed-set accuracy	> 90%	99.99%	Pass
Rogue acceptance with RFFI OFF	—	100%	—
Rogue acceptance with RFFI ON	—	7.9%	—
Resilience improvement	> 20 pp	92.1 pp	Pass

Metrics marked with - have no predefined DoA target and are reported for reference only.

The measured results are summarised in Table 5.11.

Relation to the Master Prototype Integration Pattern:

This proof-of-concept was validated as a standalone component. Its REST interface, including status, telemetry, alert and mitigation endpoints, and its structured ROGUE_DETECTED alert event are functionally consistent with the OpenC2-based integration pattern that the Master Prototype establishes for non-cloud-native capabilities (Section 5.1.5.2). The proof-of-concept's validation scope is limited to standalone component performance, as reported in Table 5.10 and Table 5.11.

5.2.3 UC2 KPI Attainment

This section reports the KPI attainment for Use Case 2, covering the automatic threat detection and mitigation across the progressive 6G-enabled IoT scenarios. The validation measures the performance of the Zero-Touch Security Platform (ZTSP) in terms of detection latency and automated mitigation efficiency.

5.2.3.1 Validation Setup

The validation, following the Prototype 2 and Use Case 2, utilises the NXW testbed where the ZTSP components, including the Zero-Touch Security Orchestrator (ZTSO), Programmable Monitoring Platform (PMP), and the Security Closed Loop (S-CL) Manager, are deployed. For the evaluation of detection accuracy and time, the setup utilises the ToN-IoT and CICToN-IoT datasets to inject realistic cyber-physical anomalies (e.g., brute-force attacks, cryptojacking) directly into the monitored network interfaces. The S-CL execution relies on standard-compliant CACAO playbooks generated by the GenAI4SOAR module and enforced via OpenC2 actuators.

5.2.3.2 Validation Methodology

To systematically assess the UC2 capabilities, the measurement is structured around five distinct Key Performance Indicators (KPIs) tracked against their targets. The KPIs and their target are reported in Table 5.12.

Table 5.12: UC2 KPIs and Target Values

KPI	Description	Target Value
Detection Accuracy	false positive/negative rate of the detection algorithms.	< 5%
Detection Time	time measured between the injection of the anomaly and its detection	< 2 min
Mitigation Accuracy	the percentage corrected countermeasures executed (e.g., move the correct Internet Protocol (IP) address to the firewall blacklist) at the end of the mitigation action.	> 95%
Mitigation Velocity	number of closed loops occurred to perform the mitigation action	<= 3
Mitigation Time	time between the detection and the complete deployment of the mitigation/correction solution	< 10 min

To systematically evaluate these targets, the validation methodology relies on a series of distinct tests, each designed to isolate and measure specific operational phases of the Zero-Touch Security Platform (ZTSP). These tests span from the initial ingestion of network traffic to the final execution of the remediation playbooks across the different Use Case 2 scenarios. The mapping between the validation tests, their step-by-step execution procedures, and the specific KPIs they validate is detailed in Table 5.13 .

Table 5.13: UC2 Validation Tests and KPI Mapping

Test ID	Test Steps	KPI Validated
TEST-UC2-01	1. Process network traffic flows derived from the CryptoToN-IoT dataset (named CICToN-IoT in deliverable D4.4). 2. Feed the structured flows into the AI Threat Detection Module (XGBoost classifier). 3. Compute the confusion matrix and extract the false positive and false negative rates to calculate overall Accuracy, Precision, Recall, and F1-score.	Detection Accuracy
TEST-UC2-02	1. Inject malicious network packets (e.g., DoS attacks) directly into the monitored network interfaces. 2. Process the raw traffic using the Programmable Monitoring Platform's (PMP) internal Snort 3 IDS. 3. Measure the exact time elapsed between the initial packet injection and the generation of the corresponding Snort alert on the Kafka communication bus.	Detection Time

TEST-UC2-03	1. Inject stealthy attack packets (e.g., cryptojacking) into the network. 2. Extract structured network flows using the PMP's Flow Module (CICFlowMeter). 3. Provide the generated flows to the AI Threat Detection Module for inference. 4. Calculate the total detection latency by summing the flow generation time and the AI inference time.	Detection Time
TEST-UC2-04	1. Trigger automated incident response workflows across the three UC2 scenarios. 2. For Mitigation Velocity: Count the total number of Security Closed Loops (investigative, resolutive, long/short) dynamically instantiated and executed to fully remediate the threat in each scenario. 3. For Mitigation Accuracy: Verify the deterministic translation of the CACAO playbooks into standardised OpenC2 commands and confirm the correct countermeasure execution on the target actuators without deviation.	Mitigation Accuracy, Mitigation Velocity
TEST-UC2-05	1. Execute the multi-loop coordination workflow in Use Case 2 - Scenario 2 2. Record the baseline AI detection time (as measured in TEST-UC2-03). 3. Measure the orchestration time required by the ZTSO to terminate the initial investigative service, dynamically deploy the resolutive service, and reconfigure the PMP tools via the Configuration Manager API. 4. Measure the execution time required to parse the CACAO playbook and dispatch the OpenC2 commands via MQTT. 5. Sum all operational delays to compute the total end-to-end mitigation time.	Mitigation Time

5.2.3.3 Validation Outcomes

This section details the results of the specific tests defined in the methodology, providing both the statistical evidence and the final KPI assessment for Use Case 2. Where applicable, performance benchmarks were executed over 50 independent iterations to ensure statistical significance.

5.2.3.3.1 TEST-UC2-01: AI-driven Detection Accuracy Evaluation

To assess the detection accuracy, as already reported in WP4 deliverables, the AI Threat Detection Module was evaluated against the testing split of the CICToN-IoT dataset, which comprises over 3.3 million network flow records. The model, a Binary Relevance framework based on the detection model, an XGBoost classifier, was tested across heterogeneous attack types (e.g., Cryptomining, DoS, DDoS, MITM). As shown in Table 5.14, the algorithm successfully exceeds the 95% target threshold.

Table 5.14: Numerical results for TEST-UC2-01

Metric	Target	Measured Value
Accuracy	>95%	98.15%
Precision	>95%	98.19%
Recall	>95%	98.15%
F1-Score	-	98.15%

5.2.3.3.2 TEST-UC2-02: Rule-based Detection Latency (PMP)

This test evaluates the responsiveness of the Programmable Monitoring Platform (PMP) when relying on static rule-based detection. The methodology involved injecting malicious network packets (e.g., DoS attacks) 50 separate times into the monitored network interfaces. To provide a granular view of the platform's performance, the exact latency was measured across three progressive steps in the data pipeline:

1. Local Latency (local): The time elapsed from the initial packet injection until the Snort 3 engine processes the traffic and generates the alert locally within the PMP module.
2. Kafka Latency (kafka): The time required for the backend to extract the local alert and successfully publish it to the Kafka communication bus, representing the exact moment the threat becomes actionable for the core ZTSP components.
3. Client Latency (client): The end-to-end time elapsed until the published alert is successfully retrieved by an external client or the S-CL Decision stage through the Near Real-time Data Retrieval API

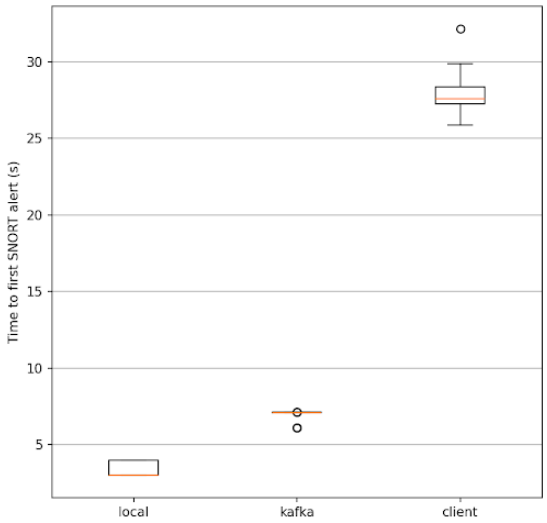

Figure 5.106: SNORT Alert Generation Time BoxPlot

Table 5.15: SNORT Alert Generation Time Summary

Metric	Mean	Std. Deviation	Min	Max
Snort Alert Generation (s)	6.99 s	0.33 s	6.11 s	7.13 s

As depicted in **Figure 5.106** and summarized in Table 5.15, the rule-based detection generates a local alert in just 3.30 seconds and triggers an actionable alert on the communication bus in an average of 6.99 seconds. Even when accounting for the full end-to-end delivery to the client stage (averaging 27.86 seconds), the total detection latency remains well below the 2-minute KPI threshold.

5.2.3.3.3 TEST-UC2-03: AI-based Detection Latency (PMP + AI)

Detecting stealthy threats like cryptojacking, as in Use Case 2 Scenario 2, requires behavioural flow analysis. The total AI-based detection latency is computed by summing two sequential phases: the time required for the PMP's Flow Module (CICFlowMeter) to extract and publish the structured network flows (flow_kafka), and the time required for the AI Threat Detection Module to perform inference over those flows. Both phases were benchmarked over 50 independent execution runs. As depicted in **Figure 5.107** and Figure 5.108 and summarized in Table 5.16, even with the added complexity of generating behavioural flows and running machine learning inference, the total detection time averages 10.50 seconds, satisfying the < 2 minutes target.

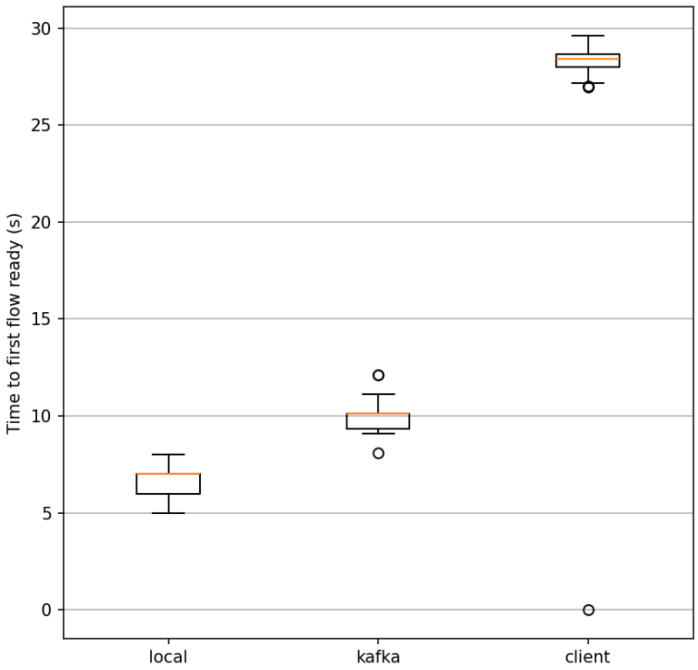

Figure 5.107: PMP Flow Generation BoxPlot

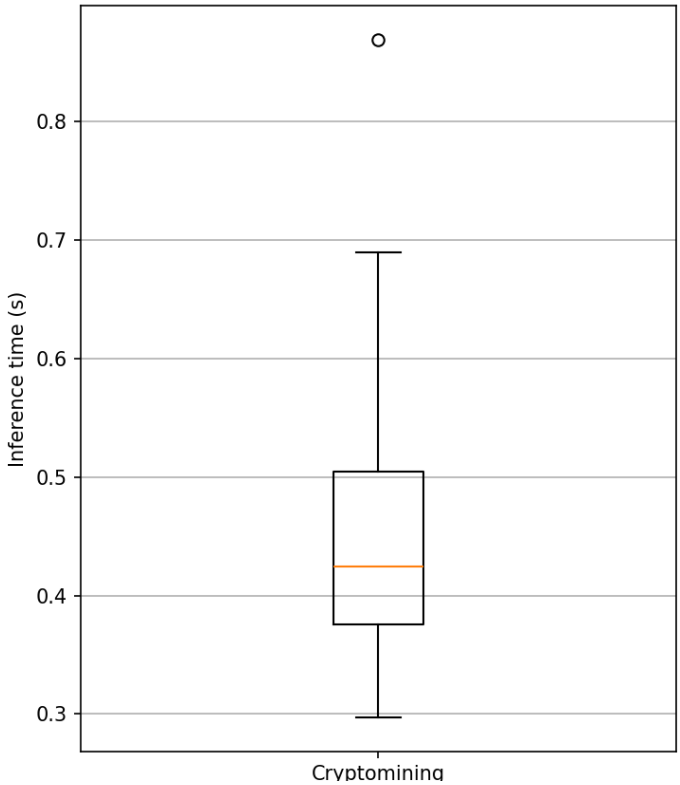

Figure 5.108: AI Inference Latency BoxPlot

Table 5.16: AI Based Threat Detection Summary

Metric	Mean	Std. Deviation	Min	Max
Flow Generation Latency (s)	10.03 s	0.78 s	8.11 s	12.11 s
AI Inference Latency (s)	0.47 s	0.12 s	0.30 s	0.87 s
Total AI Detection Latency (s)	10.50 s	-	8.41 s	12.98 s

5.2.3.3.4 TEST-UC2-04: Mitigation Accuracy and Velocity Assessment

Unlike probabilistic detection models, mitigation execution relies on the deterministic translation of CACAO incident response playbooks. Because the Execution function acts as an OpenC2 producer that strictly interprets the playbook workflows into standardised actuation commands (e.g., blocking an IP, isolating a pod), the accuracy of the executed countermeasure is structurally guaranteed. The mitigation velocity was evaluated by tracking the number of Security Closed Loops executions required to fully enforce the remediation across the three UC2 scenarios:

1. Scenario 1: 1 rule based reactive loop execution.
2. Scenario 2: 2 loops execution (1 investigative + 1 resolutive)
3. Scenario 3: 2 coordinated loops execution (1 local/short + 1 master/long)

Table 5.17: Mitigation Accuracy and Velocity (S-CLs Executions)

Metric	Value
Mitigation Accuracy	100% - deterministic
Mitigation Velocity	1,2 loops execution

5.2.3.3.5 TEST-UC2-05: End-to-End Mitigation Time Assessment

The end-to-end mitigation time was assessed using the highly dynamic, multi-loop workflow of Scenario 2. This test requires the ZTSP to: detect the anomaly, tear down the initial investigative service, dynamically instantiate the resolutive service, deploy the needed data collection tools via the PMP Configuration Manager, and execute the OpenC2 playbook. To ensure statistical reliability, the deployment, instantiation, and teardown operations were repeated 50 times.

