Holistic, omnipresent, resilient services
for future 6G wireless and computing ecosystems

D5.4 – Final HORSE Platform Release

Work package	WP5
Task	T5.3, T5.4, T5.5
Due date	31/12/2025
Submission date	31/12/2025
Deliverable lead	TID
Version	1.0
Authors	All partners
Reviewers	Xavi Masip (UPC), Fabrizio Granelli (CNIT)

Abstract	This document reflects the final output of the project, including the modules development and the Use Cases validating the HORSE platform.

Keywords	Deployment, validation, testbed, KPI.

DOCUMENT REVISION HISTORY

Version	Date	Description of change	List of contributor(s)
v0.1	18/10/2025	Initial ToC	TID
v0.2	21/11/2025	First contribution of partners	All partners
v0.3	12/12/2025	Second contribution of partners	All partners
v0.4	22/12/2025	Reviewed version	CNIT, UPC, TID
v1.0	29/12/2025	Final refinements and adjustments	TID

Disclaimer

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the other granting authorities. Neither the European Union nor the granting authority can be held responsible for them.

Copyright notice

Project co-funded by the European Commission in the Horizon Europe Programme		
Nature of the deliverable:	DEM	
Dissemination Level		
PU	*Public, fully open, e.g. web*	**X**
SEN	*Sensitive, limited under the conditions of the Grant Agreement*	
Classified R-UE/ EU-R	*EU RESTRICTED under the Commission Decision No2015/ 444*	
Classified C-UE/ EU-C	*EU CONFIDENTIAL under the Commission Decision No2015/ 444*	
Classified S-UE/ EU-S	*EU SECRET under the Commission Decision No2015/ 444*	

*R: Document, report (excluding the periodic and final reports)
DEM: Demonstrator, pilot, prototype, plan designs
DEC: Websites, patents filing, press & media actions, videos, etc.
DATA: Data sets, microdata, etc
DMP: Data management plan
ETHICS: Deliverables related to ethics issues.
SECURITY: Deliverables related to security issues
OTHER: Software, technical diagram, algorithms, models, etc.

Table of contents

List of figures

List of tables

Abbreviations

3D CAD	3-Dimensional Computer Aided Design
5G	Fifth Generation of Mobile Networks
6G	Sixth Generation of Mobile Networks
AI	Artificial Intelligence
API	Application Programming Interface
AR	Augmented Reality
cKB	Common Knowledge Base
CN	Core Network
DDoS	Distributed Denial of Service
DEME	Detector and Mitigation Engine
DN	Data Network
DNS	Domain Name System
DRL	Deep Reinforcement Learning
DT	Digital Twin
DTE	Distributed Trustable AI Engine
EM	Early Modeling
ES	Elastic Search
ePEM	End-to-End Secure Connectivity Manager
FAR	Forwarding Action Rule
gNB	Next Generation Node B
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IA-DT	Impact Analysis Digital Twin
IBI	Intent-based Interface
INFR	Infrastructure
IP	Internet Protocol
IT-X	Project Iteration X
JSON	JavaScript Object Notation
k8s	Kubernetes
LRT	Light Rail Transit
ML	Machine Learning

NDT	Network Digital Twin
NEF	Network Exposure Function
NF	Network Function
NTP	Network Time Protocol
P&P	Prediction and Prevention
P&P NDT	Prediction and Prevention Network Digital Twin
PAG	Policies and Data Governance
PCAP	Packet CAPture
PDU	Protocol Data Unit
PFCP	Packet Forwarding Control Protocol
PreProc	Pre-Processing
QoS	Quality of Service
RAN	Radio Access Network
REST	Representational State Transfer
RTR	Reliability, Trust and Resilience
SEID	Session Endpoint Identifier
SM	Smart Monitoring
SMF	Session Management Function
TEID	Tunnel Endpoint Identifier
TX.Y	Task X.Y
UE	User Equipment
UI	User Interface
UPF	User Plane Function
URL	Uniform Resource Locator
VM	Virtual Machine
WebUI	Web-based User Interface
WPX	Work Package X
XML	Extensible Markup Language
XR	Extended Reality
VR	Virtual Reality

1. Introduction

The present deliverable provides the final release of the HORSE platform, representing the culmination of the research, design, integration, and validation activities carried out throughout the project's lifecycle. It consolidates all technical components, functionalities, and architectural features developed to realize the HORSE vision for intelligent, secure, and sustainable 6G networks.

This document also reports the overall results of the validation process, which has been conducted via both a well-defined set of nine demonstrations showcasing the platform's capabilities and in two key real-world use cases aligned with the project's objectives. The validation activities have assessed performance, scalability, interoperability and compliance with the framework defined in earlier deliverables. The outcomes presented here confirm the maturity and readiness of the HORSE platform for potential exploitation and further experimentation within the 6G research ecosystem.

1.1. Structure of the document

This document is organized into several key sections to present a comprehensive overview of the final HORSE platform and its validation results.

- Section 2 summarizes the final architecture of the HORSE platform, describing the main components and their integration
- Section 3 presents the HORSE final platform release, including information about the different demonstrations carried out, and detailing the demo #0 which is a full demonstration of the platform, illustrating its capabilities and operational workflows.
- Section 4 focuses on the validation of the HORSE platform through representative use cases: Secure Smart LRT Systems and Remote Rendering to Power XR Industrial applications.

This structure ensures clarity in presenting the technical development, implementation, and the outcomes of the validation activities.

2. Final HORSE Platform architecture

This section presents the final architectural design of the HORSE platform, which becomes the conceptual foundation supporting the distinct releases delivered in the project that will be validated and demonstrated through two real-world use cases: Secure Smart LRT Systems and Remote Rendering to Power XR Industrial.

The description of the final HORSE platform is accompanied by two canonical workflows developed for threat detection and prediction. These workflows illustrate how the platform's modules interact, demonstrating the end-to-end security capabilities. Specifically, they show the full process, from the detection of emerging threats, the anticipation to future risks, the enforcement of the appropriate mitigation and prevention strategies, guided by an assessment of the potential impact of the preventive actions within an emulated environment, which ensures that preventive measures are both effective and validated before their deployment in the real infrastructure.

2.1. Final HORSE architecture

The HORSE platform is designed for future 6G wireless and computing ecosystems, providing a human-centric approach to security, by enabling top-down, bottom-up and end-to-end security solutions. The proposed architecture advances existing solutions through an intelligent and adaptive security layer that leverages AI methods to improve system resilience and response, enabling early threat detection, predictive risk assessment and automated mitigation. Overall, the security layer is designed to minimize incident impact and reduce response time by leveraging advanced predictive orchestration mechanisms.

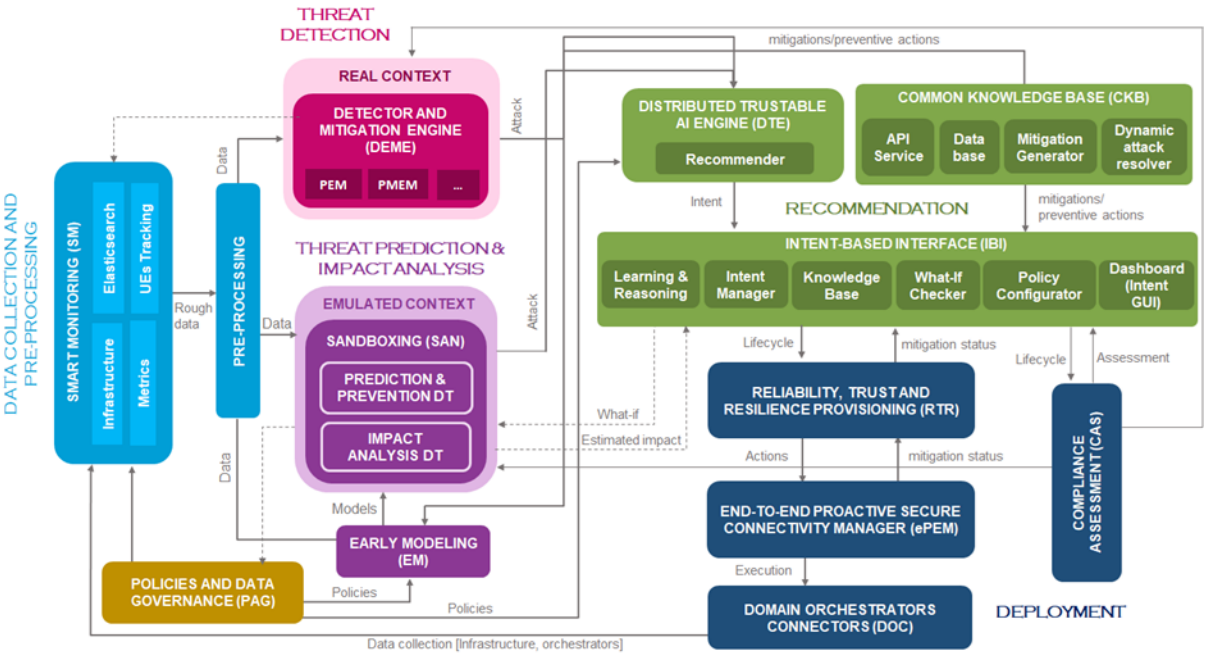

Figure 1. HORSE final architecture.

The final architectural design was conducted within WP2 and documented in Deliverable D2.4 "HORSE Landscape and Architectural Design" [1]. Figure 1 presents the final HORSE architecture, offering a visual overview of the main components and highlighting their interfaces and communication flows.

The HORSE platform provides a complete set of functionalities including: (i) data collection and preprocessing, (ii) threat detection, (iii) threat prediction and analysis, (iv) recommendation, (vi) deployment.

In the context of **Data Collection and Preprocessing**:

The Smart Monitoring (SM) component continuously gathers data from the underlying infrastructure, domain orchestrators, and resource usage associated with the lifecycle of 6G services.

The Pre-processing component unifies and standardizes all collected data, coordinating large-scale and heterogeneous sources within a unified, extensible data framework

The Policies and Data Governance (PAG) component provides the mechanisms required to define and enforce data policies for information stored and processed by the HORSE platform in Elasticsearch, covering anonymization, access control, privacy-preserving rules, and data retention.

In the context of **Threat Detection**:

The Detector and Mitigation Engine (DEME) performs threat detection directly in the real infrastructure. Its focus is on both identifying attack scenarios capable of causing disruption across significant portions of the network, as well as providing high-level mitigation guidance, particularly for threats requiring immediate response.

In the context of **Threat Prediction & Impact Analysis**:

The Sandboxing (SAN) operates in an emulated "network-in-network" environment, enabling realistic scenario emulations. This controlled context allows experimentation with service deployments, alternative topologies, traffic paths, and security network functions across multiple networks. SAN encompasses two NDT modules:

- Prediction & Prevention NDT predicts anomalies and threats within the emulated environment.
- Impact Analysis NDT estimates the impact of enforcing either mitigation or preventive strategies on the emulated context before they are applied to the real 6G infrastructure.

The Early Modeling (EM) provides the emulated environment with all the structural and behavioral models it requires, describing potential threats and attacks, their expected impact on the 6G system, and the projected impacts of enforcing mitigation or preventive actions.

In the context of **Recommendation**:

The Distributed Trustable AI Engine (DTE) based on the results from both the real infrastructure and the emulated environment, produces high-level mitigation and prevention strategies expressed as intents to be applied across 6G components.

The Common Knowledge Base (CKB) represents the centralized knowledge repository, which stores, manages and enhances information such as vulnerabilities, attack patterns, mitigation strategies, and preventive actions.

The Intent Based Interface (IBI) component translates high-level intents into executable workflows. It validates intents against relevant policies, making use of ML techniques, and consults the Impact Analysis NDT to ensure that the proposed mitigation or preventive actions meet acceptable impact thresholds. The IBI dashboard presents the current network status, detected or predicted anomalies and attacks, and the status of all executed mitigation and preventive measures.

In the context of **Deployment**:

The Reliability, Trust and Resilience (RTR), provides the mechanisms required for secure operational performance. It defines the mitigation and preventive actions to be applied by the ePEM, delivered as Ansible security playbooks derived from the workflows produced by the IBI.

The Compliance Assessment (CAS) component guarantees that ePEM's enforcement policies comply with HORSE policies and meet regulatory requirements, including 3GPP and ENISA guidelines.

The end-to-end (E2E) secure connectivity manager (ePEM) component operates as an Operations Support System (OSS) that executes the mitigation and preventive strategies defined by RTR across the infrastructure. It also maintains information about deployed applications, network services, and available resources.

The Domain Orchestrator Connectors (DOC) component integrates management and orchestration capabilities across all network segments, including RAN, transport, core, near edge, far edge and cloud.