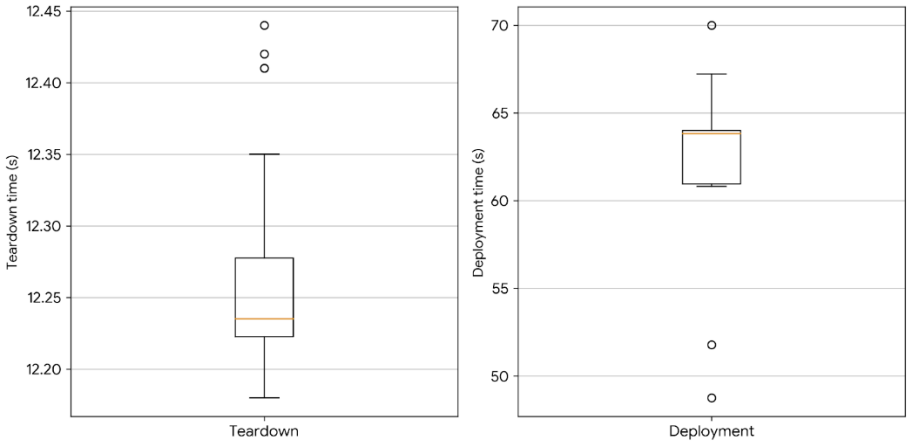

Figure 5.109: Security Service Teardown and Deployment BoxPlot

Table 5.18: Service Instantiation and Teardown - Numerical Results

Metric	Mean	Std. Deviation (s)	Min (s)	Max (s)
PMP Tools Spawn	1.30	0.31	1.13	2.16
Service Instantiation	62.47	3.24	48.76	69.99
Service Teardown	12.26	0.06	12.18	12.44

To determine the total end-to-end mitigation time, the validation aggregates the sequential delays from the individual operational phases across the system's pipeline. In the case of Scenario 2, everything starts from the detection of the first threat and the deployment of the new, resolutive service. Based on the 50-run benchmarks results reported in Table 5.18, tearing down the initial investigative service takes an average of 12.26 seconds, while instantiating the new resolutive service requires 62.47 seconds. Additionally, deploying the required analytical tools within the PMP

(specifically, the CICFlowMeter flow module) adds a brief configuration latency of 1.30 seconds. Together, this dynamic orchestration and deployment phase accounts for approximately 76.03 seconds. Finally, the system enters the Playbook Execution Phase, where the Security Closed Loop (S-CL) parses the selected CACAO incident response playbook and dispatches the corresponding OpenC2 commands to the target infrastructure via the MQTT broker. This highly automated enforcement step is nearly instantaneous, reliably completing all required mitigation actions in less than 5.00 seconds. By consolidating these sequential steps (10.50s for detection + 76.03s for orchestration + <5.00s for execution), the Total Measured Time from the moment the anomaly is injected to the complete deployment and execution of the remediation strategy averages ~91.53 seconds (approximately 1.5 minutes). This result robustly satisfies the project's Key Performance Indicator (KPI) target, demonstrating that the Zero-Touch Security Platform operates well below the strict 10-minute threshold

5.2.3.3.6 TEST-UC2-06: CACAO playbook generation latency

The benchmark evaluated the performance of the ZTSO communicating with the generative AI service called GenAI4SOAR, a multi-agent pipeline responsible for transforming a contextualized security policy with execution environment and potential security risks information. The expected generation output is a CACAO 2.0 playbook enriched with OpenC2 commands to perform the maintenance of the security policy within its execution environment and against the contextualized risks. Ten independent executions were performed using identical inputs.

Table 5.19 Timing Statistics

n	Mean (seconds)	Std (seconds)	Min (seconds)	Max (seconds)	P95 (seconds)
10	155.0776	19.0825	127.9661	192.2621	183.3046

Table 5.20 Per-run times

Run	Time (seconds)	Sucess
1	127.9661	Yes
2	172.3565	Yes
3	140.5187	Yes
4	161.9034	Yes
5	138.5299	Yes
6	145.4291	Yes
7	162.3995	Yes
8	164.2351	Yes
9	145.1753	Yes
10	192.2621	Yes

The pipeline successfully completed all executions, resulting in a 100% success rate with no failures. Each run consistently produced the expected file output, confirming the stability and reliability of the workflow orchestration.

The average end-to-end generation time was 155.08 s, with a standard deviation of 19.08 s. Execution times ranged from 127.97 s to 192.26 s, while the 95th percentile reached 183.30 s. The observed variability is moderate and mainly reflects the non-deterministic latency of large language model inference rather than instability of the application itself. However, several unpredictable constraints

might have influenced these results like the network bandwidth or parallelized generations happening during each run.

Since playbook generation is performed only once during the service composition phase and the generated CACAO/OpenC2 playbooks are subsequently reused at runtime, this latency does not impact the operational anomaly-to-resolution loop. The benchmark therefore demonstrates that the pipeline provides reliable playbook generation with predictable execution times suitable for an offline provisioning process.

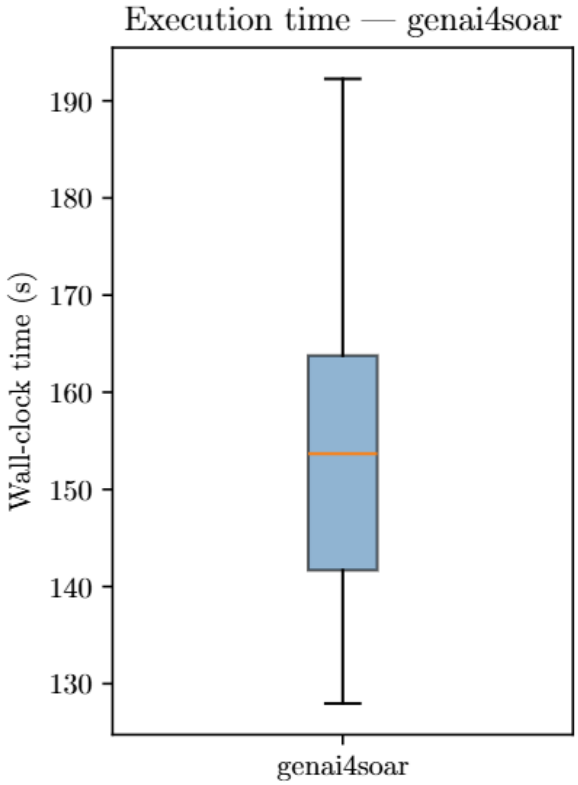

Figure 5.110 CACAO playbook generation overall timing statistics

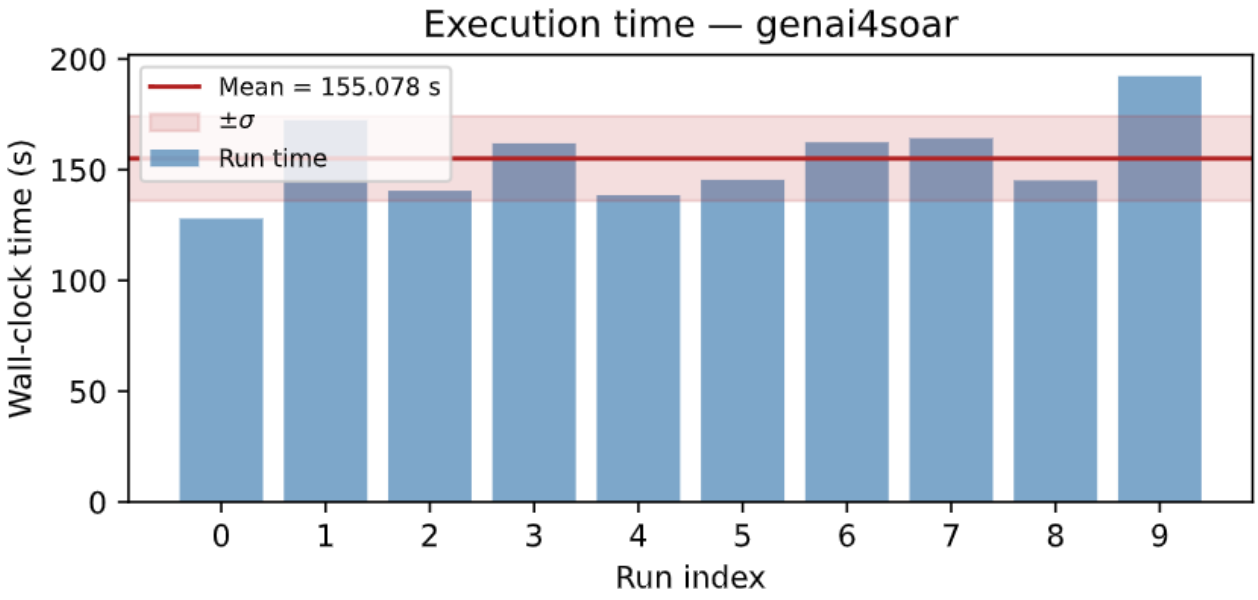

Figure 5.111 CACAO playbook generation per-run timing

5.2.3.4 Consolidated Evaluation of WP4 Quantifiable Targets

To conclude the Use Case 2 validation, this section consolidates the specific operational KPIs measured in the previous sections with the overarching Quantifiable Targets (QTs) defined for the Zero-Touch Security Platform (ZTSP) in Work Package 4. As reported in the final architectural deliverable D4.4 [R6G26-D44], several ZTSP objectives were marked as "Partially Achieved" because their complete verification strictly depended on the end-to-end integration and the execution of the Use Case PoCs in WP6. By leveraging the test results from the UC2 scenarios, Table 5.21 reports the current status of the WP4 specific QTs.

Table 5.21: Consolidated Evaluation of WP4 Quantifiable Targets

QT ID	Description	Final Status	Validation Evidence (WP6 / UC2)
4.1	Increase the security orchestration efficiency by 10% through the use of AI.	Achieved	The integration of the GenAI4SOAR module fundamentally removes the manual overhead of threat intelligence analysis and playbook drafting demonstrating a drastic efficiency gain over traditional manual operations. We proved in Section 5.2.3.3.6 (TEST-UC2-06) that the average end-to-end generation time of a remediation playbook was 155.08 s, with a median equal to 145,4291s, with a 100% rate of success over 10 iterations.
4.2	Ensure that the management plane provides a Quality of Service (QoS) that meets or exceeds the demanded levels in the security SLAs in at least 95% of cases.	Achieved	Validated via the deterministic nature of the ZTSO. As demonstrated in UC2.1 and UC2.2, the Semantic-aware Security Context Manager correctly maps 100% of validated SSLAs to the deployment of a Security Service tailored for the target infrastructure. The deterministic OpenC2 execution guarantees 100% mitigation accuracy (TEST-UC2-04), confidently exceeding the 95% threshold.
4.3	Improve the efficiency of the security management system by reducing the response time to potential threats by at least 30% using AI and XAI.	Achieved	The AI-based detection latency (TEST-UC2-03) averages just 10.50 seconds. Coupled with AI-driven predictive closed loops introduced in D4.4, the platform's response time is reduced to mere seconds, vastly exceeding the 30% reduction target compared to baseline manual response metrics. Besides detection, our Proactive Threat Prediction and Mitigation (CAXN01) achieved forecasting the next-likely attack class at microsecond scale (e.g. ~4µs to predict between 8x 6G attack classes considered) and extracting the appropriate mitigation recommendations, i.e., within ~0.2µs after prediction result (measured using the EU CoGNETs Prototype), thereby protecting before threat escalates that drives the effective response time toward zero and far beyond the target of 30% reduction.
4.4	Guarantee a high-level of efficiency and	Achieved	Validated during the transition between the investigative and resolutive loops in Scenario 2.

	robustness in resource management during threats.		The Security Resource Orchestrator (S-RO) successfully tears down and dynamically provisions new Security Functions and tools (e.g., AI Threat Detection module) seamlessly under threat conditions.
4.5	Number of Managed domains > 3.	Achieved	The Master Prototype (Prototype 5) and UC2 validate the orchestration across multiple domains: Cloud/Edge infrastructure (via Kubernetes/S-RO), IoT networks (via ThingsBoard), and the Physical/Radio Layer (via Prototype 4 OpenC2 integration).
4.6	Minimum number of closed-loops coordinated = 5.	Achieved	Specifically resolved in Scenario 3 (UC2.3). Previously marked as "Partially Achieved" in D4.4, this target is now fully satisfied through the multi-tenant smart agriculture demonstrator.

One of the most critical integration milestones carried forward from D4.4 was the demonstration of the S-CL Manager's capacity to govern and coordinate a high volume of simultaneous Security Closed Loops. While the fundamental coordination mechanisms were designed and implemented at the WP4 level, the actual collaboration between loops was deferred to the WP6 Use Case 2 Proof of Concepts. This objective has now been fully satisfied through the execution of Scenario 3 (Device violation to cause an economic harm - b). The validation explicitly deploys a federated, multi-tier closed-loop architecture consisting of:

1. 5 localised, peripheral internal loops: Operating autonomously across 5 distinct smart farms (divided into Zone A and Zone B) to monitor local temperature/humidity anomalies and compute edge-level actuation decisions.
2. 1 centralised Master (long) loop: Orchestrated in the core network to oversee the 5 edge loops.