2.2. HORSE canonical end-to-end workflows

Two canonical end-to-end workflows have been formalized to validate the core functional capabilities of the HORSE platform and to establish the baseline operational data flow. The first workflow addresses real-time threat detection, while the second targets predictive threat detection. These workflows serve as reference templates from which use-case-specific workflows have been derived for Secure Smart LRT Systems and Remote Rendering to Power XR Industrial scenarios.

2.2.1. Threat Detection Workflow

The first canonical workflow is defined for real-time threat detection within the operational environment, as depicted in Figure 2. The workflow initiates with telemetry acquisition from the underlying infrastructure. The SM module ingests measurements from both infrastructure components and domain orchestrators, forwarding them to the Pre-processing module, which performs data normalization and schema alignment to produce a unified data representation. The normalized data is then streamed to the DEME, where continuously running detection pipelines, based on rule-driven analytics, statistical models, and ML-based classifiers, identify potential threats and attack patterns.

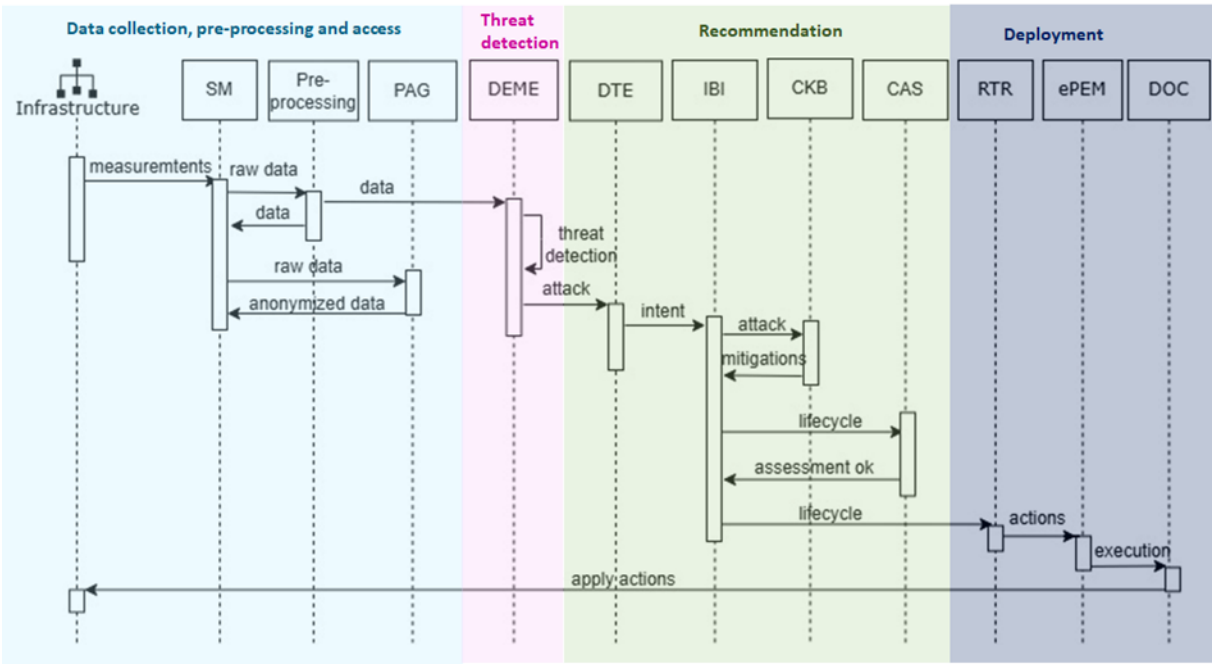

Figure 2. Threat detection canonical workflow.

In parallel, the processed data is passed through the PAG component for anonymization and policy-compliant handling before being stored in the SM's Elasticsearch. This repository

provides feedback data for periodic retraining and refinement of the ML models used across HORSE components. Upon detection of an attack or anomaly, DEME issues a notification to the DTE, which synthesizes the event into a high-level mitigation intent expressed in a machine-readable format.

The intent is forwarded to the IBI component, which queries the CKB to retrieve appropriate mitigation strategies. The IBI then generates a concrete operational lifecycle describing the sequence of mitigation steps required to address the detected threat. This lifecycle undergoes compliance validation by the CAS component to ensure alignment with HORSE policies and external regulatory frameworks. Once validated, the lifecycle is transferred to the RTR component, which derives the final set of actionable mitigation instructions to be deployed in the infrastructure.

These actions are sent to the ePEM, which upon verifying consistency with the available infrastructure state, the Domain Orchestrator Connectors (DOC) executes the necessary orchestration mechanisms and deploy the required security controls or configuration changes across the target network segments, thereby executing the mitigation procedures triggered by this workflow.

2.2.2. Threat Prediction Workflow

The second canonical workflow is defined for predictive threat analysis within the NDT environment, as illustrated in Figure 3. The workflow starts with telemetry ingestion from the operational infrastructure, which is forwarded to the Prediction & Prevention NDT. This component processes the collected data to predict the likelihood of upcoming threats or attacks. When it predicts a potential threat above a defined confidence threshold, it triggers the DTE, which synthesizes the event into a high-level preventive intent and submits it to the IBI.

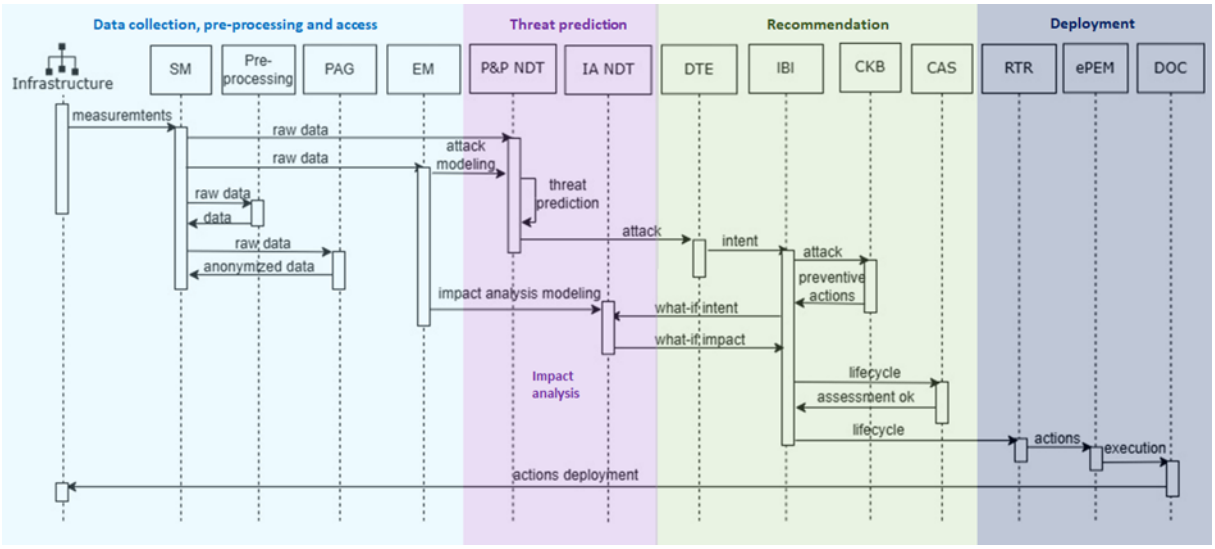

Figure 3. Threat prediction canonical workflow.

Upon receiving the intent, the IBI queries the CKB to retrieve the corresponding preventive strategies and enforcement primitives. As in the detection workflow, the IBI constructs an operational lifecycle describing the full sequence of proactive actions required to mitigate the predicted threat. However, given the inherent uncertainty of predictive analytics, this lifecycle undergoes an additional validation phase executed within the SAN environment. Specifically, the lifecycle is passed to the Sandboxing module, where the Impact Analysis NDT emulates the execution of the proposed preventive actions to assess their expected impact. The Impact Analysis NDT returns quantitative impact metrics to the IBI, which evaluates them against predefined admissibility policies. If the predicted impact falls within acceptable thresholds, the

lifecycle is forwarded to the CAS component to verify alignment with HORSE policies and the applicable regulatory framework. Upon successful validation, the IBI submits the approved lifecycle to the RTR module, which derives the final set of executable preventive actions.

These actions are then handed to the DOC component, which executes the necessary orchestration mechanisms across the infrastructure, thereby proactively reacting to the predicted threat.

3. HORSE Final Platform Release

To reach the final validation of the HORSE platform, we followed a structure pathway to achieve the final integration (Figure 4). This path in divided into four stages, each building upon the previous one.

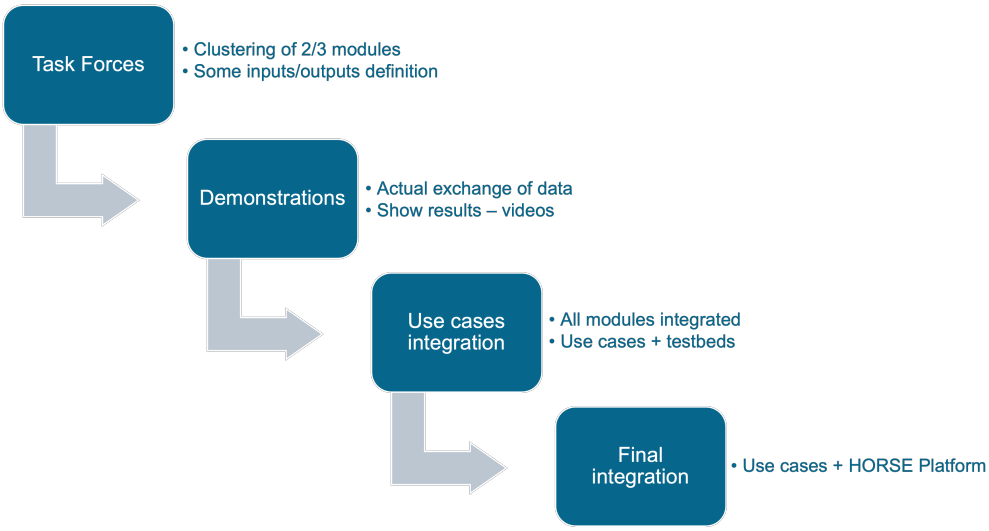

Figure 4. HORSE integration process.

1. Task forces. Focus on clustering and organizing 2 to 3 core modules of the platform, while simultaneously defining their interfaces—the critical inputs and outputs that will allow them to communicate effectively.
2. Demonstrations. The individual modules begin to interact and exchange data in real conditions. We show tangible results through videos and live demonstrations that prove the technical feasibility and showcase how the system behaves under realistic loads.
3. Use Cases integration. Bringing together all the modules developed and begin testing them against the defined use cases and their associated testbeds. This phase ensures that every component works not just in isolation, but as part of the complete HORSE ecosystem.
4. Final integration. All use cases and the complete HORSE platform are unified into a cohesive, production-ready system. At this point, we have a fully validated platform ready for deployment and for supporting future 6G network architectures.

The final validation of the HORSE platform was also conducted through the execution of several demos, each focusing on specific modules and attack scenarios, described in previous deliverables. For reporting and documentation purposes, the project's GitHub repository has been used as a centralized platform where all modules' developments are hosted: https://github.com/HORSE-EU-Project.

Also, in the GitHub of the project, we have included specific links for the demonstrations carried out during the last phase of the project: https://github.com/HORSE-EU-Project/Final-demos.

The demonstrations developed, including the specific attack or functionality to be analyzed and the testbeds used for each one of them are summarized in Table 1.

Table 1. Summary of the demonstrations carried out in the project.

Demo #	Name	Testbed
0	Hello world with a DDoS in Download link	UPC

1	Detecting NTP DDoS Amplification attack	CNIT
2	Prediction & Impact Analysis DNS Amplification attack	UMU
3	Prediction & Impact Analysis DDoS in download link	UMU
4	Detecting DT data poisoning	CNIT
5	Enforcing multidomain mitigation actions	UPC
6	Man in the Middle on IBI	UPC
7	API & Network Functions Exposure	CNIT
8	Predicting attack on signaling PFCP traffic	UPC

3.1. HORSE Platform demonstration and validation

Demo 0 is designed to showcase the full operational capability of the HORSE platform. Its main purpose is to validate the complete integration of all components of the HORSE architecture, demonstrating how the various modules communicate, exchange data, and operate as a unified cybersecurity framework.

To this end, two complementary demonstration scenarios have been developed: one focused on threat detection and another one on threat prediction. The threat detection scenario illustrates the platform's ability to perform real-time monitoring, collect telemetry from heterogeneous sources, detect attacks and provide effective mitigation strategies. The threat prediction scenario highlights the platform's proactive capabilities, including threat prediction and the enforcement of proactive strategies.

Together, these two scenarios validate the end-to-end functionality of the HORSE framework and demonstrate its capacity to deliver security solutions that encompass both the detection and the prediction of cyber threats.

3.1.1. Testbed and Context Environment

Demo 0 has been conducted in the UPC testbed, depicted in Figure 5, which consists of a 5G communication infrastructure, with all components deployed as Docker containers.

In addition to the core testbed architecture, a suite of dedicated APIs has been developed to provide fine-grained orchestration of experiments across the 5G infrastructure. These interfaces support real-time traffic monitoring, PCAP file management, attack simulation, and dynamic enforcement of mitigation policies.

Traffic monitoring is enabled through TCPdump deployed on multiple key nodes, including UEs, gNodeBs, UPF instances, the DNS server, and the gateway router.

For traffic generation and testbed interaction, a dedicated FastAPI-based service manages all packet-capture (.pcap) files generated during the experiments within the 5G environment. This API provides centralized, on-demand access to traffic captures, supporting workflow integration for post-analysis and performance validation.

A mitigation-oriented API has been developed to support automated enforcement of defensive strategies by dynamically interacting with nodes affected during attack scenarios. This interface enables a range of coordinated mitigation actions detailed in D5.3 Second HORSE Release: HORSE IT-2 version [2].

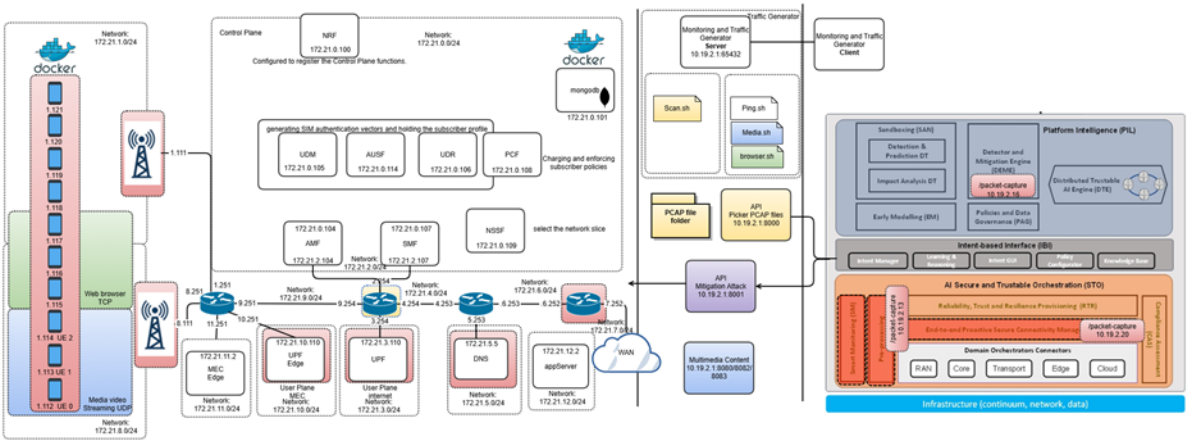

Figure 5. Demo 0 UPC testbed configuration.

3.1.2. Demo 0 Threat detection

This demonstration illustrates the operational behavior of the HORSE platform during a threat-detection scenario. The demonstration, as an illustrative example, focuses on a DNS amplification attack, and considers the platform responds by enforcing DNS rate-limiting as a mitigation measure. The demo highlights how HORSE identifies the malicious traffic pattern and applies the appropriate mitigation strategies. When the attack is initiated, the DEME component of the HORSE platform detects the abnormal traffic patterns, and the recommendation components triggers DNS rate-limiting as the mitigation strategy, which is enforced in the infrastructure by the deployment components.

The demo highlights the end-to-end workflow, from measurements collection, pre-processing, threat detection and classification of the malicious traffic, to the definition of the mitigation strategies and its enforcement on the relevant network nodes. Figure 6 presents the complete workflow used in this scenario.

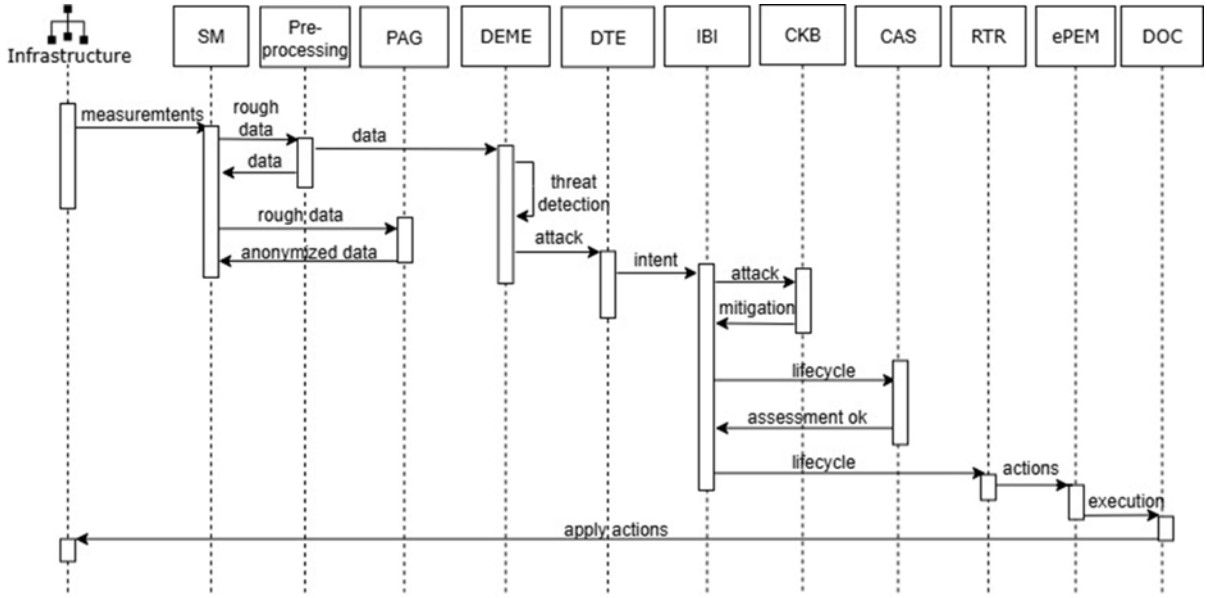

Figure 6. Demo 0 – Threat detection workflow.

A video recording of the entire demonstration is available at the GitHub repository of the project. The video provides a step-by-step walkthrough of the attack simulation, the detection process, and the automated mitigation actions executed by the HORSE platform, offering a clear visual representation of the HORSE platform behavior and interactions between the different components.

Table 2 summarizes all interactions among the HORSE components involved in the demonstration. For each component, the table provides an example of the corresponding input/output format, as well as a visual capture demonstrating the component's behavior during the execution of the demo. This unified view offers a clear understanding of how the different modules collaborate throughout the threat detection workflow.

Table 2. Demo 0 – Threat detection workflow HORSE components interactions.

Component From	Component To	Format Example
SM	Pre-processing	JSON formatted information for specific values based on the pcap data ingested the last two minutes
Pre-processing	SM	`[ { "timestamp": "<EPOCH_TIMESTAMP>", "instances": [ { "instance": "<IP_ADDRESS_OR_NODE_ID>", "features": [ { "feature": "<E.G._NTP_DNS_NEF>", "value": "<COUNTER_VALUE>" } ] }, // ... potentially more instance objects for multi-node scenarios ] } ]`
SM	PAG	• Pcap file index: pcap-data • IMEI numbers index: imei_index

		• Demo data indices:
		  • holo_demo_data
		  • holo_demo_data_api
		  • holo_demo_data_ddos

PAG	SM	• Anonymized Pcap file index: anonym_index
		• Anonymized IMEI numbers index: anonym_imei_index
		• Anonymized demo data indices:
		  • anonym_holo_demo_data
		  • anonym_holo_demo_data_api
		  • anonym_holo_demo_data_ddos

Pre-processing	DEME	[
		  {
		  "timestamp": "<EPOCH_TIMESTAMP>",
		  "instances": [
		  {
		  "instance": "<IP_ADDRESS_OR_NODE_ID>",
		  "features": [
		  {
		  "feature": "<E.G._NTP_DNS_NEF>",
		  "value": "<COUNTER_VALUE>"
		  }
		  ]
		  },
		  // ... potentially more instance objects for multi-node scenarios
		  ]
		  }
		]

DTE	IBI	{
		  "intent_type": "mitigation",
		  "threat": "ddos_amplification",
		  "host": ["172.22.0.7"],
		  "duration": 600
		}

IBI	cKB	JSON document via HTTP POST in the following format: {'attack_name': 'dns_amplification}

IBI	CAS	{
		  "input": {
		  "command": "add",
		  "intent_type": "mitigation",
		  "threat": "dns_amplification",
		  "attacked_host": [
		  "172.22.1.1"
		  ],
		  "mitigation_host": "172.21.3.254",
		  "action": {
		  "name": "dns_rate_limiting",
		  "fields": {
		  "rate": "9mbps",
		  "source_ip_filter": "0.0.0.0/0"
		  }
		  },
		  "duration": "600",
		  "intent_id": "14911220-c3e9-4c5b-a4f8-16245a7b1317"
		  }
		}

CAS	IBI	{'actions_needed': ["Rate must be at least 10mbps and include 'mbps'"], 'allow': False, 'pass_percentage': 50}

IBI	CAS	{
		"input": {
		"command": "add",
		"intent_type": "mitigation",
		"threat": "dns_amplification",
		"attacked_host": [
		"172.22.1.1"
		],
		"mitigation_host": "172.21.3.254",
		"action": {
		"name": "dns_rate_limiting",
		"fields": {
		"rate": "10mbps",
		"source_ip_filter": "0.0.0.0/0"
		}
		},
		"duration": "600",
		"intent_id": "14911220-c3e9-4c5b-a4f8-16245a7b1317"
		}
		}

CAS	IBI	{"allow": true}

IBI	RTR	{
		"command": "add",
		"intent_type": "mitigation",
		"threat": "dns_amplification",
		"attacked_host": "172.22.1.1",
		"mitigation_host": "172.21.3.254",
		"action": {
		"name": "dns_rate_limiting",
		"fields": {
		"rate": 10,
		"source_ip_filter": "0.0.0.0/0",
		"duration": 600
		}
		},
		"duration": 600,
		"intent_id": "4186821c-ec23-47e6-b008-4281d7f608ca"
		}

RTR	ePEM	{
		"command": "add",
		"intent_type": "mitigation",
		"intent_id": "34bea1d9-415c-48ac-b214-742cb6",
		"threat": "ddos_downlink",
		"target_domain": "",
		"action": {
		"name": "block_pod_address",
		"intent_id": "34bea1d9-415c-48ac-b214-66ec94742cb6",
		"fields": {
		"blocked_ips": [
		"172.22.0.7"
		],
		"device": "172.21.3.254",
		"interface": "eth4",
		"duration": 600
		}
		},
		"attacked_host": "172.22.0.7",
		"mitigation_host": "172.21.3.254",
		"duration": 600,
		"status": "playbook_created, sent_to_epem",
		"info": "Playbook created, forwarding to ePEM in background, Sent to

		ePEM → ePEM sent to DOC → DOC successfully enforced action", "ansible_command": "---\n- name: Block IP addresses\n hosts: 172.21.3.254\n become: true\n tasks:\n - name: Block all incoming traffic from IP address\n iptables:\n chain: INPUT\n in_interface: eth4\n protocol: all\n source: 0.0.0.0/32\n jump: DROP\n when: ansible_os_family == \"Debian\" or ansible_os_family == \"RedHat\"\n\n handlers:\n - name: Reload firewall rules\n service:\n name: iptables # Replace with the name of your firewall service\n state: reloaded\n when: ansible_os_family == \"Debian\" or ansible_os_family == \"RedHat\"" }

ePEM	DOC	{
		"command": "add",
		"intent_type": "mitigation",
		"intent_id": "34bea1d9-415c-48ac-b214-742cb6",
		"threat": "ddos_downlink",
		"target_domain": "",
		"action": {
		"name": "block_pod_address",
		"intent_id": "34bea1d9-415c-48ac-b214-66ec94742cb6",
		"fields": {
		"blocked_ips": [
		"172.22.0.7"
		],
		"device": "172.21.3.254",
		"interface": "eth4",
		"duration": 600
		}
		},
		"attacked_host": "172.22.0.7",
		"mitigation_host": "172.21.3.254",
		"duration": 600,
		"status": "playbook_created, sent_to_epem",
		"info": "Playbook created, forwarding to ePEM in background, Sent to ePEM → ePEM sent to DOC → DOC successfully enforced action",
		"ansible_command": "---\n- name: Block IP addresses\n hosts: 172.21.3.254\n become: true\n tasks:\n - name: Block all incoming traffic from IP address\n iptables:\n chain: INPUT\n in_interface: eth4\n protocol: all\n source: 0.0.0.0/32\n jump: DROP\n when: ansible_os_family == \"Debian\" or ansible_os_family == \"RedHat\"\n\n handlers:\n - name: Reload firewall rules\n service:\n name: iptables # Replace with the name of your firewall service\n state: reloaded\n when: ansible_os_family == \"Debian\" or ansible_os_family == \"RedHat\""
		}

DOC	Infra	curl -X 'POST' \
		'http://10.19.2.19:8001/api/mitigate' \
		-H 'accept: application/json' \
		-H 'Content-Type: application/json' \
		-d '{
		"action": {
		"fields": {
		"duration": "600",
		"source_ip_filter": [
		"172.22.0.7"
		]
		},
		"intent_id": "34bea1d9-415c-48ac-b214-66ec94742cb6",
		"name": "block_pod_address"
		},
		"command": "add",

		"info": "awaiting reinforcement", "intent_id": "34bea1d9-415c-48ac-b214-66ec94742cb6", "intent_type": "mitigation", "status": "pending", "testbed": "upc", "threat": "dns_attack" }'

3.1.3. Demo 0 Threat prediction

This demonstration illustrates the operational behaviour of the HORSE platform during a threat-prediction scenario. The demonstration focuses on a DDoS Downlink attack, for which the platform proactively responds by blocking identified malicious IP addresses as a preventive measure. The demo highlights how the Prediction and Prevention DT predicts the attack, determines the appropriate preventive action, and forwards this decision to the Impact and Analysis DT, which evaluates its impact in an emulated environment, and finally enforces the preventive action on the real infrastructure.