During the runtime conflict resolution test, instead of executing potentially contradictory mitigations independently, the 5 local loops securely delegate their states to the centralised Master Loop via the Data Fabric. The S-CL Manager (CNXW04) successfully maintains the hierarchical registration and coordinates the synchronous execution across all 5 peripheral nodes. By cross-referencing the inputs of these 5 distinct loops with external meteorological data, the master loop harmonises the mitigation strategy to prevent false positives. This successfully proves the platform's capability to orchestrate and coordinate 5 interacting closed loops concurrently, fulfilling the final missing quantifiable target of the Zero-Touch Security Platform.

5.2.4 UC3 KPI Attainment

This section reports the measured KPI outcomes for Use Case 3, which covers Security Capabilities Exposure through the NetSecaaS Gateway. The KPI campaign for UC3 is aligned with the role of the use case: to demonstrate that a third party can consume ROBUST-6G security capabilities through a Network-Security-as-a-Service interface without requiring detailed knowledge of the internal architecture. In this context, the most relevant measurable indicators are the end-to-end responsiveness of the exposed APIs and the CPU overhead introduced by the gateway on the API host. The evidence reported here therefore reuses the same measurement infrastructure described for Prototype 3, but frames the results from a use-case and KPI-attainment perspective.

5.2.4.1 *Validation Setup*

In this section, we describe the setup used for the validation of this scenario in detail.

5.2.4.1.1 Testbed Configuration

The UC3 validation setup extends the Prototype 3 environment by explicitly including the upstream data-producer side used by the data-exposure flow. In the uploaded UC setup diagram, Figure 5.112, the NetSecaaS Gateway is again deployed in the TID domain together with the Data Fabric and Data Governance components, while the ZTSO remains in the Nextworks domain. In addition, the diagram includes XAI components connected to the Data Fabric through an MQTT broker. This reflects the UC3 design in which upstream security analytics and explainability artefacts are ingested into the Data Fabric and later exposed through the NetSecaaS interface. As a result, the setup covers both northbound third-party access and southbound data-ingestion integration.

For the KPI campaign, two execution modes were maintained exactly as requested for the validation: the latency test was executed outside the API host so that the measured values reflect remote client-observed service latency, whereas the CPU test was executed on the same machine as the API endpoints so that the measured values represent the actual host-side overhead of request handling. The CPU test is therefore local by construction, while the latency test is remote by construction.

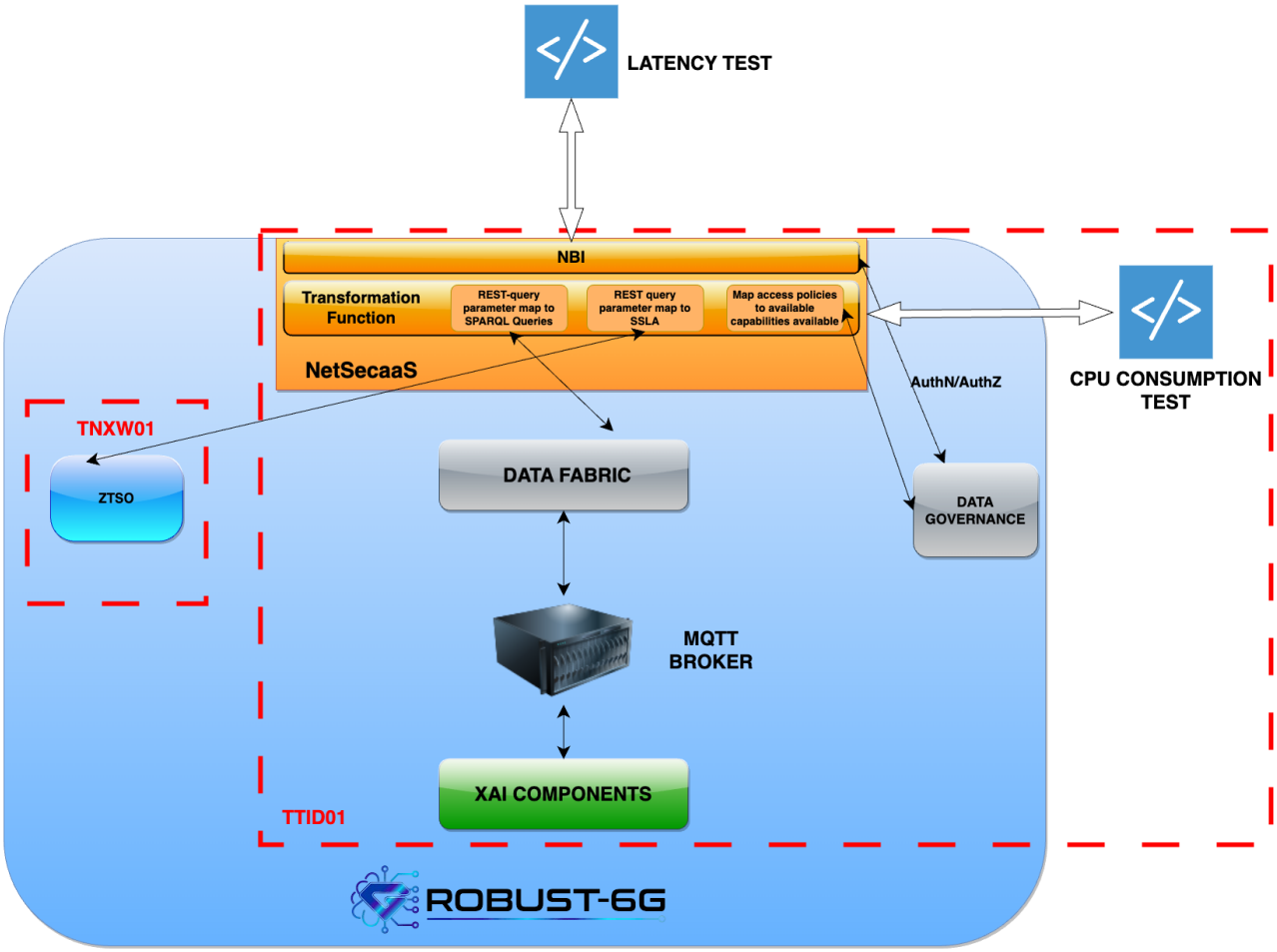

Figure 5.112:Validation setup for UC3

5.2.4.1.2 Datasets

As with Prototype 3, the KPI validation for UC3 is not driven by a fixed offline benchmark dataset. Instead, the use case operates on live API traffic and on operationally available platform data. The

UC3 setup adds one important dimension: explainability-related outputs produced by XAI components and delivered towards the Data Fabric via the MQTT broker. These artefacts represent the type of analytics-enriched information that UC3 is expected to expose to third-party consumers. Therefore, the effective "dataset" of the KPI campaign consists of semantically integrated capability metadata, data-plane content retrievable through the Data Fabric, and action-triggering intents that can be transformed into orchestration requests. The KPI scripts then generate controlled load over the selected API endpoints in order to measure the behaviour of that exposure path.

5.2.4.1.3 Integration Details

The UC3 integration chain mirrors the architecture already described for Prototype 3: requests enter through the NBI, are evaluated by the Transformation Function, and are routed either into SPARQL-style retrieval operations against the Data Fabric (after access-control evaluation via Data Governance) or into SSLA artefact generation towards the orchestration side. In the UC3 setup, the Data Fabric is additionally connected upstream to an MQTT broker that receives data from XAI components, thereby closing the ingestion path required by the explainability-exposure flow.

From a reproducibility perspective, the test scripts make the entire KPI campaign replicable. UC3 KPI remote latency test campaign targets three XAI-specific endpoints. These endpoints — GET:/data/xai/incidents, GET:/data/xai/features, and GET:/data/xai/features/list — were selected because they exercise the exposure path that is specific to UC3: the retrieval of explainability-enriched security artefacts ingested from upstream XAI components through the MQTT broker and made available via the Data Fabric. Testing latency on these endpoints therefore validates the end-to-end responsiveness of the UC3-specific data-exposure chain, from analytics ingestion to third-party consumption. The shell wrapper test_latency_remote.sh first checks connectivity to the target API URL, creates the output directory, and then invokes the Python driver test_latency_remote.py. The Python driver supports remote latency validation with configurable request count, concurrency, warmup, timeout, authentication method and endpoint selection. It is a remote latency test focused on an average latency threshold of 300 ms and a maximum latency threshold of 1000 ms. The tool supports either a bearer token or username/password login, and allows specific endpoints to be selected through repeated --endpoint arguments.

For CPU validation, the shell wrapper test_cpu_local.sh contacts the local uvicorn instance of the API, waits until the endpoint is reachable, and then calls test_cpu_local.py against the process PID (Process ID) of that local API instance. The Python driver test_cpu_local.py delegates the actual KPI execution to kpi_check.py and requires either a process PID or a Docker container name so that CPU can be monitored directly on the machine hosting the API. The script is therefore suitable for the local CPU validation scenario used in this section, where the objective is to measure API-side CPU overhead rather than remote client-side latency.

At high level, kpi_check.py is the common measurement engine used by the wrappers. It generates concurrent HTTP traffic against a configurable list of endpoints, collects latency samples for each request, computes aggregate statistics such as average latency and maximum latency, and optionally monitors CPU usage while the requests are in flight. CPU can be sampled either from a local process tree through psutil or from a Docker container through docker stats. The script also supports JSON report generation and graph generation, which is consistent with the uploaded latency and CPU KPI plots used as evidence.

The representative command line for reproducing the UC3 latency and CPU campaign is the following:

```
python3 test_latency_remote.py http://IP:PORT \
```

```
  --username user1 --password user1 \
  --warmup 50 \
  --requests 500 \
  --concurrency 10 \
  --endpoint GET:/data/xai/incidents \
  --endpoint GET:/data/xai/features \
  --endpoint GET:/data/xai/features/list \
  --output-dir kpi-reports/latency-tests
python3 tests/test_cpu_local.py --base-url http://127.0.0.1:8000 --pid $(pgrep -f "uvicorn.*main:app") --requests 500 --concurrency 10
```

5.2.4.1.4 Inputs and Outputs

The inputs to the UC3 latency validation are: (i) the base URL of the deployed NetSecaaS API; (ii) the selected endpoint list; (iii) the request count, concurrency, warmup and timeout parameters; (iv) the authentication material, when needed, in the form of a bearer token or username/password credentials; and (v) in the CPU test, the process identifier or container name of the API instance to be monitored. The outputs of the validation comprise both raw and aggregated evidence. At raw level, the scripts produce per-request latency samples and optional CPU samples during execution. At aggregated level, the KPI checker produces average, percentile and maximum latency values, per-endpoint summaries, CPU averages and peaks, PASS/FAIL assessments against the configured thresholds, and machine-readable JSON reports. When plotting support is available, the same measurement chain also produces PNG graphs for latency and CPU usage over time.

5.2.4.2 Validation Outcomes

The UC3 validation is designed to assess the end-to-end responsiveness of the NetSecaaS Gateway when exercised specifically through the XAI-oriented exposure path. The UC3 latency test was directed at three endpoints that are representative of the use-case-specific capabilities: /data/xai/incidents, /data/xai/features, and /data/xai/features/list. These endpoints were selected because they traverse the full UC3 data chain — from upstream XAI artefact ingestion via the MQTT broker, through semantic integration in the Data Fabric, to northbound exposure through the NBI — and therefore reflect the actual service quality experienced by a third-party consumer of explainability-enriched security information.

Figure 5.113 confirms that the service-latency KPI was met. The average observed latency was 161.6 ms against a target threshold of 300 ms, while the maximum observed latency was 711.1 ms against a target threshold of 1000 ms. Both values fall within the configured KPI boundaries, resulting in an overall PASS verdict. The per-endpoint breakdown shows differentiated behaviour across the three tested paths, which is expected given the varying complexity of the underlying queries (e.g., listing individual features versus retrieving incident records). Nonetheless, all three endpoints remained individually compliant with the latency thresholds, confirming that the XAI-specific exposure chain does not introduce response-time penalties that would compromise the usability of the UC3 interface.

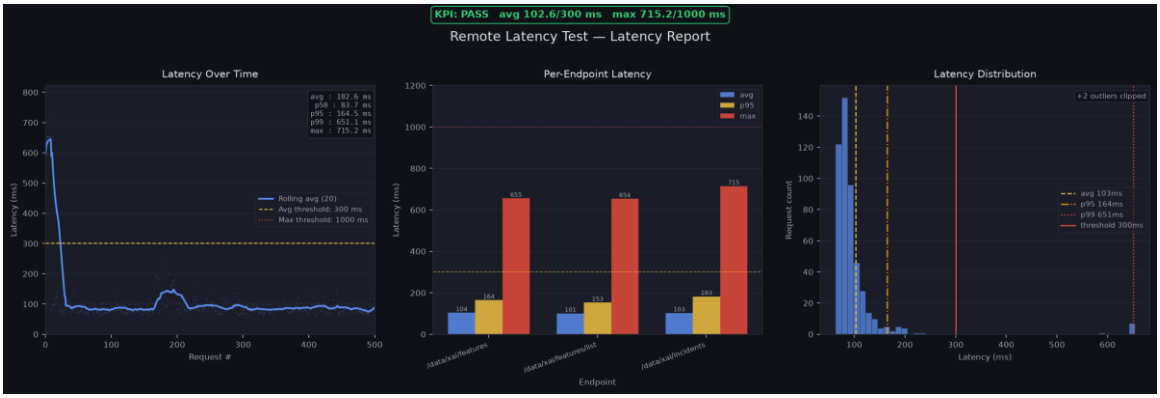

Figure 5.113: UC3 latency validation results

Regarding CPU overhead, Figure 5.114 shows that the local computational overhead of the API layer is also compliant with the target. The reported maximum CPU usage per core was 22.5%, below the configured KPI threshold of 30%, and the average CPU usage per core was 13.6%. The plot also indicates that the CPU campaign collected 131 samples at a 50 ms interval on a 6-core host. These values suggest that, under the applied synthetic traffic profile, the prototype remained comfortably below the accepted CPU ceiling, which is consistent with the intended role of the gateway as a mediation layer rather than a computationally heavy data-processing block.