The demo highlights the end-to-end workflow, from measurements collection, threat prediction, definition of preventive strategies, pre-assessment of their impact in an emulated environment, and finally the deployment of the selected preventive strategies on the relevant nodes of the operational infrastructure. Figure 7 presents the complete workflow considered in this scenario.

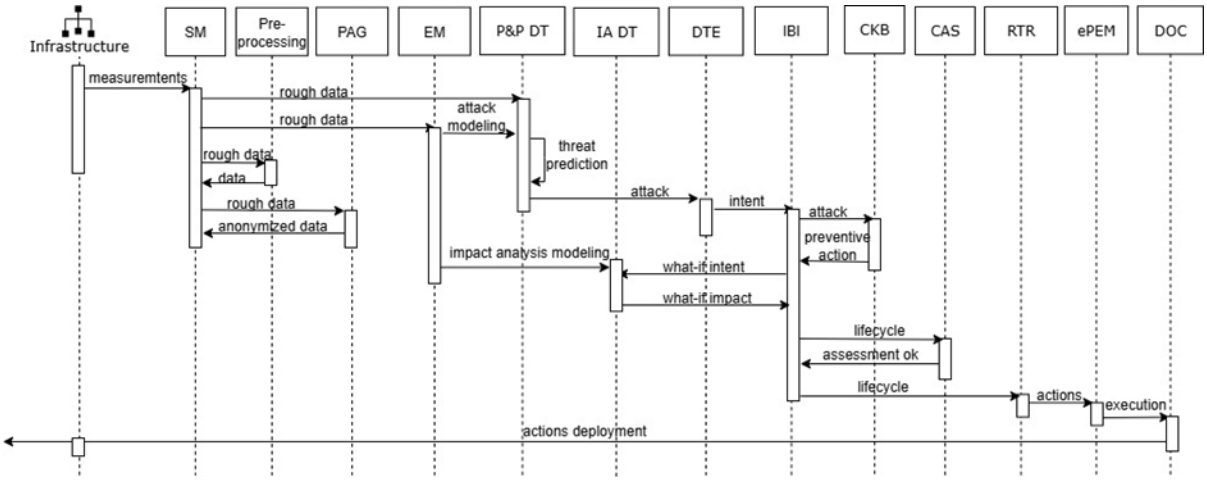

Figure 7. Demo 0 – Threat prediction workflow.

A complete video recording of this demo is available at the project GitHub repository. It provides a step-by-step walkthrough of the attack simulation, the prediction, the pre-assessment of the preventive strategy, and the enforcement in the real infrastructure, offering a clear visual representation of the HORSE platform's behavior and the interactions among its components.

Table 3 summarizes all interactions among the HORSE components involved in the demonstration. For each component, it provides an example of the corresponding input/output format, along with a visual capture illustrating the component's behavior during the demo execution. This unified view offers a clear understanding of how the different modules work together throughout the threat prediction workflow.

Table 3. Demo 0 – Threat prediction workflow HORSE components interactions.

Component From	Component To	Format Example
SM	P&P NDT	YAML File via HTTP request, which includes information regarding the infrastructure.

		name: horse-complete-2 nodes: - name: name1 vendor: HOST config: config_path: /home/name1 config_file: name1-config.py file: /configuracion/name1-config.py image: generic:latest interfaces: eth1: name: Ethernet1

SM	EM	{"ThreatModel":{
		"ThreatModelElement":{
		"ThreatActor":{"Source":"internet"},
		"CyberAttack":{
		"Type":"DDoS_Downlink",
		"Pattern":"",
		"Vector":{
		"AttackLocation":"dns-c1",
		"Parameter":{"Description":"DDoS Downlink on dns-c1","Protocol":"TCP","Flag":"SYN","Duration":"300"},
		"AttackTimestamp":"2025-03-20T10:48:44.613"},
		"ATT_CK":{"Type":"Network Denial of Service","ID":"T1498"}},
		"ControlAction":{
		"Mitigation":{
		"MitigationAction":{"Type":"FilterNetworkTraffic","ATT_CKID":"M1037"},
		"MitigationCondition":{"FilterCondition":{"SourceAddress":"192.168.1.100/32"},"isCNF":"false","type":"FilterNetworkTrafficCondition"}}},
		"id":"tme_6d3c88w5rt45873g98el723f7g92j63l"},
		"xmlns:xsi":"http://www.w3.org/2001/XMLSchema-instance",
		"id":"tm_8b2c65n3dr87345s54gd746b7h83t82k",
		"xsi:noNamespaceSchemaLocation":"threatModel.xsd"}}

EM	P&P NDT	<ThreatModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="threatModel.xsd" id="tm_8b2c65n3dr87345s54gd746b7h83t82k"> <ThreatModelElement id="tme_6d3c88w5rt45873g98el723f7g92j63l">

		<ThreatActor><Source>internet</Source></ThreatActor> <CyberAttack> <Type>DDoS_Downlink</Type><Pattern/> <Vector> <AttackLocation>dns-c1</AttackLocation> <Parameter><Description>DDoS Downlink on dns-c1</Description><Protocol>TCP</Protocol><Flag>SYN</Flag><Duration>300</Duration></Parameter> <AttackTimestamp>2025-03-20T10:48:44.613</AttackTimestamp> </Vector> <ATT_CK><Type>Network Denial of Service</Type><ID>T1498</ID></ATT_CK> </CyberAttack> <ControlAction> <Mitigation> <MitigationAction><Type>FilterNetworkTraffic</Type><ATT_CKID>M1037</ATT_CKID></MitigationAction> <MitigationCondition> <FilterCondition><SourceAddress>192.168.1.100/32</SourceAddress></FilterCondition> <isCNF>false</isCNF><type>FilterNetworkTrafficCondition</type> </MitigationCondition> </Mitigation> </ControlAction> </ThreatModelElement> </ThreatModel>
EM	IA NDT	<ThreatModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="threatModel.xsd" id="tm_8b2c65n3dr87345s54gd746b7h83t82k"> <ThreatModelElement id="tme_6d3c88w5rt45873g98el723f7g92j63l"> <ThreatActor><Source>internet</Source></ThreatActor> <CyberAttack> <Type>DDoS_Downlink</Type><Pattern/> <Vector> <AttackLocation>dns-c1</AttackLocation> <Parameter><Description>DDoS Downlink on dns-c1</Description><Protocol>TCP</Protocol><Flag>SYN</Flag><Duration>300</Duration></Parameter> <AttackTimestamp>2025-03-20T10:48:44.613</AttackTimestamp> </Vector>

		<ATT_CK><Type>Network Denial of Service</Type><ID>T1498</ID></ATT_CK> </CyberAttack> <ControlAction> <Mitigation> <MitigationAction><Type>FilterNetworkTraffic</Type><ATT_CKID>M1037</ATT_CKID></MitigationAction> <MitigationCondition> <FilterCondition><SourceAddress>192.168.1.100/32</SourceAddress></FilterCondition> <isCNF>false</isCNF><type>FilterNetworkTrafficCondition</type> </MitigationCondition> </Mitigation> </ControlAction> </ThreatModelElement> </ThreatModel>

P&P NDT	DTE	[
		{
		"prevention": "ddos_downlink",
		"confidence": 0.5
		}
		]

DTE	IBI	JSON document via HTTP POST in the following format:
		{
		"intent_type": "prevention",
		"threat": "ddos_downlink",
		"host": ["172.22.0.7"],
		"duration": 600
		}

IBI	cKB	JSON document via HTTP POST in the following format: {'attack_name': 'ddos_download_link'}

IBI	IA NDT	JSON document via HTTP POST in the following format:
		{
		"id": "3d6bcd2c-e3ce-44c0-9373-f07367a580c8",
		"topology_name": "horse_ddos",
		"attack": "DDoS_Downlink",
		"what-condition": {
		"KPIs": {
		"element": {
		"node": "dns-c1",
		"interface": "eth1"
		},
		"metric": "packets-per-second",
		"duration": "30s"
		}
		},
		"if-condition": {
		"action": {
		"type": "monitor",
		"value": "*",
		"unit": "*",
		"duration": "30s"
		},
		"element": {
		"node": "*",

		"interface": "*", "network": "*", "ref": "*_*_*" } } }

IA NDT	IBI	JSON Document via HTTP POST in the following format:
		{
		"id": "3d6bcd2c-e3ce-44c0-9373-f07367a580c8",
		"topology_name": "horse_ddos",
		"attack": "DDoS_Downlink",
		"what": {
		"KPIs": {
		"element": {
		"node": "dns-c1",
		"interface": "eth1"
		},
		"metric": "packets-per-second",
		"result": {
		"value": "20000",
		"unit": "packets-per-second"
		}
		}
		}
		}

IBI	IA NDT	JSON over HTTP POST:
		{
		"id": "3d6bcd2c-e3ce-44c0-9373-f07367a580c8",
		"topology_name": "horse_ddos",
		"attack": "DDoS_Downlink",
		"what-condition": {
		"KPIs": {
		"element": {
		"node": "dns-c1",
		"interface": "eth1"
		},
		"metric": "packets-per-second",
		"duration": "15s"
		}
		},
		"if-condition": {
		"action": {
		"type": "block_pod_ip",
		"value": "internet",
		"unit": "*",
		"duration": "30s"
		},
		"element": {
		"node": "ceos2",
		"interface": "eth1",
		"network": "*",
		"ref": "ceos2_eth1_*"
		}
		}
		}

IA NDT	IBI	JSON over HTTP POST:
		{
		"id": "3d6bcd2c-e3ce-44c0-9373-f07367a580c8",
		"topology_name": "horse_ddos",
		"attack": "DDoS_reverse",
		"what": {
		"KPIs": {
		"element": {
		"node": "dns-c1",
		"interface": "eth1"
		},
		"metric": "packets-per-second",

		"result": {
		"value": "5000",
		"unit": "packets-per-second"
		}
		}
		}
		}

IBI	CAS	JSON document over HTTP POST:
		{
		"input": {
		"command": "add",
		"intent_type": "prevention",
		"threat": "ddos_downlink",
		"attacked_host": [
		"172.22.0.7"
		],
		"mitigation_host": "ceos2",
		"action": {
		"name": "block_pod_address",
		"fields": {
		"blocked_pod": "attacker",
		"device": "ceos2",
		"interface": "eth4"
		}
		},
		"duration": "600",
		"intent_id": "a18d922d-5818-4820-91c3-0b19ed754d05"
		}
		}

CAS	IBI	JSON Answer in the HTTP Response in the following format: {allow: True} / {allow: False} and Response code: 200 for accepted and partially accepted, 500 for detected MITM

IBI	RTR	JSON Document via HTTP POST:
		{
		"command": "add",
		"intent_type": "prevention",
		"threat": "ddos_downlink",
		"attacked_host": "172.22.0.7",
		"mitigation_host": "ceos2",
		"action": {
		"name": "block_pod_address",
		"fields": {
		"blocked_pod": "attacker",
		"device": "ceos2",
		"interface": "eth4",
		"duration": 600
		}
		},
		"duration": 600,
		"intent_id": "a18d922d-5818-4820-91c3-0b19ed754d05"
		}

RTR	ePEM	{
		"command": "add",
		"intent_type": "mitigation",
		"intent_id": "34bea1d9-415c-48ac-b214-742cb6",
		"threat": "ddos_downlink",
		"target_domain": "",
		"action": {
		"name": "block_pod_address",
		"intent_id": "34bea1d9-415c-48ac-b214-66ec94742cb6",
		"fields": {
		"blocked_ips": [
		"172.22.0.7"
		],
		"device": "172.21.3.254",
		"interface": "eth4",
		"duration": 600

		} }, "attacked_host": "172.22.0.7", "mitigation_host": "172.21.3.254", "duration": 600, "status": "playbook_created, sent_to_epem", "info": "Playbook created, forwarding to ePEM in background, Sent to ePEM → ePEM sent to DOC → DOC successfully enforced action", "ansible_command": "---\n- name: Block IP addresses\n hosts: 172.21.3.254\n become: true\n tasks:\n - name: Block all incoming traffic from IP address\n iptables:\n chain: INPUT\n in_interface: eth4\n protocol: all\n source: 0.0.0.0/32\n jump: DROP\n when: ansible_os_family == \"Debian\" or ansible_os_family == \"RedHat\"\n\n handlers:\n - name: Reload firewall rules\n service:\n name: iptables # Replace with the name of your firewall service\n state: reloaded\n when: ansible_os_family == \"Debian\" or ansible_os_family == \"RedHat\"" }

ePEM	DOC	{
		"command": "add",
		"intent_type": "mitigation",
		"intent_id": "34bea1d9-415c-48ac-b214-742cb6",
		"threat": "ddos_downlink",
		"target_domain": "",
		"action": {
		"name": "block_pod_address",
		"intent_id": "34bea1d9-415c-48ac-b214-66ec94742cb6",
		"fields": {
		"blocked_ips": [
		"172.22.0.7"
		],
		"device": "172.21.3.254",
		"interface": "eth4",
		"duration": 600
		}
		},
		"attacked_host": "172.22.0.7",
		"mitigation_host": "172.21.3.254",
		"duration": 600,
		"status": "playbook_created, sent_to_epem",
		"info": "Playbook created, forwarding to ePEM in background, Sent to ePEM → ePEM sent to DOC → DOC successfully enforced action",
		"ansible_command": "---\n- name: Block IP addresses\n hosts: 172.21.3.254\n become: true\n tasks:\n - name: Block all incoming traffic from IP address\n iptables:\n chain: INPUT\n in_interface: eth4\n protocol: all\n source: 0.0.0.0/32\n jump: DROP\n when: ansible_os_family == \"Debian\" or ansible_os_family == \"RedHat\"\n\n handlers:\n - name: Reload firewall rules\n service:\n name: iptables # Replace with the name of your firewall service\n state: reloaded\n when: ansible_os_family == \"Debian\" or ansible_os_family == \"RedHat\""
		}

DOC	Infra	curl -X 'POST' \
		'http://10.19.2.19:8001/api/mitigate' \
		-H 'accept: application/json' \
		-H 'Content-Type: application/json' \
		-d '{
		"action": {
		"fields": {
		"duration": "600",
		"source_ip_filter": [
		"172.22.0.7"
		]
		},
		"intent_id": "34bea1d9-415c-48ac-b214-66ec94742cb6",

		"name": "block_pod_address" }, "command": "add", "info": "awaiting reinforcement", "intent_id": "34bea1d9-415c-48ac-b214-66ec94742cb6", "intent_type": "mitigation", "status": "pending", "testbed": "upc", "threat": "dns_attack" }'

4. HORSE Use Cases Validation

4.1. Use Case 1: Secure Smart LRT Systems (SS-LRT)

4.1.1. Description

EFACEC is a supplier of LRT (Light Rail Transit) systems to major operator's city transports operators and offers the EFARAIL solution integrating distinct subsystems like Public Information, network and communications, AVLS – Automatic Vehicle Location System, Video Surveillance, Regulation, Signaling Systems, etc.

The network and communications systems are responsible for interconnecting all systems handled by EFARAIL, including vehicles on the move and remote stations that can be spread along the railway network.

Typically, for city transport operators the network and communications to vehicles are supported by TETRA networks (that are very slow and low in performance and bandwidth capability), while the connection to remote stations uses fiber optics due to long distances to the control room.

In Use Case 1 the main goal consists of simulating congestion in the network by amplifying the volume of traffic through Denial of Services attacks (specifically, DDoS Downlink), verifying the impact on the Metro's subsystems as well as validating the HORSE policies and mitigation actions.

4.1.2. Context environment and deployment

Due to some restrictions and operational issues that EFACEC has encountered, basically focused on the fact that there is no real option to deploy the UC infrastructure in an operational scenario to be attacked, the deployment and management of the Use Case have been carried out by the team TID-UPC-UMU. EFACEC, under agreement, has entrusted their effort to that team to develop the Use Case, contributing to guarantee that the development is accurately mirroring the real infrastructure.

Specifically, the deployment of the EFARAIL solution was performed in 2 different testbeds, that communicates through a VPN:

- Virtual Machine running in UMU with PID – Public Information Display client application Simulator.
- Virtual Machine running in UPC with Vehicle Simulator application generating information that simulates vehicle movement for AVLS – Automatic Vehicle Location System.

This figure shows the environments integration between the two locations:

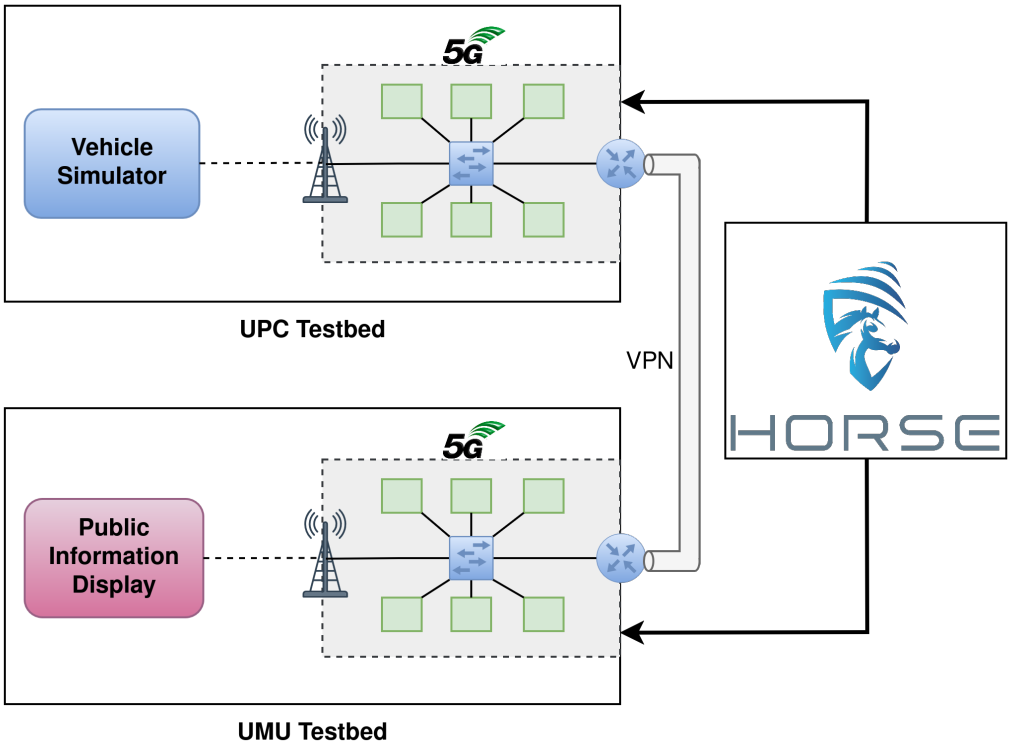

Figure 8. Use Case 1 integration.

The following figure shows the network architecture for both testbeds:

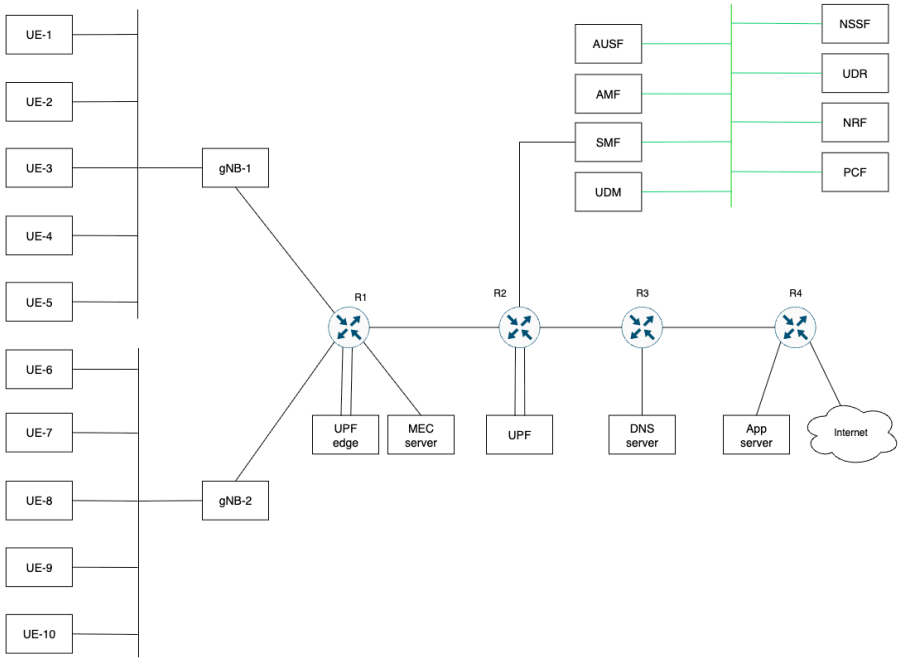

Figure 17: Use Case 1 Testbeds configuration.

In the testbed made available by UMU and UPC, all the HORSE components and the EFACEC virtual machines are installed. Those virtual machines simulate, one the PID – Public Information Display (@UMU) and another one the vehicles movement (@UPC).

The goal of this Use Case is to measure the impact on the LRT operation, when network congestion occurs, caused by an amplification attack, disturbing communication between the the tram stops and the vehicle information.

Two subsystems were deployed in the testbeds:

i. Public Information Display (PID).
ii. Automatic Vehicle Localization (AVLS)

And for this purpose, a simulator system of PID and a simulator system for AVLS have been developed.

Next figures show the PID simulator information shown in a specific station of the line considering normal performance (Figure 9) and when the system is under an attack (Figure 10).

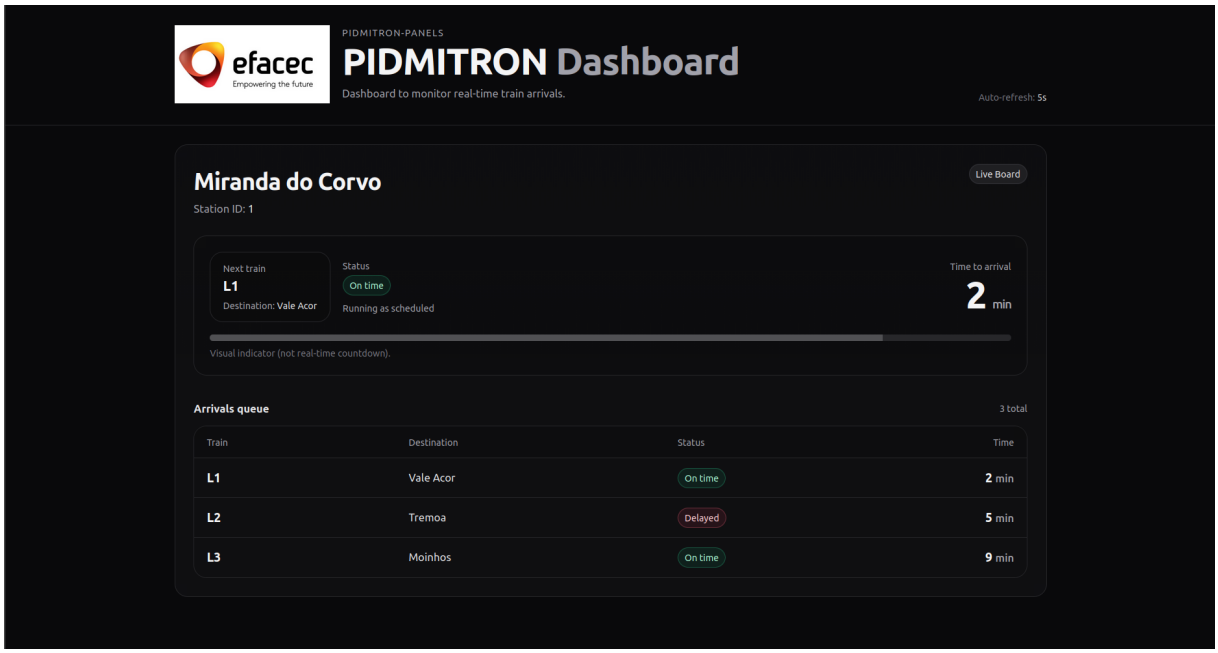

Figure 9. PID simulator in normal behavior.