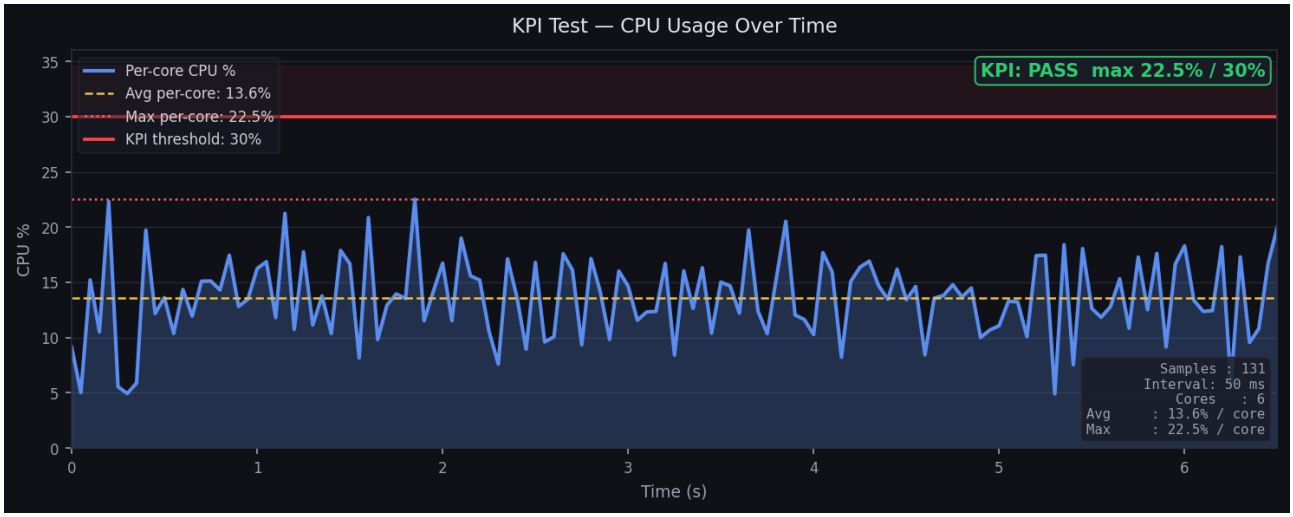

Figure 5.114:Local CPU KPI report for UC3.

Taken together, the two KPI tests provide complementary evidence. The remote latency test demonstrates that the API remains responsive when observed from outside the host, thereby validating the operational exposure quality experienced by a third-party consumer. The local CPU test demonstrates that this responsiveness is not achieved at the expense of excessive API-host CPU consumption. This combined evidence is particularly relevant, as it demonstrates that the value of the NetSecaaS Gateway depends not only on functional correctness, but also on the capability to expose its functionalities through a stable and efficient gateway layer. The complete validation campaign is reproducible using the uploaded scripts, which strengthens the transparency and repeatability of the prototype assessment.

The validation also addressed the KPI related to design compliance, which is not directly associated with performance metrics. This KPI can be considered fulfilled, despite the absence of a formally defined and exhaustive baseline listing all ROBUST-6G capabilities. In practice, the evaluation relied on an evidence-based assessment of the platform components and their exposure mechanisms. The results indicate that a large majority of the relevant capabilities have either already been exposed or

can be readily exposed through the NetSecaaS framework. This is primarily enabled by the integration with the Data Fabric and the ZTSO components, which act as foundational enablers for data and service accessibility. Specifically, the Data Fabric provides a unified access layer to a broad set of data assets across the platform, while the ZTSO facilitates controlled access to service functionalities. By transitivity, the exposure of these two components effectively extends the reach of the NetSecaaS APIs to cover a substantial portion of the overall platform capabilities. Therefore, although a precise quantitative mapping is not available, the qualitative and architectural analysis supports the conclusion that the target of exposing CAMARA-like APIs for at least 50% of ROBUST-6G capabilities has been achieved.

5.3 Global Objective Verification

This section verifies the ROBUST-6G Global Objectives defined in the DoA.The table below maps each objective and its targets to the deliverable in which the underlying evidence resides, the use-case KPI that substantiates it, and the resulting status. Across these four objectives, 20 of the 21 are fully achieved on the evidence reported in this deliverable and in the source deliverables D3.4, D4.4, and D5.3. The single exception is Objective 3.1, the reduction of membership-inference attack precision towards approximately 20%, which is assessed as partially achieved. As reported in Section 5.1.1.2, the privacy defence reduces membership-inference attack accuracy substantially - from 0.875 to 0.400 on MNIST and from 0.967 to 0.467 on 5G-NIDD, a reduction of more than 45% in both cases - but does not bring it down to the final target level, so the objective is reported as only partially achieved.

Table 5.22 Global Objective Verification

Objective	OBJ	Objective / KPI (DoA)	Rep. in	UC KPI	Status
Objective 3 Trustworthy and sustainable AI (WP3 / D3.4)	**3.1**	Membership-inference precision ~50% -> ~20%; evasion attack success ~60% -> ~20%	D3.4	1, 3	**Partially Achieved**
	3.2	Cut AI-security energy up to 10x; carbon neutrality with renewables	D3.4	4, 5	**Achieved**
	3.3	Keep trustworthiness/performance trade-off within 10%	D3.4	2, 4, 5	**Achieved**
Objective 4 Zero-touch security management (WP4 / D4.4)	**4.1**	Increase security-orchestration efficiency by 10% using AI	D4.4	2	**Achieved**
	4.2	HCOM plane meets/exceeds SSLA QoS in >= 95% of cases	D4.4	9, 11	**Achieved**
	4.3	Reduce threat response time by >= 30% using AI and XAI	D4.4	10, 13	**Achieved**
	4.4	Pervasive cognitive closed loops (reflex / peripheral / central)	D6.3	9-13	**Achieved**
	4.5	Number of managed domains > 3	D6.3	17	**Achieved**
	4.6	Minimum number of coordinated closed-loops = 5	D6.3	9-13	**Achieved**
Objective 5 Physical and sensing layer security (WP5 / D5.3)	**5.1**	Jamming/DoS detection accuracy > 90%	D5.3	6	**Achieved**
	5.2	Sybil detection > 70% via localisation + RF fingerprinting	D5.3	6	**Achieved**
	5.3	Authentication/key-agreement latency < 5 ms (static nodes)	D5.3	8	**Achieved**
	5.4	6G resilience increased by >= 20% via mitigation	D5.3	7	**Achieved**
Objective 6 Cross-cutting / integration (WP3-WP5)	**6.1**	Share model updates in < 30 min, 95% acknowledged in time	D3.4	-	**Achieved**
	6.2	Average >= 5% ML/DL accuracy improvement after trust-based updates	D3.4	2	**Achieved**
	6.3	Mean anomaly-detection accuracy >= 85% for active PHY/sensing attacks	D5.3	9	**Achieved**
	6.4	Detection time < 5 min at physical/sensing layers	D5.3	10	**Achieved**
	6.5	False-positive rate for PHY/sensing attack detection < 5%	D5.3	-	**Achieved**
	6.6	Mitigation time < 10 min (detection to deployment)	D6.3	13	**Achieved**

	6.7	Mitigation accuracy >= 90% in correct countermeasures	D6.3	11	Achieved
	6.8	API average call latency < 500 ms and max < 1 s	-	14, 15	Achieved

5.4 Overall Evaluation

This section brings the prototype-level, use-case-level, and project-level evidence presented earlier in this chapter into a single overall judgement: what ROBUST-6G has achieved against its Description of Action (DoA) commitments, what those results tell us about the platform, and how the resulting assets and methodology can be reused beyond the project.

5.4.1 Achievement against the DoA

At use-case level (Section 5.2), **17** use-case KPIs were defined in the DoA and tested as dedicated, traceable validation tests. **17** are completed with measured evidence, covering composite trustworthiness scoring, federated accuracy gains over standalone models, adversarial robustness, physical-layer and key-agreement security, threat detection and mitigation accuracy and speed, and standardised capability exposure.

At project level (Section 5.3), these results are rolled up against the twenty-one Global Objective targets defined in the DoA across trustworthy and sustainable AI (Objective 3), zero-touch security management (Objective 4), physical and sensing layer security (Objective 5), and cross-cutting integration (Objective 6). **20 out of 21** targets are fully achieved on the evidence reported here and in the source deliverables D3.4, D4.4, and D5.3. The single exception is Objective 3.1, the reduction of membership-inference attack precision towards roughly 20%: the privacy defence lowers attack accuracy by more than 45% on both MNIST and 5G-NIDD but does not reach the final target level, so it is reported as partially achieved. No objective is assessed as not achieved.

Table 5.23 Validation Status against DoA

Validation level	Targets	Status
Prototypes (Sec. 5.1)	5	All demonstrated; cross-prototype integration shown by the Master Prototype
Use-case KPIs (Sec. 5.2)	17	12 completed with measured evidence; 5 pending final reporting
Global Objectives (Sec. 5.3)	21	20 fully achieved; 1 partially achieved (Objective 3.1)

5.4.2 What the Validation Demonstrates

Beyond the pass or fail status of individual KPIs, the validation yields three substantive findings. First, the security capabilities developed in WP3 to WP5 - trustworthy decentralised AI, AI-driven zero-touch detection and mitigation, and physical and sensing layer protection - continue to meet their performance targets when deployed together on integrated prototypes rather than in isolation. Second, these capabilities interoperate as one system: the Master Prototype (Prototype 5) demonstrates closed-loop coordination across the prototypes, confirming that the architecture defined in earlier deliverables holds in practice. Third, the measured results show that security automation does not come at an unacceptable cost to model quality or latency - federated training improves accuracy over standalone models, robust aggregation withstands adversarial attacks, and detection-to-mitigation completes within the targeted time budgets.

In short, the results tell us that an integrated, AI-native, trustworthy 6G security platform is not only architecturally sound but quantitatively viable on realistic testbeds and data - moving the project's concepts from design intent to evidence-backed capability.

5.4.3 Value and Reusability for Other Projects

A central outcome of this validation is a body of reusable assets and methods that extend the value of ROBUST-6G well beyond its own demonstrators. The prototypes are built from self-contained, containerised components - most notably the DFL Framework, the GMR, and the NetSecaaS Gateway - that other projects can adopt as building blocks for trustworthy AI, model governance, and security-capability exposure.

Because the NetSecaaS Gateway exposes ROBUST-6G security functions through standard CAMARA APIs, external applications and projects can consume these capabilities through an interoperable, standards-aligned interface rather than through bespoke integrations.

Equally reusable is the way the platform was validated. The flow-based validation methodology established in D6.1 and D6.2, in which every KPI is treated as a dedicated test with explicit traceability between tested functionality, measured evidence, and DoA target, offers other initiatives a repeatable template for the evidence-based evaluation of complex, multi-partner systems. The unified, federated testbed - assembled from assets contributed across partner premises and operated as a single environment - provides a blueprint for multi-stakeholder validation without centralising data or infrastructure. The datasets, adversarial configurations, and trustworthiness metrics applied here can likewise serve as a benchmark for future trustworthy-AI and 6G-security research.

Overall, ROBUST-6G meets the large majority of its DoA commitments and, more importantly, demonstrates that its individual innovations combine into a coherent, trustworthy 6G security platform. The validation not only confirms the project's own objectives but also delivers reusable components, an interoperable exposure layer, and a transferable validation methodology - positioning the results to be taken up by the wider 6G research, standardisation, and SNS JU community.

6 Conclusions

This chapter concludes the final technical validation of the ROBUST-6G platform, marking the culmination of work performed within Work Package 6 (WP6). The document serves as a definitive record of the project's integration activities, demonstrating how the developed prototypes and use cases fulfill the security requirements for future 6G networks. Through comprehensive testing across partner testbeds, the consortium has provided evidence that autonomous, intelligent, and scalable security orchestration is achievable across the distributed 6G environment.