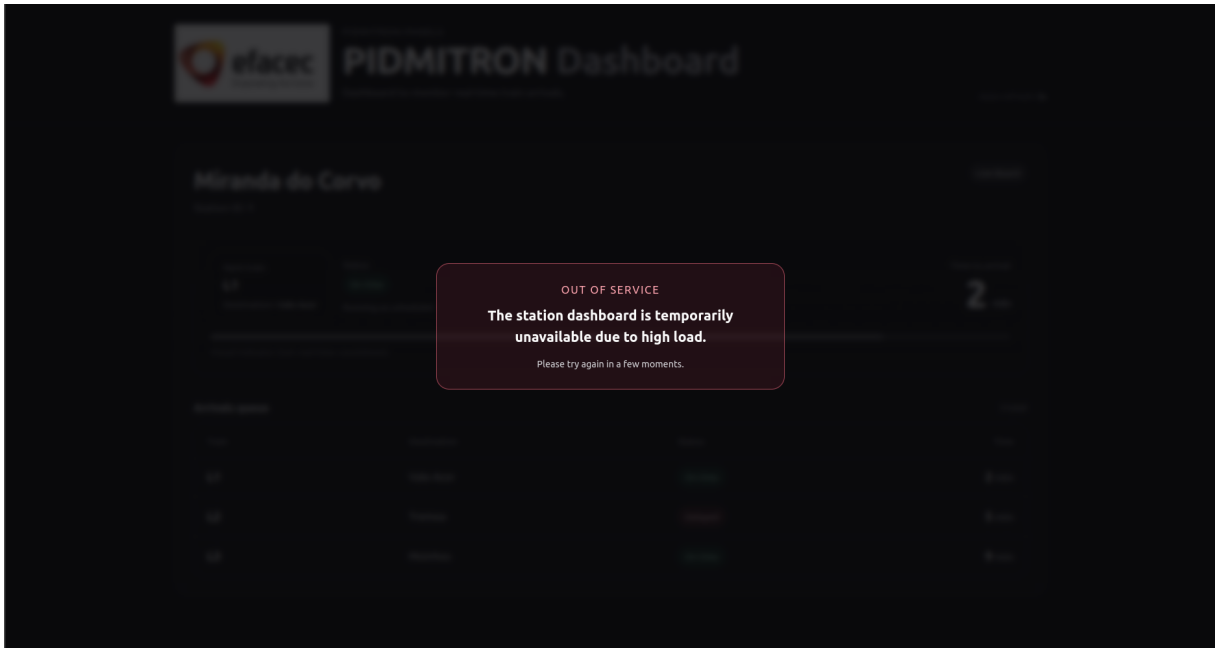

Figure 10. PID simulator under attack.

The PID Simulator aims to provide a user-facing panel that closely resembles what passengers would see on real platforms, enabling realistic testing and demonstrations under controlled conditions.

The simulator exposes a web dashboard and a REST API. The dashboard presents the live information for a specific station, including the upcoming train arrivals with their line/identifier, time-to-arrival (in minutes), and destination. The API is intended to be consumed by the Vehicle Simulator, located in UPC, which will continuously update—near real time—the operational state of trains associated with the station shown on the panel (e.g., minutes to arrival, status such as on-time/delayed/early, and the station's service availability when applicable).

The information displayed in the PIDs is managed by the simulator deployed at UPC. Each simulated train is modelled as an independent User Equipment (UE) connected to the 5G infrastructure, interacting with backend services and station systems. This approach enables the evaluation of service continuity, periodic reporting, and dynamic status updates under realistic operational conditions.

While a train is in circulation towards a station, the simulated UE periodically transmits operational updates with a one-minute granularity. These updates include:

The estimated time of arrival at the next station.

The current operational status of the train, classified as: On time, Early, or Late.

The reported estimated time of arrival is dynamically adjusted according to the operational status, allowing the simulation to reflect typical variations encountered in real railway operations, such as minor delays or early arrivals.

When the estimated time of arrival reaches zero, the train reports its arrival at the station. Upon arrival, two main operational scenarios are considered:

The station represents the end of the line, and the train terminates its journey, remaining at the station or leaving service.

The train continues its operation, performing a short dwell time at the station before resuming circulation and generating new estimated time of arrival updates.

4.1.3. KPI validation

The following table shows the KPIs defined to evaluate the HORSE benefits for a TRAM operator system:

Table 4. KPIs defined for Use Case 1.

Use case objective	KPI	Target
Resilience and disaster recovery	Down Time	Improve the down time in 50 %
Resilience and disaster recovery (remotely operation)	Availability	Improve the availability in 20 %
Decision support system	Statistics availability time	Capability to calculate the operation statistics data almost in real-time

The initial purpose of these KPIs was defined considering that HORSE validation would run on a real operational scenario. However, as mentioned before, several operational issues make this assumption to be not realistic, so the project team decided to go for an emulated scenario that, with the advice from EFACEC, was designed to be close to the reality. In this context the new scenario becomes an emulated one deployed in two real testbeds and without real traffic, what drive the need to adapt the validation process initially defined.

Therefore, considering this new layout, KPI values will be calculated comparing data gather from clients using EFARAIL (including solutions based on TETRA and WIFI) and data obtained from the HORSE platform running on UMU and UPC testbeds. In particular:

Resilience and disaster recovery down time KPI, pretends to compare the estimated time measured when the EFARAIL system is unresponsive due to DDoS, up to the system recovers after a mitigation action is in place.

Resilience and disaster recovery (remotely operation) availability KPI, pretends to compare the estimated time when the EFARAIL system remote operation is not available to control PID simulator messages due to DDoS attack performed to the PID simulator and the time that the operator needs to control the PID message using the remote operation EFARAIL system.

Decision support system objective for KPI Statistics availability time, pretends to compare the time that EFACEC current deployed solution in clients takes to show reports of vehicle operation performance using TETRA and WIFI technology against the much performant HORSE 5G/6G network with cyber-security focus environment.

KPI - Resilience and disaster recovery

From EFACEC experience in all client projects deployed all over the world, EFACEC service never experienced disruptions in TRAM operations due to cyber-attacks. Consequently, it is not possible to reproduce any cyber-attack in those clients because EFACEC is not authorized to interrupt or disrupt the normal operation of the TRAM line.

However, we can estimate that if a cyber-attack machine is connected in a remote TRAM station, flooding the fiber optics network with a DDoS attack, the detection of that attacker machine is to be completely manual. Therefore, even though the disruption of the operation is immediately seen in the EFARAIL application, the detection of the origin of the attack can take at least 1 hour using network switch and router supplier management tools. With the port of

the switch identified, where the flooding DDoS has been generated, we can take up to 5 minutes to disconnect the port from the network and up to 1 hour to find the attacker machine in the remote TRAM station.

So, with an estimated time of 2h 5m to fix the cyber-attack consequences, we evaluate in the UMU + UPC testbed for a similar attack and the results are around 2-10 minutes. This would give a KPI of 92-98% improvement in the downtime.

Resilience and disaster recovery (remotely operation)

The remote operation of the TRAM network is possible by using a VPN, but in case of a cyber-attack, like the one described, the consequences for the operation are the same. The estimation of the resolution of the cyber-attack consequences is very similar (around 2h 5m).

So, with an estimated time of 2h 5m to fix the cyber-attack consequences, we evaluate in the UMU + UPC testbed for a similar attack and the results are around 2-10 minutes maximum. This would give a KPI of 92-98% improvement in the downtime and availability.

Decision support system

Typically, EFARAIL solution uses TETRA communication along the line and WIFI in the TRAM parking to communicate between vehicles and OCC.

TETRA is a very low data rate communication used along the line and some packages can have a delay in reception on the server and correspondent storage in the database.

WIFI is present in the vehicles parking area and assures that at the end of the service of a vehicle, the travelling values are dispatched to the server and correspondent storage in the database. This is the reason why travel reports of the vehicles are made available only on the next morning, reporting values from the day before.

From EFARAIL solution with communication based in TETRA and WIFI, the results obtained on 17th November from daily operation are:

- Average communication delay between TRAM Vehicle and OCC = 12 m 31 s
- Maximum communication delay between TRAM Vehicle and OCC = 15 h 54 m
- Minimum communication delay between TRAM Vehicle and OCC = 5 s

To validate this KPI, as it is somehow open, we collected a similar file from UPC testbed and calculate the new Average, Maximum, and Minimum delay, giving us way better values that the former deployment, around 99 % (average).

4.2. Use Case 2: Remote Rendering to Power XR Industrial ($R^2$2XRI)

4.2.1. Description

The Remote Rendering for XR Industry 4.0 use case addresses secure, low-latency visualization and collaboration on high-fidelity industrial CAD and XR content over a 5G network infrastructure. In this scenario, computationally intensive XR applications are executed on powerful remote servers and streamed in real time to Augmented Reality glasses, enabling engineers, designers, and other industrial stakeholders to collaboratively inspect, review, and

discuss complex 3D data while being represented as avatars in a shared virtual environment. The use case supports geographically distributed collaboration, reduces the need for physical prototypes, and accelerates design and validation workflows, but it is highly sensitive to latency, jitter, and service disruptions that can severely degrade user experience and productivity. Because the streamed data and collaborative sessions involve confidential industrial information, cybersecurity is a critical enabler rather than an auxiliary feature. Within the HORSE project, this use case is integrated into a testbed where cyber-attacks and mitigation mechanisms are systematically introduced and evaluated, allowing HORSE's threat prediction, detection, and mitigation capabilities to enhance the resilience, security, and reliability of XR collaboration.

With the involvement of HORSE, Hololight can run multiuser sessions with smooth user experience and secure data streaming. The users are able to work together on a CAD model and interact with each other. With HORSE, Hololight sees great potential in scaling up the technology securely. HORSE delivers an integrated ecosystem where intelligent security policies, AI-driven decision making and real-world validation converge. The structured integration methodology ensures a technically robust, resilient, and deployment-ready platform. The integration with the HOLO XR Use Case validates the platform in critical applications that demands ultra-low latency and end-to-end security orchestration.

4.2.2. Context environment and deployment

Figure below illustrates the network topology of the CNIT testbed specifically configured for the Use Case 2. This setup is to allow the integration of HOLO (Holographic/Mixed Reality) devices.

The network can be broken down into three main sections: the UE with associated gNBs, the Edge/Core Network components, and the 5G Core/Control Plane.

The HOLO components are integrated at the edge of the network, close to the UEs.

- HOLO CIVR Glasses: this is a specific type of UE connected to gNB 2. This placement suggests a scenario where the holographic device requires low latency and high bandwidth, characteristic of 5G New Radio (NR) and Edge Computing.
- HOLO Server is strategically placed between the gNBs and the UPF Edge. It's connected to a switch/router that interfaces with gNB 0 and gNB 1. This server likely hosts the application or content required by the HOLO CIVR Glasses and potentially other UEs mixed reality services.

The design places the HOLO devices (HOLO CIVR Glasses) and the dedicated application server (HOLO Server) at the Edge of the 5G network. The HOLO Server is connected directly to the UPF Edge/gNB cluster, by-passing the central UPF Core for the data plane (user traffic) to minimize latency and maximize performance for demanding Mixed Reality applications.

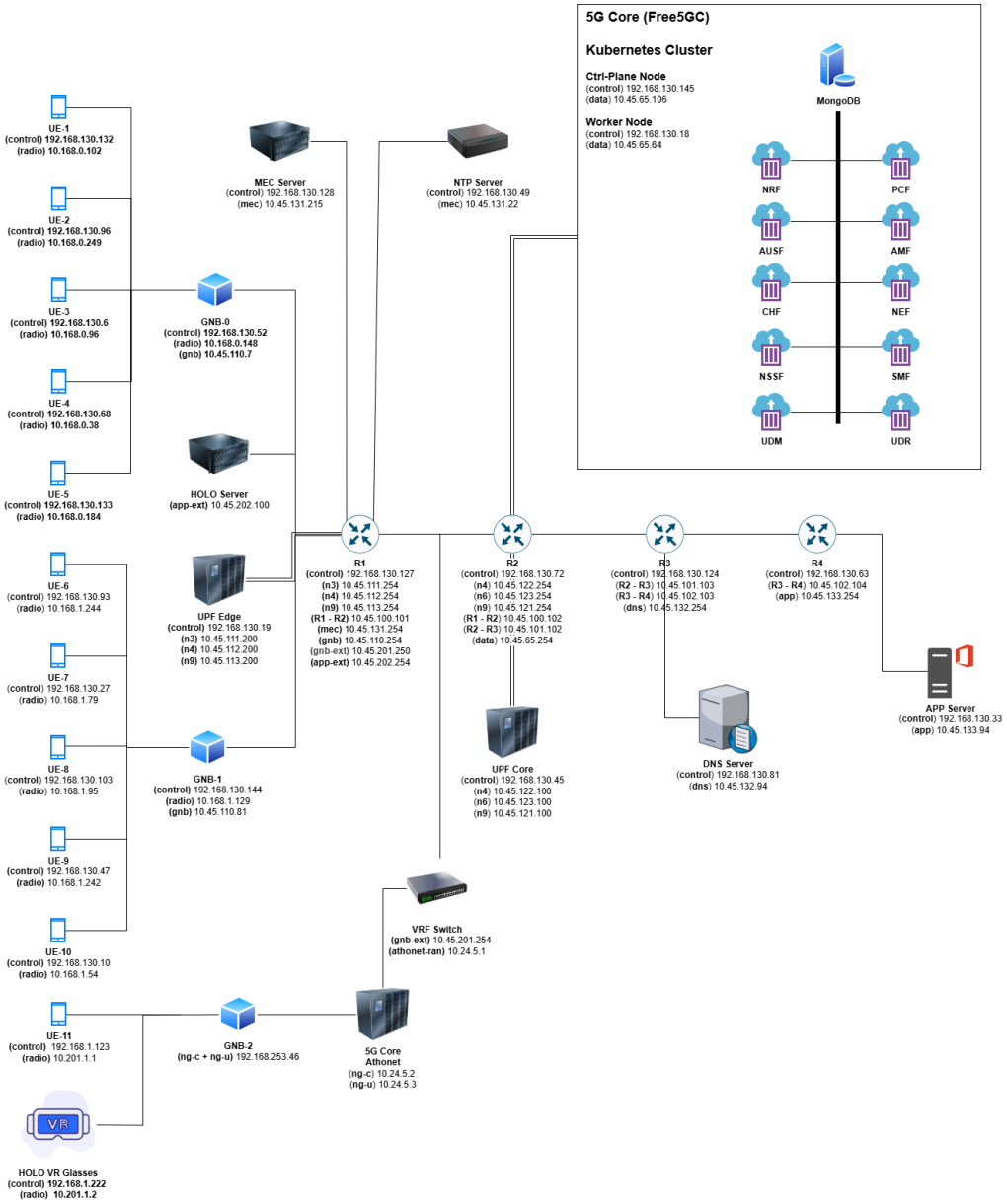

Figure 11. XR streaming in CNIT 5G testbed – Cyber-attack mechanism.

The selected attack for this demo is a Distributed Denial of Service (DDoS) attack carried out via a UDP Flood aimed specifically at the critical application server hosting the holographic/Mixed Reality service. This attack targets the Application Layer over the User Plane (UP) and exploits the high-speed connectivity of the 5G network.

This scenario highlights a common security challenge in 5G: a compromised UE can leverage the high bandwidth and low latency of the network to launch highly effective volumetric attacks against critical edge services like those supporting holographic communication.

The UDP flooding DDoS attack, directed from UE-11 to the HOLO App Server on UDP port 50100, significantly impacted the service's average latency across all three experimental runs.

The following table summarizes the minimum, maximum, and average latency observed during the designated "Under Attack" periods for each run.

Table 5. Latency performance during the experimental attacks in Use Case 2.

Run	Duration	Min. latency (ms)	Max. latency (ms)	Avg. latency (ms)
1	5 min 50 sec	100.57	343.55	148.19
2	6 min 20 sec	73.70	189.83	131.53
3	6 min 40 sec	90.29	188.94	138.86

Across all three runs, the UDP flooding attack successfully caused a sustained denial of service due to increased latency, pushing the average response time for the HOLO application well above the normal operating range, and in one instance, causing a catastrophic spike exceeding 343 milliseconds. This validates the attack's effectiveness in exploiting the edge-deployed HOLO service.

The HORSE platform, as detailed in D5.3, validates its capabilities for the Remote Rendering to Power XR Industrial (R22XRI) use case (UC2) through Demo 10. This demonstrator addresses two primary threats: API Exposure and a DDoS attack for network congestion, which specifically impacts the low-latency requirements of the XR rendering service. The HORSE framework employs a comprehensive three-step approach — Prediction, Detection, and Mitigation — to handle this DDoS attack.

4.2.3. KPI validation

The use case targets the following long-term Key Performance Indicators (KPIs):

1. Reduction of design faults by up to 90%.
   This KPI reflects the ability of XR-based visualization and collaboration to detect design errors earlier in the product development lifecycle, thereby reducing late-stage faults.

2. Reduction of prototyping costs by up to 50%.
   This KPI measures the extent to which virtual prototypes and XR-based design reviews can replace or reduce physical prototyping, logistics, and rework costs.

3. Reduction of time-to-market by up to 20%.
   This KPI captures improvements in development efficiency resulting from faster decision-making, early issue detection, and remote multi-user collaboration.

These KPIs are directly linked to industrial productivity, cost efficiency, and competitiveness and represent long-term organizational outcomes rather than short-term technical metrics.

Table 6. KPIs defined for Use Case 2.

Use case objective	KPI	Target
Detection of Design Fault	Decrease of errors	- 90 % faults
Cost reduction of prototypes	Reduction of costs	- 50 % costs
Faster time-to-market	Faster release of the products	+ 20 % faster release on the market

Nature of the KPIs: The targeted KPIs for this use case describe organizational and process-level impacts that emerge only through sustained, real-world use. Unlike low-level network or system KPIs such as latency, throughput, or packet loss, these indicators are strongly influenced by human decision-making, collaborative practices, and organizational workflows. They span multiple phases of the product lifecycle and depend on adoption maturity, experience, and integration into existing industrial processes. As a result, they cannot be reliably validated through short-term laboratory experiments or isolated technical measurements. Hence, the KPI validation is done through surveys of the user feedback and potential trends witnessed by the customers

Need for User- and Producer-Centric Evaluation: The primary beneficiaries of the XR remote rendering solution are industrial end users—such as engineers, designers, and operators—as well as the producers and integrators of the XR solution, including HOLO. These stakeholders are uniquely positioned to evaluate whether design faults are detected earlier, whether physical prototypes are avoided, and whether development cycles are shortened in practice. Their direct involvement in daily workflows and decision-making processes makes them the most credible sources for assessing the real impact of the solution. Consequently, structured self-reported assessments collected through questionnaires and interviews represent the most appropriate instruments for KPI validation.

Use of Scientific Questionnaires: Structured scientific questionnaires were selected as the primary evaluation method because they enable systematic and repeatable data collection across multiple organizations and roles. The use of Likert-scale items and comparative baseline questions support before-and-after assessments relative to non-XR workflows. In addition, open-ended questions provide qualitative insights that capture contextual and experiential factors that are otherwise unobservable through technical metrics alone. This mixed-method approach is well established in human–computer interaction research, industrial usability studies, and technology adoption research, ensuring methodological rigor and credibility.

Cybersecurity-Focused Assessment: In this use case, cybersecurity is not merely a technical requirement but a fundamental enabler of KPI achievement. Cyber-attacks can directly increase latency and jitter, disrupt XR sessions, and degrade collaborative performance, thereby negatively affecting productivity and user experience. Moreover, security breaches undermine trust in the technology, which can significantly limit industrial adoption. To capture this dimension, a dedicated producer-focused questionnaire was designed to assess the perceived importance of cybersecurity, the role of HORSE in ensuring secure and reliable XR services, and how threat prediction, detection, and mitigation mechanisms contribute to maintaining KPI performance under adverse conditions.

Results of the KPI validation

Two major subjects have been considered for the validation of the KPIs. The details can be found in annex, section 7.4. Conclusions are stated below:

1. Adoption of Hololight Stream and Hololight Space with cybersecurity enabled in industrial processes

   Every single respondent (5 out of 5) confirmed that XR remote rendering has improved product quality and development efficiency.

   The deployment of Hololight Stream and Space has proven to be highly effective for this cohort. The most pronounced ROI is found in Design Fault Detection (where performance is consistently high) and Prototyping Cost Reduction (where 80% of users save more than 25%). While Time-to-Market gains are modest for some (<10%), the overall efficiency and quality improvements make the solution a critical asset, provided cybersecurity requirements are met.

2. Evaluation of the HORSE Platform and Cybersecurity in XR Industry

The internal survey of HOLO leaders reveals a perfect alignment on the strategic importance of the HORSE platform. The data indicates that cybersecurity is not viewed merely as a compliance requirement but as a fundamental enabler of XR performance (latency/jitter) and market adoption. The leadership team unanimously validates that the HORSE platform provides the necessary threat prediction and mitigation capabilities to ensure operational stability in industrial and defense environments.

5. Conclusion

This deliverable shows the successful culmination of the HORSE platform development, delivering a mature, integrated solution that embodies the project's vision for intelligent, secure, and sustainable 6G networks. The final platform architecture and release demonstrate robust scalability, interoperability and management capabilities, validated through comprehensive demonstrations and real-world use cases.

6. References

[1] HORSE Deliverable 2.4: “HORSE Landscape and Architectural Design”. https://horse-6g.eu/?wpdmdl=801&ind=1751979687243

[2] HORSE Deliverable 5.3: “Second HORSE Release: HORSE IT-2 version”. https://tntlabunigeit-my.sharepoint.com/:b:/g/personal/horse-cloud_tnt-lab_unige_it/IQAEXAFICvOfR6FAfkqd0f2RAUtGI_jGtyL-sNP5CIGxS8Q?e=0poZdv

7. Annex

7.1. Train operation logs for UC1

Below is an example of the logs generated by one of the trains in operation, reporting its operational status, its arrival at the station, and the fact that it reaches the end of the line and terminates the service.

root@a37929fe1fa5:/efacec_train# python3 basic_train.py L1

==========

🚆 EFACEC TRAIN INFORMATION PROGRAM - Train L1

==========

📍 Getting available stops...

URL: http://192.168.150.16/api/stops

✅ Stops fetched successfully:{"stops":[{"id": 1,"name": "Miranda do Corvo"}],"outOfService": false}

==========

🚂 STARTING TRAIN SIMULATION L1

==========

🚂 Getting trains for stop: 1

URL: http://192.168.150.16/api/stops/1/arrivals

✅ Train information fetched successfully:{"stop": {"id": 1,"name": "Miranda do Corvo"},"arrivals": [{"trainId": "L1","minutes": 2,"status": 0,"destination": "Vale Acor"}, {"trainId": "L2","minutes": 5,"status": 1,"destination": "Tremoa" },{"trainId": "L3","minutes": 9,"status": 0,"destination": "Moinhos"}]}

🔍 Train L1 found at Miranda do Corvo (2 min(s) to arrival)

🚄 Train L1 traveling to Miranda do Corvo

[11:27:42] 🚂 Train L1 → Miranda do Corvo: 2 min(s) remaining

✏️ Updating train L1 at stop 1

URL: http://192.168.150.16/api/stops/1/arrivals/L1

Payload: {"minutes": 2,"status": 0}

Status: on time

Status Code: 200

✅ Train updated successfully:

{"minutes": 2,"status": 0,"stopId": 1,"trainId": "L1"}

[...]

[11:29:42] 🚂 Train L1 → Miranda do Corvo: 0 min(s) remaining

✏️ Updating train L1 at stop 1

URL: http://192.168.150.16/api/stops/1/arrivals/L1/

Payload: {"minutes": 0,"status": 0}

Status: on time

Status Code: 200

✅ Train updated successfully:

{"minutes": 0,"status": 0,"stopId": 1,"trainId": "L1"}

⏰ [11:29:42] ✅ Train 011 ARRIVED at Miranda do Corvo 🛑 Delete train 011 from Miranda do Corvo and end

🗑️ Deleting train L1 from stop 1

URL: http://192.168.150.16/api/stops/1/trains/L1Status Code: 200

✅ Train deleted successfully

7.2. Detailed Latency Analysis During Attack for UC2

7.2.1. Run 1

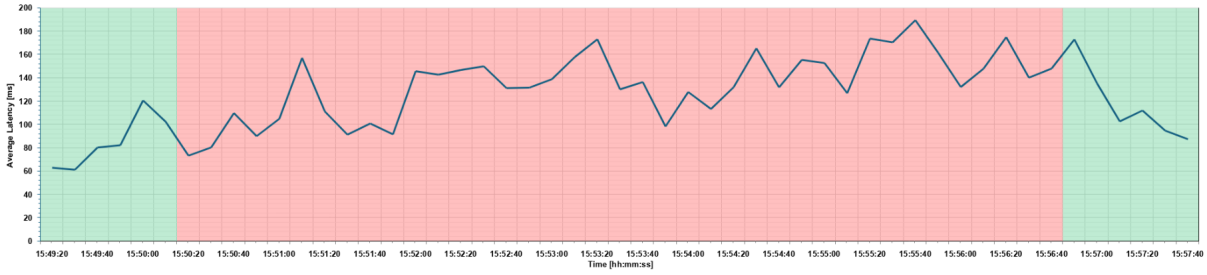

Figure 12. Latency analysis for run 1 in UC2.