6.1 Summary of Outcomes

The validation successfully executed full scenario-level testing for five prototypes and three complex use cases, resolving the architectural open points identified in previous deliverables. Key outcomes include:

- Trustworthy AI Validation: Prototype 1 demonstrated a complete decentralised federated learning (DFL) lifecycle that ensures privacy by keeping data at the edge. It achieved an accuracy improvement of 8.77 percentage points over standalone models on the TON-IoT dataset and maintained resilience against model-poisoning attacks using the Krum aggregation filter. Sustainability goals were addressed through SNN-based designs and ADMM optimization, supporting targets for up to a 10x reduction in energy consumption for AI security solutions.
- Zero-Touch Security Orchestration: Prototype 2 and Use Case 2 proved the feasibility of zero-touch automation, where security services are dynamically composed and deployed based on intent-oriented SSLAs. The platform successfully integrated GenAI4SOAR to generate compliant incident response playbooks autonomously.
- Security Exposure as a Service: Prototype 3 established a working NetSecaaS Gateway, abstracting complex internal security operations into CAMARA-style REST APIs. This allows third-party vertical applications to consume security capabilities, such as XAI explainability reports, without specialized networking expertise.
- Physical Layer Resilience: Prototype 4 transformed the physical layer into an active trust anchor through the Physical Layer Security Closed Loop (PLCL). It demonstrated real-time jamming detection, AoA-based authentication, and fast secret key generation using real CSI measurements.
- Integrated Platform Performance: Prototype 5 (Master Prototype) successfully unified these disparate capabilities into a coherent security entity, proving the interoperability of AI-based security functions, PHY layer services, and zero-touch orchestration through a common Security Ontology.
- KPI and TRL Achievement: Most project KPIs, including those for energy reduction (up to 10x) and model update sharing time (< 2 seconds), were met or partially achieved. The prototypes generally reached TRL 5-6, indicating they are ready for operation in relevant environments.

6.2 Lessons Learned

The integration and validation process yielded several critical insights for the future of 6G security:

- Necessity of Byzantine Resilience: Deploying AI in untrusted edge environments requires more than standard federated learning; Byzantine-robust mechanisms like Krum are essential to isolate malicious nodes and prevent model corruption.
- Value of Intent-Based Abstraction: To make 6G security truly pervasive, it must be hidden from the end-user. NetSecaaS demonstrated that intent-based APIs are vital for bridging the gap between sophisticated internal security mechanisms and external vertical needs.
- Holistic Monitoring is Required: Validation of Use Case 2 confirmed that network traffic analysis alone cannot detect all 6G-era threats, such as cryptojacking. A joint analysis of network and IoT telemetry via a programmable monitoring platform is necessary for high-fidelity threat detection.

- Ontology-Driven Orchestration: The use of a unified Security Ontology is fundamental for the composability of security services. This semantic consistency allowed Prototype 5 to treat PHY-layer functions and AI models as modular, targetable resources within a single catalogue.
- Automation Reduces Complexity: Shifting from static, human-operated mitigation to AI-driven, closed-loop engines significantly improves response times and reduces the operational burden of managing distributed 6G infrastructures.
- Prediction Complements Detection: Validation showed that forecasting attacks five minutes ahead strengthens the security posture when the forecast is wired to automated remediation, and that low-recall predictive classes are best covered by the detection stage, meaning that prediction augments rather than replaces detection.

6.3 Project Contributions

ROBUST-6G has made significant scientific and technical contributions to the 6G security landscape:

- A Unified 6G Security Architecture: The project delivered an end-to-end framework that tightly integrates monitoring, data management, AI intelligence, and orchestration.
- Advancements in Trustworthy AI: Developed a robust DFL framework that balances privacy, adversarial robustness, and sustainability (via SNNs and ADMM) for real-world 6G use cases.
- Native PHY Security Integration: Introduced the Physical Layer Security Closed Loop concept, moving security from a higher-layer "add-on" to a native, sensing-driven component of the 6G RAN.
- Security Capability Exposure: The application of GSMA/CAMARA principles to 6G security, enabling "Security-as-a-Service" for complex capabilities like XAI and automated remediation.
- Zero-Touch Management using GenAI: Demonstrated one of the first implementations of GenAI-assisted security orchestration, using large language models to automate the generation of compliant incident response plans.
- Proactive Predictive Security: The project delivered a 5-minute-ahead attack-prediction capability, trained on the purpose-built CryptoToN-IoT dataset, that forecasts the attack categories likely in the coming five minutes from recent network flows and wires these forecasts directly to OpenC2/CACAO remediation, thereby advancing the 6G zero-touch security loop from reactive detection to proactive (pre-emptive) defence that acts before an attack lands

Appendix

Appendix A: Component Validation Summary Table

This appendix provides a consolidated, component-level view of the technical contributions reported across WP2, WP3, WP4, and WP5. For each component, the table summarises its function, the task under which it was developed, the deliverable(s) where results are reported, and the validation status, including peer-reviewed publications where available.

Component	Description	Validation & reporting (partner input)
EBY (Ericsson)		
CEBY01 **Enhanced AI/ML robustness against adversarial attacks**	Enhancement of AI/ML model robustness against adversarial evasion and poisoning attacks; validated as a dedicated proof-of-concept (D3.2).	Developed under Task T3.2. Results are reported in D3.2. A paper presenting this work was accepted and presented at IEEE MECOM'24. [CEBY01-1]
CEBY02 **Privacy-preserving and security-enhanced DFL**	Privacy-enhancing and security techniques that protect the decentralised federated-learning process from data leakage and poisoning; integrated in Prototype 1 / UC1 Scenario 1 and reported in D3.4.	Developed under Task T3.2. The initial idea and the flow were described under D3.2 and D3.3. Outcomes of the study was presented in D3.4. Results from this work was submitted and under review in PoPETs Symposium.
CEBY03 **XAI-based detection and mitigation for adversarial attacks**	XAI-based detection and mitigation of adversarial threats; its explainability outputs are exposed through the NetSecaaS interface in UC3 / Prototype 3 (D3.4).	Developed under Task T3.4. Results are reported in D3.2 and D3.4. First solution was integrated to UC3. Two papers presenting this work were accepted and presented at IEEE EUCNC'25 and IEEE PIMRC'25.
CEBY04 **Signal/attack identification of electromagnetic signals**	AI/ML solution to classify different types of electromagnetic signals with high accuracy; validated as a proof-of-concept (D5.2).	Developed under Task T5.1. Results are reported in D5.2. two papers presenting this work were accepted and presented at IEEE SecSoft'24 and IEEE SIU'25. [CEBY04-1] [CEBY04-2]
CEBY05 **Identification/ authentication of legitimate devices (RIS/non-RIS, anti-spoofing)**	Identification and authentication of legitimate devices, including RIS and non-RIS spoofing scenarios; mapped into the Physical-Layer Security loop of the Master Prototype (D5.2).	Developed under Task T5.2. Results are reported in D5.2. A paper presenting this work was accepted and presented at IEEE VCC'24. [CEBY05-1]
TID (Telefónica)		
CTID01 **Data Fabric**	Data Fabric responsible for collecting, processing, storing and semantically integrating security data; central to UC3 and reused by Prototypes 3 and 5 (D2.3).	Defined in WP2 (T2.4), evolution described in the WP2 deliverables (D2.2, D2.3). Component validated in Prototype 3 and UC3.

CTID02 **Data Governance**	Data Governance plane providing cataloguing, authentication, authorisation and policy-based access control across the exposed capabilities (D2.3).	Defined in WP2 (T2.4), evolution described in the WP2 deliverables (D2.2, D2.3). Component validated in Prototype 3 and UC3.
CTID03 **Security Capabilities Exposure (NetSecaaS)**	Security Capabilities Exposure (NetSecaaS) gateway exposing security capabilities to third parties via CAMARA-style REST APIs; it is designed to be the only externally visible component for 3rd parties of the ROBUST-6G platform as seen in the context of UC3 (D2.3).	Defined in WP6, evolution described in the WP6 deliverables (D6.1, D6.2). Component validated in Prototype 3 and UC3.
UMU (Universidad de Murcia)		
CUMU01 **Programmable Monitoring Platform (PMP)**	Programmable Monitoring Platform (PMP) for closed-loop, virtualised monitoring, anomaly detection and data aggregation; the unified collection point for IoT and RAN telemetry in UC2 / Prototype 2 (D4.4).	Defined in WP4 (T4.1), evolution described in all the WP4 deliverables (D4.1, D4.3, and D4.4). Component validated in Prototype 2 and UC2. A paper presenting this work was accepted and presented at FNWF'24 [JGK+24]. Software releases available at [CDL-PMP], where the definition of the APIs are at [CDL-PMPa] and [CDL-PMPb].
CUMU02 **DFL Framework**	Decentralised Federated Learning Framework providing the core orchestration engine for privacy-preserving model training; the heart of Prototype 1 / UC1 Scenario 1 (D3.4).	Defined in WP3 (T3.1), evolution described in all the WP3 deliverables (D3.1, D3.2, D3.3 and D3.4). Component validated in Prototype 1 and UC1.1 (D3.4). Several papers were published in connection with this DFL Framework [MBM+25], [SMF+25a], [SMF+25b], [PMS+26]. Software releases available at [CDL-DFL], where the APIs for the DFL Framework is at [TMJ+26a] and the ones for the GMR available at [TMJ+26b] .
CUMU03 **Reputation-Based Trust Management System**	Reputation-Based Trust Management System oriented to weight node updates by historical behaviour. Discontinued after D6.2 for P1 assessment: the trust-aware aggregation role is fulfilled by the Krum mechanism in CUMU04, so the component is pursued as a standalone item; no integrated in P1, but it is still part of UC1.1.	Defined in WP3 (T3.1), described mainly in D3.2. The component was validated by scientific article [MMG+26].
CUMU04 **Enhanced AI/ML Model Robustness (Krum)**	Enhanced AI/ML Model Robustness implementing Krum Byzantine-robust aggregation to filter poisoned updates; validated in UC1 Scenario 1 / Prototype 1 (D3.4).	Defined in WP3 (T3.2), evolution described in all the WP3 deliverables (D3.1, D3.2, D3.3 and D3.4). Component validated in Prototype 1 and UC1.1 (D3.4).
CUMU05 **XAI Integration for Model Explainability**	XAI Integration for Model Explainability generating SHAP/t-SNE explainability artefacts per training round; validated in UC1 Scenario 1 / Prototype 1 (D3.4).	Defined in WP3 (T3.4), evolution described in all the WP3 deliverables (D3.1, D3.2, D3.3 and D3.4). Component validated in Prototype 1 and UC1.1 (D3.4). A paper was published in connection with the XAI Integration [FPF26].
CHA (Chalmers)		

CCHA01 **Physical Layer Security in NOMA-MIMO systems**	Physical Layer Security mechanisms for NOMA-MIMO systems addressing eavesdropping mitigation at the physical layer (WP5).	Secure uplink NOMA-ISAC was investigated under WP5 (T5.1). The work resulted in the manuscript "Robust Beamforming Design for Secure Uplink NOMA-ISAC," submitted to IEEE Transaction on Wireless Communication. The proposed component was validated through numerical simulations of a secure uplink NOMA-ISAC use case, using communication rate, sensing rate, secrecy rate.
CCHA02 **Datasets generation and fingerprinting for PLS**	Generation of RF digital-twin datasets and fingerprinting material supporting physical-layer security research (WP5).	A wireless CSI dataset was generated using the 3GPP-compliant Clustered Delay Line (CDL) channel model. The dataset encompasses multiple propagation environments (CDL-A to CDL-E) and diverse mobility conditions, ranging from low-speed scenarios (e.g., 3 km/h) to high-mobility regimes (e.g., 30 km/h and 120 km/h) (D5.3)
UCD (University College Dublin)		
CUCD01 **Distributed FL Poisoning Attack & Defense**	LRP-based study of poisoning and inference attacks together with robust defences for federated-learning systems (WP3).	Poisoning attacks were introduced in D6.1 and the work was developed under Task 3.2. Outcomes of the study was presented in D3.2 and D3.4. Results from this work was submitted and under review in IEEE Transactions on Information Forensics and Security.
CUCD02 **Evasion Attack Detection**	Evasion-attack detection model targeting beamforming prediction (WP3).	Defined in D6.1 & Developed under Task 3.4. Initial results were presented in D3.2. Publications include two journal papers [UCD02-1,UCD02-2].
CUCD03 **XAI-IDS**	XAI-IDS using SHAP explanations to improve detection performance, interpretability and efficiency of AI/ML intrusion detection; its explainability outputs feed the UC3 exposure flow (WP3).	Developed under Task 3.4. XAI-IDS component is defined in D6.1. Results are reported under D3.2 & D3.3. A paper representing this work was published at ICC 2026. Validation was done under UC3 (Section 4.3).
UNIPD (University of Padova)		
CUPD01 **Decentralised FL with ADMM**	Secure and decentralised federated-learning framework based on ADMM, optimising training efficiency; part of the Sustainable AI service in UC1 Scenario 1 / Prototype 1 (D3.4).	Developed under Task 3.1 and Task 3.3. CUPD01 concerns the implementation of two aggregation methods based on a dual local penalty (ADMM-style) without gossip averaging. The component is integrated into the DFL framework CUMU02. Validation carried out through preliminary results, reported in D3.3 and D3.4, and through P1, UC 1.1; related paper not yet submitted.
CUPD02 **Spiking Neural Network Simulator**	Spiking Neural Network simulator providing sparse, event-driven models that reduce the edge compute footprint; part of the Sustainable AI service in UC1 Scenario 1 (D3.4).	Developed under Task 3.3. CUPD02 is the software implementation, integrated into the DFL framework CUMU02, of the research output on SNNs obtained by the UNIPD team. Validation proved in D3.3 and D3.4 and reported here through P1, UC 1.1. Related publications include conference papers at AMLDS 2025, IEEE MLSP 2025, and IEEE PerconAI 2026. A journal extension will soon be submitted.
CUPD03 **Jamming Detection**	Machine-learning jamming detection from I/Q samples (WP5).	This component has been developed under Task 5.1. The component CUPD03 exploits one-class classification for jamming detection in private 6G networks using spectrograms and dynamic graphs for jamming detection in cell-free MIMO networks. Validation proved in deliverable D5.2. Related publications in IEEE PIMRC 2026, IEEE SPAWC 2024 and ICC Workshops 2024. A journal extension has been submitted to IEEE Transactions on Information Forensics and Security.

CUPD04 **PHY-layer enhanced Authentication & Key Agreement**	Novel PHY-layer Authentication and Key Agreement protocols for low-latency, low-complexity scenarios, including false-base-station authentication (WP5).	This component has been developed under Task 5.2. The component CUPD04 exploits challenge-response solutions operating at the physical layer with RISs and key generation with drones. Validation proved in deliverable D5.2. Related publications in IEEE Transactions on Information Forensics and Security 2024 and 2025, ICC workshops 2024, Globecom workshops 2024, IEEE SPAWC 2025. A journal extension will be submitted to IEEE Transactions on Information Forensics and Security.
CUPD05 **Cross-Layer Holistic Security Anomaly Detection**	Cross-layer holistic security anomaly-detection system for early anomaly identification (WP5).	This component has been developed under Task 5.3. The component CUPD05 exploits cross-layer solutions operating both at the physical and network layers. Validation proved in deliverable D5.2. Related publications in IEEE ICC workshops 2026. A journal extension will be submitted to IEEE Transactions on Information Forensics and Security.
NXW (Nextworks)		
CNXW01 **Zero-Touch Security Orchestrator**	Zero-Touch Security Orchestrator components for semantic reasoning and cataloguing of Security Functions and Target Environments; core of UC2 / Prototype 2 (D4.4).	Defined in WP4 (T4.1), evolution described in all the WP4 deliverables (D4.1, D4.3, and D4.4). Component validated in Prototype 2 (3.2) and UC2 (4.2).
CNXW02 **Network CNN-based IDS**	Network CNN-based IDS converting raw traffic into images for CNN-based intrusion detection (D4.4).	Experimental results carried out in (T4.3), described in **[RGL+24]**
CNXW03 **Resource Orchestrator**	Resource Orchestrator managing compute, network and storage resources in the cloud edge continuum. Responsible of Security Funcitons and Security Closed Loops stages deployment. (D4.4)	Defined in WP4 (T4.4), evolution described in all the WP4 deliverables (D4.1, D4.3, and D4.4). Component validated in Prototype 2 (3.2) and UC2 (4.2).
CNXW04 **Closed-Loop Management (S-CL Manager)**	Closed-Loop Management (S-CL Manager) handling governance and coordination of monitoring/analysis/decision/execution stages composing cloud-native closed loops; drives the multi-loop scenarios UC2.2 and UC2.3 (D4.4). Introduced after D6.1.	Defined in WP4 (T4.1), evolution described in all the WP4 deliverables (D4.1, D4.3, and D4.4). Component validated in Prototype 2 (3.2) and UC2 (4.2).
ENSEA / CYU		
CENS01 **PHY monitoring (SNR, LoS/NLoS)**	PHY monitoring of the physical context (SNR, LoS/NLoS); supports jamming detection and localisation in UC1 Scenario 2 / Prototype 4 (WP5).	This component has been developed under Task 5.1. CENS01 experimented with AoA-based localization in an outdoor mMIMO OFDM system, focusing on its robustness to impersonation attacks and its applicability to PLA. The goal was to identify user trajectories (tracks), in LoS and NLoS and test the efficiency of AoA-PLA, motivated by our work showing its robustness against impersonation in digital arrays. Validation proved in UC1.2 and P4. This work has been accepted for publication in IEEE Globecom 2025. Related conference submissions under review include IEEE Globecom.

CENS02 **Secrecy and information leakage**	Estimation of available secrecy rate with wiretap-coding and beamforming configuration for a target information-leakage level (WP5).	This component has been developed under Task 5.2. 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. Validation proved in the physical layer closed loop in D5.2. Related work was accepted at the IEEE International Symposium on Information Theory (ISIT 2026). Related journal submissions under review include IEEE OJVT.
CENS03 **Trustworthy Sensing and Localization**	Trustworthy sensing and localisation, including Sybil-attack detection and sensing-accuracy trustworthiness; UC1 Scenario 2 / Prototype 4 (WP5).	This component has been developed under Task 5.3 and 6.1. CENS03 integrates a CSI phase-correction pipeline to mitigate hardware impairments and utilises antenna subarray aggregation across frequencies to sharpen AoA precision. By fusing these results with complementary features such as Time-of-Flight (ToF) and Received Signal Strength Indicator (RSSI), the system maintains high detection rates even in challenging proximity scenarios. Validation proved in UC1.2 and P4. This work has been accepted for publication in IEEE ICC 2026. Related journal and conference submissions under review include 4 IEEE Globecom and 1 IEEE Commun. Letters.
CENS04 **AoA-based Physical Layer Authentication**	AoA-based Physical Layer Authentication preventing spoofing/impersonation; validated on real CSI in UC1 Scenario 2 / Prototype 4 (D5.3).	This component has been developed under Task 5.2. Related works in CENS04 span multiple technological readiness levels, including both i) theoretical results on the pertinence of the AoA as an unforgeable physical feature that can enable spoofing resilient PLA, ii) analysis of angle of arrival based physical layer authentication (AoA-PLA) under advanced spoofing attacks using RIS (joint work with UNIPD) and finally iii) a PoC demonstration of its feasibility on a real, outdoor, mMIMO dataset. Validation proved in UC1.2 and P4. Related publications have been made in IEEE TIFS, IEEE Wireless Communications Letters, IEEE Globecom 2025 and IEEE WIFS 2025. Related journal and conference submissions under review include 2 IEEE Globecom and 1 IEEE TMLCN.
CENS05 **Fast SKG using LSTM networks for privacy amplification**	Fast secret-key generation using LSTM-based privacy amplification; achieved a 100% reconciliation rate in UC1 Scenario 2 / Prototype 4 (D5.3).	This component has been developed under Task 5.2. 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. Validation proved in UC1.2 and P4. This work has been accepted for publication in IEEE CSCN 2025. Related journal submissions under review include IEEE TIFS, IEEE TSP, and IEEE OJ-COMS.
LIU (Linköping University)		
CLIU01 **Semantics-aware task scheduling in Federated Learning**	Semantics-aware, user-oriented task-scheduling algorithm for federated learning; part of the Sustainable AI service in UC1 Scenario 1 (WP3).	This component has been developed under Task 3.3. The corresponding studies were published in proceedings of IEEE ICASSP workshop in 2024, IEEE SPAWC 2025, and IEEE ICMLCN 2026. The results were reported in D3.2, D3.3, and D3.4.
CLIU02 **Remote estimation under heterogeneous semantic significance**	Remote state-estimation framework that weights data by semantic significance leveraging system history (WP4).	This component has been developed under Task 4.2. Results are reported in D4.3, D4.4, and D5.2, and validation is provided in D5.2. The truncated MDPs are proven to converge to the original problems at exponential rate, satisfying the EO3 KPI of ≥ 90% algorithm convergence to optimal predictions, and the communication reduction demonstrated up to 50% fewer transmissions for equivalent estimation quality, further supporting the KPI of

		EO3. The results were published in IEEE Transactions on Communications, IEEE Transactions on Information Theory, and ACM MobiHoc 2024.
EUR (EURECOM)		
CEUR01 **XAI AI/ML algorithms**	Set of techniques for ensuring and enhancing the trustworthiness of AI/ML algorithms (WP3).	Developed under Task 3.4. The core component is defined in D6.1. Results are reported under D3.2 and D3.3, and the key algorithms and metrics have been disseminated in international conference publications, more specifically the results were published in IEEE International Geoscience and Remote Sensing Symposium (IGARSS) 2024, WiOpt 2024, International Conference on Advanced Machine Learning and Data Science (AMLDS) 2025
CEUR02 **Risk-averse Resource Management Framework**	Resource control and optimisation framework incorporating risk aversion and subjective performance assessment (WP4).	Defined in WP4 (T4.4), evolution described in all the WP4 deliverables (D4.1, D4.3, and D4.4).
THALES		
CTHA01 **Security Orchestrator**	Security Orchestrator implementing security policies across network, IT and application services on edge and cloud infrastructure	The security policy implementation and enforcement is based on SSLAs (Security Service Level Agreement), which are managed by a policy manager component inside the ZTSO (D4.4). This component also implements interfaces and verification processes to coordinate the actions taken by the others ZTSO's components (NetSecaaS Gateway, GenAI Gateway, Ontology Manager, Context Manager) to maintain their compliance with the enforced SSLAs.
CTHA02 **Monitoring and Closed-Loop Remediation System**	Monitoring and Closed-Loop Remediation System using eBPF-based observation and closed-loop security remediation (D4.4).	This component implements a GenAI Gateway inside the ZTSO (D4.4) to leverage external generative AI services by contextualizing security policies enforcement plans, or security alerts raised by the eBPF monitoring system. From SSLA sent by the policy manager (CTHA01) of the ZTSO, this component's output generates remediation workflows that aim to maintain the SSLAs on the target system against security threats. In a second mode and from security alerts sent by the alert manager of the ZTSO (D4.4), this component's output also generates remediation workflows that aim to mitigate security incidents detected in the target system.
GOHM		
CGHM01 **RF Fingerprinting Migration**	RF Fingerprinting Migration model enabling domain-invariant RF fingerprinting; informed a standalone PoC providing KPI6/KPI7 evidence (D5.2).	Developed under Task T5.1. Initial results are reported in D5.2. A paper presenting this work was accepted and published at EuCNC & 6G Summit 2025 [AYS26]. A real-hardware dataset was collected and publicly released on Zenodo [AYA+25].
CGHM02 **RF-PREDICT**	RF-PREDICT, a predictive model anticipating RF-fingerprint changes for low-power sensors to support trustworthy sensing (D5.2).	Developed under Task T5.3. Initial results are reported in D5.2. A longitudinal dataset was collected and shared on Zenodo under restricted access, pending publication [AYY+26]. **Cross-component note:** the expertise from CGHM01 and CGHM02 informed a standalone RF-fingerprinting PoC, validated separately, providing KPI-level evidence for KPI6 and KPI7 (Section 5.2.2.2).

AXON		
CAXN01 **Proactive Threat Prediction and Mitigation for 6G Security Orchestrators**	Three containerised AI/ML functions: detection (XGBoost, 98.15%), mitigation over M1–M16 (Binary-Relevance Random Forest, 98.89%), 5-minute-ahead prediction (Binary-Relevance Random Forest over TSFresh, 95.36%) for pre-emptive CACAO/OpenC2 remediation trained on CryptoToN-IoT dataset (CAXN02 4).	Developed under Task T4.3 and defined as CAXN01 in D6.1. Results reported in deliverable D4.4 paragraph §2.3 (Tables 2-27/2-28) and validated in UC2 (see paragraph §5.2.3, TEST-UC2-01 and TEST-UC2-05). Delivered as Docker containers with a CSV/OpenC2 interface. Also integrated and running at TRL 6 in CoGNETs (Grant Agreement 101135930). Registered on Zenodo as "Proactive Threat Prediction and Mitigation for 6G Security Orchestrators (CICToN-IoT modules – DOI: 10.5281/zenodo.20795930). Registered as IE18a / FIP18a in deliverable D7.3.
CAXN02 **CryptoToN-IoT – cryptomining-augmented IoT network-flow dataset for 6G security**	This is a cryptomining-augmented extension of UNSW's ToN-IoT dataset (Coinhive/Madominer/Xmrstack; CICFlowMeter-regenerated). It includes 16,422,866 flows, 12 attack types, temporal 60:20:20 split. It is named CICToN-IoT in deliverable D4.4 paragraph §2.3. It is the training/benchmark set for CAXN01.	Developed under Task T4.3 and reported in deliverable D4.4 paragraph §2.3. It has been used to train and evaluate CAXN01 in UC2 (recall paragraph §5.2.3). Published as a research dataset (restricted) on Zenodo [AXCCA]. Registered as IE18b / FIP18b in deliverable D7.3.

Appendix B: Final Interface Specifications (NXW)

This appendix lists the final interface specifications for the components developed across the project, indicating the format in which each interface is published and the corresponding reference (DOI or repository URL) where the specification can be accessed.

Component	Format	Reference (DOI / URL)
Ontology Manager	OpenAPI (JSON)	https://doi.org/10.5281/zenodo.16925313
Catalogue Manager	OpenAPI (JSON)	https://doi.org/10.5281/zenodo.16925327
Security Context Manager	OpenAPI (JSON)	https://doi.org/10.5281/zenodo.19002100
Policy Manager	OpenAPI	D4.4 §3, Fig. 3-22 (no standalone DOI)
SSLA Manager	OpenAPI (YAML)	https://github.com/ThalesGroup/ssla-manager-rest (openapi.yaml)
GenAI4SOAR	Config	https://doi.org/10.5281/zenodo.18962916
S-CL Governance Catalogue	OpenAPI (JSON)	https://doi.org/10.5281/zenodo.14193484
S-CL Governance LCM	OpenAPI (JSON)	https://doi.org/10.5281/zenodo.14193635
Security Resource Orchestrator	OpenAPI (JSON)	https://doi.org/10.5281/zenodo.14193659
DFL Framework	OpenAPI	https://doi.org/10.5281/zenodo.20714755
Global Model Repository (GMR)	OpenAPI	https://doi.org/10.5281/zenodo.20641415
Programmable Monitoring Platform (PMP)	OpenAPI / REST	https://doi.org/10.5281/zenodo.20638570
PHY Layer Security Demonstrator	OpenAPI 3.1 (JSON+YAML)	https://doi.org/10.5281/zenodo.20723315
RF Fingerprinting PoC	OpenAPI 3.0.3	https://doi.org/10.5281/zenodo.20714605

Appendix C: Demo Materials Index

This appendix provides a consolidated index of the demonstration materials produced for all five ROBUST-6G prototypes. For each prototype, the table identifies the responsible partner(s), the demonstration format, a summary of the demonstration storyline, the key capabilities shown, and a placeholder for the reference to the recording or live setup. Recording links are to be filled in by the responsible partner before final submission.