- Average Latency During Attack: 148.19 ms
- Peak Impact: This run recorded the single highest latency spike of all experiments at 343.55 ms, demonstrating the most severe disruption caused by the flood traffic.
- Attack Profile: The latency was highly volatile, with several spikes above 200 ms (including 343.55 ms, 222.19 ms, 316.36 ms, and 250.08 ms) in the initial phase of the attack, indicating periods where the HOLO App Server's resources were completely overwhelmed.

7.2.2. Run 2

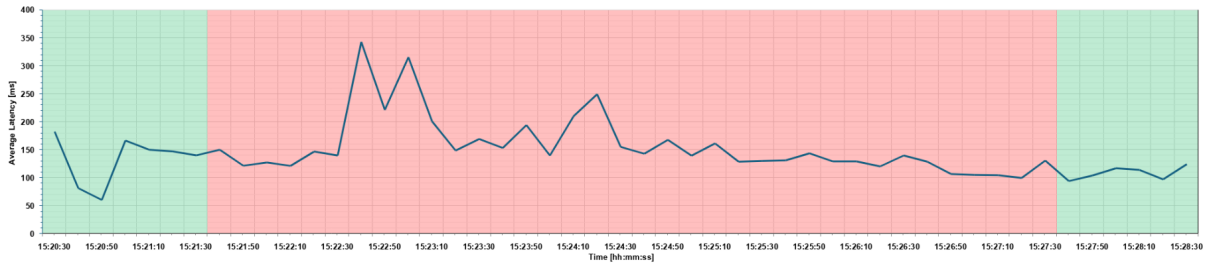

Figure 13. Latency analysis for run 2 in UC2.

- **Average Latency During Attack: 131.53 ms**
- **Peak Impact:** The maximum latency was **189.83 ms**.
- **Attack Profile:** Although the average latency was the lowest of the three runs, the attack successfully maintained the latency at a consistently elevated level, rarely dropping below 120 ms (after the initial moments) and reaching nearly 190 ms, which is significantly higher than the preceding normal traffic period (where latency was typically between 60 ms and 120 ms).

7.2.3. Run 3

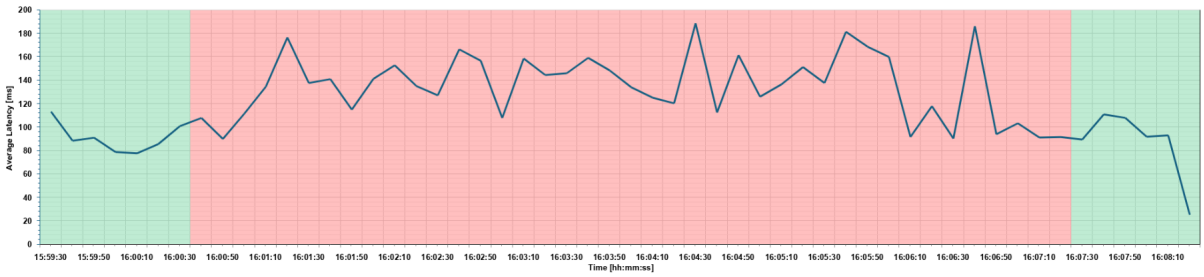

Figure 14. Latency analysis for run 3 in UC2.

- **Average Latency During Attack: 138.86 ms**
- **Peak Impact:** The maximum latency was **188.94 ms**.
- **Attack Profile:** Similar to Run 2, this run shows a sustained high-latency state. The normal period preceding the attack had values as low as **25.61 ms** (at 16:08:20, post-attack) and generally below 100 ms, meaning the attack successfully *increased the minimum latency* of the affected period to over 90 ms and capped the performance, preventing the low latency required for real-time applications like the HOLO service.

7.3. Customer survey for UC2

Survey title: Adoption of XR Remote Rendering into industry process

Target respondents: Product owners, R&D managers, engineering leads

Instructions: - Please answer based on your experience after adopting **Hololight Stream and Hololight Space**. When applicable, compare with previous projects or workflows without XR. - All responses are confidential and used only for research and project evaluation purposes.

Name: Company:

Date: Email:

Section 1 – Respondent Profile

1. Your role in the organization:
   - ☐ Product / Project Manager
   - ☐ R&D Manager
   - ☐ Engineering Lead
   - ☐ Other: _____
2. How long has your organization been using the XR remote rendering solution?
   - ☐ < 3 months
   - ☐ 3 – 6 months
   - ☐ 6 – 12 months
   - ☐ > 12 months
3. In which product lifecycle phase is XR mainly used? (select all that apply)
   - ☐ Concept design
   - ☐ Engineering & validation
   - ☐ Prototyping
   - ☐ Production

Section 2 – Design Fault Detection

Please indicate your level of agreement: (1 = Strongly disagree, 5 = Strongly agree)

4. XR-based visualization enables earlier detection of design faults than previous workflows.
   ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
5. Collaborative XR reviews improve the accuracy of design decisions.
   ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
6. XR reduces the occurrence of late-stage design errors.
   ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
7. With XR, the error detection in design is
   - ☐ < 10%
   - ☐ < 50%
   - ☐ > 50%
   - ☐ > 90%

Section 3 – Prototyping Cost Reduction

Please indicate your level of agreement:

8. XR reduced the need for physical prototypes.
   ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
9. XR-based reviews decreased the number of prototype iterations.
   ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
10. XR collaboration reduced costs related to travel, logistics, or rework.
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
11. Estimated reduction in overall prototyping costs due to XR:

☐ 0 – 10%
☐ 10 – 25%
☐ 25 – 50%
☐ > 50%

Section 4 – Time-to-Market

Please indicate your level of agreement:

12. XR collaboration accelerates design and validation cycles.
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
13. Faster issue identification using XR shortens overall development time.
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
14. Estimated change in time-to-market after XR adoption:
☐ Slower
☐ No change
☐ < 10% faster
☐ 10 – 20% faster
☐ > 20% faster

Section 5 – Security, Reliability

Please indicate your level of agreement:

15. Confidential industrial data is adequately protected during XR sessions.
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
16. Network latency and stability are sufficient for professional XR collaboration.
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
17. Cybersecurity incidents did not disrupt XR-based workflows.
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
18. How important is cybersecurity for XR adoption in your organization?
☐ Low
☐ Medium
☐ High
☐ Critical
19. **How much does your entire evaluation depend on the cybersecurity aspect of the tool?**
☐ 0 – 25%
☐ 25% – 50%
☐ 50% – 75%
☐ 75 – 100%

Section 6 – Final Evaluation

20. Overall, XR remote rendering has improved product quality and development efficiency.

☐ Yes
☐ No

21. Please briefly describe the most significant measurable benefit of XR in your organization:

Thank you for your participation. Your responses directly contribute to the scientific validation of Hololight Stream, Hololight Space and the potential benefits of cybersecurity developed by the HORSE project.

Cybersecurity Impact Survey from the producers of the XR software

Survey title: Cybersecurity Impact on XR use cases (HOLO Perspective)

Purpose: This short survey captures the perspective of XR solution producers (HOLO) on the importance of cybersecurity and the contribution of the HORSE platform to achieving the XR Industry 4.0 KPIs: - Reduction of design faults - Reduction of prototyping costs - Faster time-to-market

Target respondents: HOLO technical leads, system architects, product owners, sales executives.

Instructions: Please answer based on your experience developing, integrating, and operating the XR remote rendering solution within the HORSE framework.

Name: Designation at HOLO:

Date: Email:

Cybersecurity & KPI Enablement

Please indicate your level of agreement: (1 = Strongly disagree, 5 = Strongly agree)

1. Cybersecurity is critical for achieving the XR use case KPIs (quality, cost, and time-to-market).
   ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
2. Security-related incidents can directly degrade XR user experience (e.g., latency increase, jitter, session disruption).
   ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
3. Without advanced cybersecurity mechanisms, industrial adoption of XR remote rendering would be limited.
   ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
4. The HORSE platform improves the resilience and operational stability of XR remote rendering services.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

5. HORSE's threat prediction, detection, and mitigation capabilities help maintain XR KPI performance under cyber attack conditions.
   ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Final Assessment

6. Overall, HORSE is essential for enabling secure, low-latency XR collaboration in industrial environments.
   ☐ Yes
   ☐ No
7. In your view, how does HORSE support the achievement of XR Industry 4.0 KPIs?

Thank you for your participation.
Your input supports the scientific evaluation of HORSE's impact on secure XR applications.

7.4. Results of the KPI validation for Use Case 2

7.4.1. Adoption of Hololight Stream and Hololight Space with cybersecurity enabled in industrial processes.

Sample Size

5 Respondents (Product Owners, R&D Managers, Engineering Leads) from companies including Lockheed Martin, Trumpf, FYWARE, Riederbau, and Oberlechner Immo.

Respondent Demographics

The survey targeted professionals deeply integrated into the product lifecycle. The majority of respondents have been utilizing the XR solution for a medium-term period (6-12 months).

Roles

Product/Project Managers (40%), R&D Managers (20%), Engineering Leads (20%), Other (20%).

Key Findings: Quantitative Analysis

- Design Fault Detection

There is a unanimous consensus that the XR visualization through Hololight Stream drastically improves error detection. 100% of respondents estimated that this enables the detection of over 50% of design errors. Participants consistently rated the ability of this technology to enable early fault detection and reduce late-stage errors as very high (Ratings of 4 or 5 out of 5).

- Prototyping Cost Reduction

All respondents reported a reduction in prototyping costs, with the vast majority seeing significant savings (above 25%).

- 20% reported a reduction of 10–25%.
- 60% reported a reduction of 25–50%.
- 20% reported a reduction of > 50%.

– Time-to-Market Acceleration

XR adoption has universally accelerated development cycles, though the degree varies between "moderate" (<10%) and "significant" (10-20%).

- 60% reported time-to-market is < 10% faster.
- 40% reported time-to-market is 10–20% faster.

– Security and Reliability

Security is a paramount concern for the respondents, with the majority classifying it as "Critical."

- Importance of Cybersecurity:
  - 60% rated it as Critical.
  - 40% rated it as High.

- Evaluation Dependency: For 40% of respondents, the entire evaluation of the tool depended 75–100% on the cybersecurity aspect, highlighting that security is a "make or break" feature.

– Qualitative Benefits (Respondent Feedback)

When asked to describe the most significant measurable benefit, respondents focused on speed, collaboration, and specific use cases:

- Collaboration & Review: "Collaborative Design Review improves customer meeting and brings much better understanding".
- Speed: "Faster design reviews, without waiting for physical prototypes".
- Maintenance: "Augmented Reality based Maintenance".
- Planning: "Using XR in facility planning and error detection in CAD designs".

7.4.2. Evaluation of the HORSE Platform and Cybersecurity in XR Industry

Respondent Profile

The survey collected data from high-level leadership and technical experts at HOLO to assess the importance of cybersecurity in Extended Reality (XR) and the effectiveness of the HORSE platform. The respondents included the CEO, CTO, COO, CSO, Head of Product Development and XR experts.

Quantitative Analysis: Agreement Levels

Respondents were asked to rate five key statements on a Likert scale from 1 (Strongly Disagree) to 5 (Strongly Agree).

Summary of Results:

There was unanimous consensus among all six respondents. Every respondent selected "Strongly Agree" (5) for every question posed regarding the criticality of cybersecurity and the utility of the HORSE platform.

Metric / Survey Question	Average Score (1-5)	Consensus
Q1. Criticality: Cybersecurity is critical for achieving XR use case KPIs (quality, cost, time-to-market).	5.0	100%
Q2. User Experience: Security incidents directly degrade XR UX (latency, jitter, disruption).	5.0	100%
Q3. Market Adoption: Without advanced security, industrial adoption of XR remote rendering is limited.	5.0	100%
Q4. Resilience: HORSE platform improves resilience and operational stability.	5.0	100%
Q5. Mitigation: HORSE's threat detection helps maintain KPIs under attack.	5.0	100%

Final Assessment: HORSE Platform Viability

- Question: "Overall, HORSE is essential for enabling secure, low-latency XR collaboration in industrial environments."
  - Response: Yes - 6 respondents (100%)

All executives and technical leads confirmed that the HORSE platform is essential for secure, low-latency XR collaboration.

Qualitative Insights

Respondents provided specific feedback on how the HORSE platform supports XR Industry 4.0 KPIs. Some statements from the respondents are listed below.

- "Horse platform provides a fast and easy to deploy/maintain system to deal with XR cybersecurity topics at enterprise level in a reliable way.
- "Network threat detection and mitigation in 5G is critical in industrial and defense immersive usecases and HORSE has set up a good starting point"
- “Higher security increases usecase adoption."