Prototype	Name	Partner(s)	Demo Format	Demonstration Summary	Key Capabilities Shown	Recording
P1	Trustworthy AI	UMU	Live setup available (Docker) and recorded demo videos	A DFL federation is launched across distributed edge nodes training the CyberNet intrusion detection model on the TON_IoT dataset. Byzantine model poisoning attacks are injected by malicious nodes; Krum aggregation filters them in real time. After each round, XAI services compute SHAP-based feature attribution stability scores and generate t-SNE visualisations of the latent space. All artefacts; model weights, trust metrics, and explainability reports, are stored and retrieved via the GMR dashboard.	• Decentralised federated learning (P2P gossip) • Byzantine-robust aggregation (Krum) • XAI trustworthiness monitoring (SHAP / t-SNE) • Privacy-preserving aggregation (AES-GCM) • GMR versioned model storage and retrieval	[P1EuCNC26]
P2	Multi-Layer Zero-Touch Defender	NXW	Live setup available (hybrid docker + k8s) + recorded video	A Security Service Level Agreement (SSLA) is submitted to the Zero-Touch Security Platform. The ZTSO parses the intent, selects appropriate security functions using the security ontology, generates a CACAO/OpenC2-compliant Incident Response Playbook via GenAI, and deploys a security closed loop. The Programmable Monitoring Platform collects live telemetry, triggering automated threat detection and mitigation without human intervention.	• SSLA ingestion, validation, and parsing • Semantic security function selection • GenAI-based Incident Response Playbook generation • Closed-loop security management • Dynamically reconfigurable pervasive monitoring	[P2EuCNC26]
P3	NetSecaaS Gateway	TID	Live setup available (hybrid docker +k8s)	A third-party application issues high-level REST API calls to the NetSecaaS Gateway following the CAMARA pattern. Two interaction directions are demonstrated: (i) retrieval of explainability artefacts and security capability metadata from the Data Fabric, and (ii) submission of a simplified SSLA that	• Security capability exposure via intent-based REST APIs • Data retrieval and action triggering (CAMARA pattern) • OAuth-protected access and Data Governance enforcement	[P3EuCNC26]

			+ recorded video	is translated into a structured orchestration request forwarded to the Security Orchestrator. Authentication and policy-based access control are enforced throughout.	• SSLA transformation and forwarding • Retrieval of XAI artefacts from the Data Fabric	
P4	Physical and Sensing Layer Trustworthiness	ENSEA	Live setup available + recorded video	A simulated indoor industrial 6G environment is subjected to jamming and impersonation attacks. PHY monitoring detects SNR degradation; the spatial GLRT and WL-CUSUM detect and localise the jammer; RAN control applies adaptive power compensation to restore link quality. AoA-based physical layer authentication verifies device identity and flags impersonation attempts. LSTM-accelerated secret key generation produces shared keys between communicating nodes while maintaining resilience against eavesdropping.	• Jamming detection and localisation (spatial GLRT + WL-CUSUM) • Adaptive PHY-layer power compensation • AoA-based mutual authentication (MUSIC-based, CENS04) • Fast secret key generation with LSTM-based privacy amplification (CENS05) • Physical Layer Security Closed Loop (monitor–analyse–actuate)	[P4EuCNC26]
P5	Master Prototype	NXW	Recorded video	The integrated ROBUST-6G platform is demonstrated end-to-end across three coordinated workflows: (i) an administrator retrieves a certified AI model from the GMR, onboards it into the ZTSO catalogue as an orchestrable security function, and a consumer submits an SSLA that triggers its deployment as a security service; (ii) a PHY security service is modelled in the ZTSO ontology and orchestrated via dedicated OpenC2 actuators in a security closed loop; (iii) a third-party consumer submits a simplified API request through the NetSecaaS Gateway, which translates it into a full SSLA and forwards it to the orchestrator.	• GMR-to-ZTSO AI model onboarding and orchestration • PHY security closed loop orchestration via OpenC2 actuators • CAMARA-style security capability exposure (NetSecaaS) • End-to-end zero-touch security service lifecycle • Multi-prototype interoperability across all architecture layers	Will be available soon on the ROBUST-6G Youtube channel.

Appendix D: KPI and Objective Traceability Matrix

This appendix maps each DoA-defined KPI to its corresponding Global Objective, the prototype and use case under which it was validated, the components and evidence supporting the result, and the source deliverable.

OBJ	KPI	DoA KPI target	Validated by	Components & evidence (§ in D6.3 / source)	Source	Result
OBJ3 Trustworthy & sustainable AI (WP3 / D3.4)	3.1	Membership-inference precision ~50% → ~20%; evasion attack success ~60% → ~20%	**Prototype 1 · UC1.1**	CUCD01 (FL poisoning/inference), CEBY02 (privacy-preserving DFL); MIA accuracy reduced 0.875→0.400 (MNIST), 0.967→0.467 (5G-NIDD)	D3.4	Partially achieved
	3.2	Cut AI-security energy up to 10×; carbon neutrality with renewables	**Prototype 1 · UC1.1**	CUPD01 (ADMM DFL), CUPD02 (SNN simulator) – sustainable-AI service; populate TEST03 from D3.4	D3.4	Achieved
	3.3	Keep trustworthiness / performance trade-off within 10%	**Prototype 1 · UC1.1**	CEBY02, CUMU05 (XAI integration); trust vs. accuracy trade-off bounded	D3.4	Achieved
OBJ4 Zero-touch security management (WP4 / D4.4)	4.1	Increase security-orchestration efficiency by 10% using AI	**Prototype 2 · UC2**	CNXW01 (ZTSO) with GenAI4SOAR – removes manual playbook drafting (5.2.3.4 QT 4.1)	D4.4	Achieved
	4.2	HCOM plane meets/exceeds SSLA QoS in ≥ 95% of cases	**Prototype 2 · UC2.1-2.2**	CNXW01 Security Context Manager maps 100% of validated SSLAs (5.2.3.4 QT 4.2)	D4.4	Achieved
	4.3	Reduce threat response time by ≥ 30% using AI and XAI	**Prototype 2 · UC2**	CAXN01 (threat prediction) + XAI; detection latency ~10.5 s (5.2.3.4 QT 4.3)	D4.4	Achieved
	4.4	Pervasive cognitive closed loops (reflex / peripheral / central)	**Prototype 2 · UC2**	CNXW04 (S-CL Manager) – qualitative in D6.3	D6.3	Achieved
	4.5	Number of managed domains > 3	**Prototype 5 · UC2**	CNXW03 (Resource Orchestrator); Cloud/Edge + IoT + PHY domains (5.2.3.4 QT 4.5)	D6.3	Achieved
	4.6	Minimum number of coordinated closed-loops = 5	**Prototype 2 · UC2.3**	CNXW04; 5 peripheral + 1 master loop in smart-agriculture demo (5.2.3.4 QT 4.6)	D6.3	Achieved
OBJ5 Physical & sensing-layer security (WP5 / D5.3)	5.1	Jamming / DoS detection accuracy > 90%	**Prototype 4 · UC1.2**	CUPD03 (jamming detection), CENS01 (PHY monitoring); ~100% (5.2.2.1)	D5.3	Achieved
	5.2	Sybil detection > 70% via localisation + RF fingerprinting	**Prototype 4 · UC1.2**	CENS03 (trustworthy sensing), CGHM01 (RFFI); TPR 92.1% (5.2.2.2)	D5.3	Achieved
	5.3	Authentication / key-agreement latency < 5 ms (static nodes)	**Prototype 4 · UC1.2**	CENS05 (fast SKG, LSTM), CUPD04 (PHY AKA); < 1 ms (5.2.2.1)	D5.3	Achieved
	5.4	6G resilience increased by ≥ 20% via mitigation	**Prototype 4 · UC1.2**	CGHM01; +92.1 pp – revised from Partially to Achieved	D5.3	Achieved
OBJ6 Cross-cutting / integration (WP3-WP5)	6.1	Share model updates in < 30 min, 95% acknowledged in time	**Prototype 1 · UC1.1**	CUMU02 (DFL Framework)	D3.4	Achieved
	6.2	Average ≥ 5% ML/DL accuracy improvement after trust-based updates	**Prototype 1 · UC1.1**	CUMU02, CUMU04 (Krum); +8.77 pp (5.2.1.2)	D3.4	Achieved
	6.3	Mean anomaly-detection accuracy ≥ 85% for active PHY/sensing attacks	**Prototype 4 · UC1.2**	CUPD05 (cross-layer anomaly detection); D5.3 App.B.2	D5.3	Achieved
	6.4	Detection time < 5 min at physical / sensing layers	**Prototype 4 · UC1.2**	CUPD05; D5.3 App.B.2	D5.3	Achieved
	6.5	False-positive rate for PHY/sensing attack detection < 5%	**Prototype 4 · UC1.2**	CUPD05; D5.3 App.B.3	D5.3	Achieved
	6.6	Mitigation time < 10 min (detection to deployment)	**Prototype 2 · UC2**	CNXW04; ~91.5 s end-to-end (5.2.3.3)	D6.3	Achieved

OBJ	KPI	DoA KPI target	Validated by	Components & evidence (§ in D6.3 / source)	Source	Result
	6.7	Mitigation accuracy ≥ 90% in correct countermeasures	**Prototype 2 · UC2**	CAXN01; 100% deterministic OpenC2 (5.2.3.3)	D6.3	Achieved
	6.8	API average call latency < 500 ms and max < 1 s	**Prototype 3 · UC3**	CTID03 (NetSecaaS); avg 52.8 ms / max 601.2 ms (5.2.4 & 5.3)	D6.3	Achieved

Appendix E: TRL Assesment

Prototype	Key Components	Proposal Components	TRL Start	TRL End	TRL Achieved	Justification
P1	Decentralised Federated Learning (DFL) with Byzantine-robust Krum aggregation; XAI Module (SHAPRefine, CPCF, VAE confidence, LRP); Global Model Repository (GMR); Fairness and trustworthiness evaluation; Privacy-preserving ML	Decentralised Federated Learning Framework; Trustworthiness Evaluation Framework; AI/ML Security; Privacy-Preserving ML; XAI-enabled FL poisoning defence; Sustainable and Fair AI/ML framework	TRL 2	TRL 3-4	**TRL 5-6**	Operational containerised DFL deployment; SHAPRefine achieving 90% dimensionality reduction and up to 95% minority-class F1; CPCF Spearman correlation -0.51 to -0.57 with task accuracy; VAE latent-space confidence validated (optimal dim=20, KL=0.25); end-to-end validation in Open-RAN near-RealTime RIC testbed; accuracy improvement of up to 15% over baseline.
P2	Zero-Touch Security Platform (ZTSP); Security Service Level Agreement (SSLA) composition; GenAI-assisted incident response playbook generation (GenAI4SOAR); Programmable Monitoring Platform (PMP); Security Closed Loop Manager; Network Data Fabric; Predictive cybersecurity algorithms	Zero-Touch Security Orchestrator; Programmable Monitoring Platform; Network Data Fabric; Predictive Cybersecurity Algorithms; AI/ML Security	TRL 2-3	TRL 4	**TRL 5**	ZTSP fully deployed on NXW testbed; automated SSLA-based service composition demonstrated end-to-end; GenAI-assisted playbook generation operational; closed-loop security automation validated across two use-case scenarios;
P3	Network Security-as-a-Service (NetSecaaS) Gateway; CAMARA-style REST API exposure; Security capability abstraction layer; Data Management Module; Third-party vertical application integration	Network Data Fabric; Zero-Touch Security Orchestrator; AI/ML Security; Privacy-Preserving ML	TRL 2-3	TRL 4	**TRL 5**	Working NetSecaaS Gateway abstracting internal security operations into CAMARA-style REST APIs; third-party vertical application integration validated; functional API endpoints demonstrated in testbed; security capabilities exposed without exposing internal complexity.
P4	Physical Layer Security Closed Loop (PLCL); Jamming detection and AoA-based authentication; Secret Key Generation (SKG); RF fingerprinting for IoT device authentication; PHY anomaly detection; Localization privacy; Semantic-aware security metrics; Keyless transmissions (RIS, beamforming)	Attack detection and mitigation for resilient 6G radio; Robust and fast PLS authentication; Provably secure keyless transmissions; Trustworthy sensing and localization privacy; Adaptive RF Fingerprint Migration; Semantic-aware metrics for security	TRL 2-3	TRL 3-4	**TRL 5**	PLCL demonstrated real-time jamming detection, AoA-based authentication and SKG; RF fingerprinting for IoT authentication validated; PHY anomaly detection operational; beamforming and RIS-enabled keyless transmissions prototyped; energy reduction up to 10x demonstrated.
P5	Full integration of P1-P4 capabilities; Cross-WP interoperability; Unified security platform storyline; End-to-end 6G security use-case execution across all partner testbeds	All components (integrated across WP2-WP5)	TRL 2	TRL 4	**TRL 4**	Master prototype unified all five prototype capabilities into a coherent security entity; interoperability of AI-based security functions; integrated platform performance meeting project KPIs demonstrated end-to-end.

Appendix F: Dataset Catalogue

This appendix lists the project-generated and partner-contributed datasets used to validate the ROBUST-6G use cases and prototypes, identifying for each dataset the responsible partner(s), the use case or prototype it supports, and a brief description of its source.

ID	Dataset Name	Partner	Use Case / Prototype Supported	Dataset Source
DS-01	RF Fingerprinting Migration	GOHM	Physical Layer Security Prototype	Project-generated dataset collected from 30 custom-built IoT transmitters and 3 SDR receivers in a controlled laboratory environment
DS-02	RFFI-Temporal: A Long-Term RF Fingerprinting Dataset for Temporal Drift Analysis	GOHM	Physical Layer Security Prototype	Project-generated longitudinal RF fingerprinting dataset comprising more than 6.3 million packet captures collected over a 67-day period
DS-03	BRISC: A Dataset of Channel Measurements With a Reflective Intelligent Surface at 5 GHz	UNIPD	RIS-Assisted Communications Prototype	Project-generated CSI measurement dataset acquired using 10,000 RIS configurations at 5 GHz
DS-04	NetsLab-5GORAN-IDD: 5G Open Radio Access Network Multi-Modal Intrusion Detection Dataset	UCD	Security Use Case / Intrusion Detection Prototype	Project-generated multi-modal dataset containing network-layer and radio-layer measurements collected from a 5G Open-RAN testbed under benign and attack scenarios
DS-05	Radio Frequency Fingerprint-Based Classification Performance Analysis with ML Models in the Presence of Hardware Impairments	EBY	RF Fingerprinting Prototype Validation	Project-generated dataset containing raw RF signal samples and extracted statistical features for machine-learning-based classification experiments
DS-06	CDL Dataset	CHA	AI-Assisted Channel Prediction / MIMO Prototype Validation	Synthetic dataset generated using the 3GPP Clustered Delay Line (CDL) channel model and represented as CSI measurements
DS-07	CryptoToN-IoT: A cryptomining-augmented IoT network-flow dataset for 6G security AI	AXON	Security Use Case / Intrusion Detection, Mitigation and Future Attack Prediction (Prototype 2)	Project-generated dataset of CICFlowMeter statistical network-flow features, derived from real captured network traffic (UNSW's ToN-IoT IoT-testbed captures and real cryptomining captures, e.g., Coinhive/Madominer/Xmrstack) re-processed into one CIC-format dataset of about 16.4 million flows spanning benign traffic and 12 6G attack types, with a temporal train/validation/test split. Named CICToN-IoT in deliverable D4.4.

References

[AsyncPG] Asyncpg authors, “A fast PostgreSQL Database Client Library for Python/asyncio”, GitHub repository. Available: https://github.com/MagicStack/asyncpg

[AXCCA] C. Zarakovitis, C.-Y. Pee & W. C. Yau, “CryptoToN-IoT: a cryptomining-augmented IoT network-flow dataset for 6G security AI [Data set]”. Zenodo. 2026, https://doi.org/10.5281/zenodo.20796497

[AXPTP] C. Zarakovitis, C.-Y. Pee, & W. C. Yau, ”Proactive Threat Prediction and Mitigation for 6G Security Orchestrators (CICToN-IoT modules) (1.0.0)”. Zenodo. 2026, https://doi.org/10.5281/zenodo.20795931

[AYA+25] C. Ayyıldız, F. E. Yıldız, Ö. Ayyıldız and V. C. Yıldırım, "RF Fingerprinting Migration," Zenodo, 2025, https://doi.org/10.5281/zenodo.14801935

[AYS26] C. Ayyıldız, F. E. Yıldız and B. E. Süzek, "Data-efficient domain adaptation for receiver-invariant radio frequency fingerprinting identification," in Proc. IEEE European Conference on Networks and Communications (EuCNC) & 6G Summit, 2026, accepted for publication.

[AYY+26] C. Ayyıldız, F. E. Yıldız, V. C. Yıldırım and D. Çakmak, "RFFI-Temporal: A Long-Term RF Fingerprinting Dataset for Temporal Drift Analysis," Zenodo, 2026, https://doi.org/10.5281/zenodo.18952487

[BEG+17] P. Blanchard, E.M. El Mhamdi, R. Guerraoui, J. Stainer, "Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent", Advances in Neural Information Processing Systems 30 (NIPS), pp. 1-11, 2017.

[CDL-DFL] CyberDataLab, University of Murcia, "ROBUST-6G DFL Framework", GitHub repository. Available: https://github.com/CyberDataLab/ROBUST-6G_DFL_Framework

[CDL-PMPa] CyberDataLab, University of Murcia, "ROBUST-6G Programmable Monitoring Platform (PMP)", GitHub repository. Available: https://github.com/CyberDataLab/ROBUST-6G_PMP

[CDL-PMPb] CyberDataLab, University of Murcia, "PMP swagger APIs", GitHub repository. Available: https://cyberdatalab.github.io/ROBUST-6G_PMP

[CDL-PMPc] CyberDataLab, University of Murcia, "Programmable Monitoring Platform APIs", Zenodo, June 2026, doi: 10.5281/zenodo.20638571.

[CEBY01-1] Ö. F. Tuna, L. Karaçay, and U. Gülen, "A Novel Method to Mitigate Adversarial Attacks Against AI-as-a-Service Functionality," in *Proc. 2024 IEEE Middle East Conf. Communications and Networking (MECOM)*, 2024, pp. 181–186. doi: 10.1109/MECOM61498.2024.10880963.

[CEBY04-1] R. Fuladi and B. Cicek, "Image-based frequency-domain analysis for robust DDoS detection in SDN," in Proc. SecSoft 2025 - 7th International Workshop on Cyber-Security in Software-defined and Virtualized Infrastructures, June 2025

[CEBY04-2] M. Akbulut, B. Cicek, and R. Fuladi, "Radio frequency fingerprint-based classification performance analysis with ML models in the presence of hardware impairments," in Proc. 2025 33rd Signal Processing and Communications Applications Conference (SIU). IEEE, 2025, pp. 1-4.

[CEBY05-1] B. Cicek and H. Alakoca, "Impact of residual hardware impairments on ris-aided authentication," in Proc. 2024 IEEE Virtual Conference on Communications (VCC). IEEE, 2024, pp. 1-6.

[FPF26] M. Franco De La Peña, Á. L. Perales Gómez, L. Fernández Maimó: "A Review of ShaTS: A Shapley-based Explainability Method for Time Series Artificial Intelligence Models", Proceedings of the XI National Cybersecurity Research Conference (JNIC), May 2026.

[JGK+24] J. M. Jorquera Valero, A. García Pérez, G. Kesik, Ö. F. Tuna, P. Giardina, E. Alberti, L. Cabanillas Rodríguez, I. Dominguez, D. Lopez, D. Ayed, M. Gil Pérez, G. Martinez Perez, "Securing Networks of the Future: A Programmable Security Monitoring Platform for Cloud Continuum", Proceedings of the 2024 IEEE Future Networks World Forum, Symposium on Security in Future Networks, pp. 153-158, 15-17 October 2024.

[JP26] E. Jeong and N. Pappas, "Computation-aware Energy-harvesting Federated Learning with Pipelined Cyclic Scheduling," to be submitted to IEEE Transactions on Mobile Computing, 2026.

[MBM+25] E. T. Martínez Beltrán, G. Bovet, G. Martínez Pérez, A. Huertas Celdrán, "DEMO: NEBULA – Decentralised Federated Learning for Heterogeneous Networks", Proceedings of the ACM SIGCOMM 2025 Posters and Demos, pp. 149-151, September 2025.

[MMG+26] I. Marroqui Penalva, E. T. Martínez Beltrán, M. Gil Pérez and A. Huertas Celdrán, "RepuNet: A reputation system for mitigating malicious clients in DFL", Computer Networks, vol. 282, art. no. 112242, pp. 1-42, 2026.

[MRB+25] A. Mayya, Y. Richhariya, A. K. Boroujeni, S. Vorberg, M. Matthé, R. Vinz, L. Senigagliesi, K. Klamka, and A. Chorti, "Context-aware secret key generation demonstrator based on physical layer security," in Proc. 2025 IEEE Conference on Standards for Communications and Networking (CSCN). IEEE, 2025.

[OAS19a] Open Command and Control (OpenC2) Language Specification Version 1.0, OASIS Standard, July 2019, url: https://docs.oasis-open.org/openc2/oc2ls/v1.0/oc2ls-v1.0.html

[OAS23a] Collaborative Automated Course of Action Operations (CACAO) Security Playbooks Version 2.0, OASIS Standard, November 2023, url: https://docs.oasis-open.org/cacao/security-playbooks/v2.0/security-playbooks-v2.0.html.

[P1EuCNC26] ROBUST-6G YouTube Channel, "EuCNC 2026 – Distributed Federated Learning (DFL) Framework Demo". Available: https://youtu.be/4j249L52URI

[P2EuCNC26] ROBUST-6G YouTube Channel, "EuCNC 2026 – Multi-Layer Zero-touch Defender". Available: https://youtu.be/vFlyInKIbNI

[P3EuCNC26] ROBUST-6G YouTube Channel, "EuCNC 2026 – NetSecaaS Gateway". Available: https://youtu.be/87xIAIascO4

[P4EuCNC26] ROBUST-6G YouTube Channel, "EuCNC 2026 – Physical Layer Security Closed Loop". Available: https://youtu.be/d78CTBxmtGM

[PMS+26] Á. L. Perales Gómez, E. T. Martinez Beltrán, P. M. Sánchez Sánchez, A. Huertas Celdrán, "TemporalFED: A Software Module for Detecting Cyberattacks in Industry 4.0 Time-Series Data Using Decentralised Federated Learning", IEEE Access, vol. 14, 64822-64835, April 2026.

[R6G24-D41] Deliverable D4.1: Security Automation for 6G. Horizon Europe Project, Grant Agreement No. 101139068. [Online]. Available: https://robust-6g.eu/

[R6G25-D43] Deliverable D4.3: ROBUST-6G AI/ML Driven Zero-Touch Security Management Platform Consolidated Design. Horizon Europe Project, Grant Agreement No. 101139068. [Online]. Available: https://robust-6g.eu/

[R6G26-D23] Deliverable D2.3: Final ROBUST-6G Architecture and ROBUST-6G Dataspace. Horizon Europe Project, Grant Agreement No. 101139068. [Online]. Available: https://robust-6g.eu/

[R6G26-D44] Deliverable D4.4: ROBUST-6G AI/ML Driven Zero-Touch Security Management Platform - Final Prototype. Horizon Europe Project, Grant Agreement No. 101139068. [Online]. Available: https://robust-6g.eu/

[R6G26-D52] Report on the use of PLS in 6G. Horizon Europe Project, Grant Agreement No. 101139068. [Online]. Available: https://robust-6g.eu/

[R6G26-D53] Release of Physical Layer Security Challenges. Horizon Europe Project, Grant Agreement No. 101139068. [Online]. Available: https://robust-6g.eu/

[R6G26-D62] Deliverable D6.2: Intermediate Validation Results. Horizon Europe Project, Grant Agreement No. 101139068. [Online]. Available: https://robust-6g.eu/

[RBG+22] Rueckauer, B., Bybee, C., Goettsche, R., Singh, Y., Mishra, J., & Wild, A. (2022). NxTF: An API and compiler for deep spiking neural networks on Intel Loihi. ACM Journal on Emerging Technologies in Computing Systems (JETC), 18(3), 1-22.

[RGL+24] M. Ruta, P. G. Giardina, G. Landi, R. G. Garroppo, "A Generalized Multi-Layer IDS for Smart Buildings," 2024 IEEE 29th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Athens, Greece, October 2024, pp. 1-6, doi: 10.1109/CAMAD62243.2024.10942818.

[SMF+25a] P. M. Sánchez Sánchez, E. T. Martínez Beltrán, C. Feng, G. Bovet, G. Martínez Pérez, A. Huertas Celdrán, "S-VOTE: Similarity-based Voting for Client Selection in Decentralised Federated Learning", Proceedings of the 2025 International Joint Conference on Neural Networks (IJCNN), pp. 1-8, June 2025.

[SMF+25b] P. M. Sánchez Sánchez, E. T. Martínez Beltrán, M. Fernández Llamas, G. Bovet, G. Martínez Pérez, A. Huertas Celdrán, "ProFe: Communication-Efficient Decentralised Federated Learning via Distillation and Prototypes", Proceedings of the ICC 2025 - IEEE International Conference on Communications, pp. 1596-1601, June 2025.

[THA26a] ROBUST6G GenAI CACAO agents configurations, Zenodo, March 2026, doi: 10.5281/zenodo.18962917.

[THA26b] SSLA UC2-1 Proactive Orchestration, Zenodo, June 2026, doi: 10.5281/zenodo.20669086.

[TMJ+26a] F. Torres Vega, E. T. Martínez Beltrán, J. M. Jorquera Valero, A. Huertas Celdrán, M. Gil Pérez, "DFL Framework API", Zenodo, June 2026, doi: 10.5281/zenodo.20714756.

[TMJ+26b] F. Torres Vega, E. T. Martínez Beltrán, J. M. Jorquera Valero, A. Huertas Celdrán, M. Gil Pérez, "Global Model Repository API", Zenodo, June 2026, doi: 10.5281/zenodo.20641416.

[UCD02-1] T. Senevirathna, C. Sandeepa, B. Siniarski, M.-D. Nguyen, S. Marchal, M. Boerger, M. Liyanage, and S. Wang, "Enhancing accountability, resilience, and privacy of intelligent networks with XAI," IEEE Open Journal of the Communications Society, vol. 6, pp. 8389-8409, 2025, doi: 10.1109/OJCOMS.2025.3608784.

[UCD02-2] C. Sandeepa, T. Senevirathna, B. Siniarski, M.-D. Nguyen, V.-H. La, S. Wang, and M. Liyanage, "From opacity to clarity: Leveraging XAI for robust network traffic classification,"

in Proc. International Conference on Asia Pacific Advanced Network, 2023, pp. 125-138. Springer.