Grant Agreement No.: 101096342
Call: HORIZON-JU-SNS-2022
Topic: HORIZON-JU-SNS-2022-STREAM-B-01-04
Type of action: HORIZON-JU-RIA

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

D2.4 HORSE Landscape and Architectural Design

Revision: v1.0

Work package	WP 2
Task	T2.1, T2.2, T2.3, T2.4
Due date	31/12/2024
Submission date	20/12/2024
Deliverable lead	Universitat Politècnica de Catalunya (UPC)
Version	1.0
Authors	Eva Rodriguez (UPC), Xavi Masip (UPC), Jordi Forne (UPC), Josep Llosa (UPC), Leesa Joyce (HOLO), Alice Piemonti (MARTEL), Vito Cianchini (MARTEL), Nikos Nomikos (NKUA), Panagiotis Gkonis (NKUA), Panagiotis Trakadas (NKUA), Stefanos Venios (SUITE5), Andreas Petrou (SUITE5), Iulisloi Zacarias (TUBS), Chukwuemeka Muonagor (TUBS), Admela Jukan (TUBS), Anthony Joel Pogo Medina (UMU), Emilio García de la Calera Molina (UMU) , Sofia Giannakidou (STS), Alexandros Katsarakis (STS), Jose Manuel Manjón (TID), Fabrizio Granelli (CNIT), Orazio Toscano (ETI), Paulo Paixão (EFACEC), Pedro Elísio (EFACEC) Manuel Angel Jimenez Quesada (ATOS), George Xylouris (ZORTE), Michalis Danousis (8BELLS), Paulo Paixão (EFACEC), Pedro Elísio (EFACEC)
Reviewers	Jose Manuel Manjón (TID), Fabrizio Granelli (CNIT)

Abstract	D2.4 HORSE Landscape and Architectural Design is a public document that describes the final version of the HORSE architecture, including its building blocks, the interaction among HORSE modules, and the communication involved in these interactions. The current document is the second iteration of two deliverables. It begins by revisiting the evolution of the HORSE architectural design, starting with the reference architecture, progressing through the first iteration, and concluding with the final version defined in the second iteration. The architectural design is shaped by key applications, relevant technologies, and current standards, resulting in the final framework that will guide the development tasks in WP3 and WP4. Additionally, the document describes the communication between components, including two distinct workflows to check the final functionality of all modules and determine the data flow. We finalize by describing how the two project use cases will benefit from the HORSE framework, detailing how they interact with and utilize the HORSE components.
Keywords	HORSE Architecture; Components; Driving applications; Current Standards; Communication; Use cases mapping; Workflows

DOCUMENT REVISION HISTORY

Version	Date	Description of change	List of contributor(s)
V0.1	08/10/2024	1st version of the template for comments	Eva Rodríguez (UPC), Xavi Masip (UPC)
V0.2	14/10/2024	1st version of the Table of Contents	Eva Rodríguez (UPC), Xavi Masip (UPC), Fabrizio Granelli (CNIT)
V0.3	11/11/2024	Updates in the HORSE system description	Eva Rodríguez (UPC), Xavi Masip (UPC), Jordi Forne (UPC), Josep Llosa (UPC), Jose Manuel Manjón (TID), Orazio Toscano (ETI), Stefanos Venios (SUITE5), Andreas Petrou (SUITE5), Sofia Giannakidou (STS), Alexandros Katsarakis (STS), Anthony Joel Pogo Medina (UMU), Manuel Angel Jimenez Quesada (ATOS)
V0.4	18/11/2024	Updates in the Architectural Design	Eva Rodríguez (UPC), Xavi Masip (UPC), Jordi Forne (UPC), Josep Llosa (UPC), Fabrizio Granelli (CNIT), Manuel Angel Jimenez Quesada (ATOS), Jose Manuel Manjón (TID), Orazio Toscano (ETI), Iulisloi Zacarias (TUBS), Chukwuemeka Muonagor (TUBS), Admela Jukan (TUBS), Nikos Nomikos (NKUA), Panagiotis Gkonis (NKUA), Panagiotis Trakadas (NKUA), Stefanos Venios (SUITE5), Andreas Petrou (SUITE5), Alice Piemonti (MARTEL), Vito Cianchini (MARTEL), George Xylouris (ZORTE), Sofia Giannakidou (STS), Alexandros Katsarakis (STS), Michalis Danousis (8BELLS)
V0.5	22/11/2024	Contributions to HORSE components	Eva Rodríguez (UPC), Xavi Masip (UPC), Jordi Forne (UPC), Josep Llosa (UPC), Fabrizio Granelli (CNIT), Manuel Angel Jimenez Quesada (ATOS), Jose Manuel Manjón (TID), Orazio Toscano (ETI), Iulisloi Zacarias (TUBS), Chukwuemeka Muonagor (TUBS), Admela Jukan (TUBS), Nikos Nomikos (NKUA), Panagiotis Gkonis (NKUA), Panagiotis Trakadas (NKUA), Stefanos Venios (SUITE5), Alice Piemonti (MARTEL), Vito Cianchini (MARTEL), George Xylouris (ZORTE) , Sofia Giannakidou (STS), Alexandros Katsarakis (STS), Michalis Danousis (8BELLS)
V0.6	29/11/2024	Updates in the Canonical Workflows	Eva Rodríguez (UPC), Xavi Masip (UPC), Jordi Forne (UPC), Josep Llosa (UPC), Fabrizio Granelli (CNIT), ATOS, Jose Manuel Manjón (TID), Orazio Toscano (ETI), Iulisloi

			Zacarias (TUBS), Chukwuemeka Muonagor (TUBS), Admela Jukan (TUBS), Nikos Nomikos (NKUA), Panagiotis Gkonis (NKUA), Panagiotis Trakadas (NKUA), Stefanos Venios (SUITE5), Alice Piemonti (MARTEL), Vito Cianchini (MARTEL), George Xylouris (ZORTE), Sofia Giannakidou (STS), Alexandros Katsarakis (STS), Michalis Danousis (8BELLS)
V0.7	29/11/2024	Updates in the Use cases	Leesa Joyce (HOLO), Paulo Paixão (EFACEC), Pedro Elísio (EFACEC)
V0.8	06/12/2024	Version for internal peer review	Eva Rodríguez (UPC), Xavi Masip (UPC)
V0.9	13/12/2024	Revision from TID	Jose Manuel Manjón (TID)
V0.10	13/12/2024	Revision form CNIT	Fabrizio Granelli (CNIT)
V0.11	17/12/2024	Version for QA	Eva Rodríguez (UPC), Xavi Masip (UPC)
V1.0	20/12/2024	Quality assessment and final version to be submitted.	Fabrizio Granelli (CNIT)

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:	R	
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.

Executive summary

This document represents the final version (iteration IT-2) of the HORSE architectural design, aimed to develop an autonomous, self-evolving and extendable 6G-ready architecture providing a human-centric approach to security workflows by enabling end-to-end security solutions. This document is the final outcome of the work conducted in WP2 tasks, with a particular focus on "T2.4 – Architectural design".

The deliverable starts by outlining the evolution of the HORSE architectural design, tracing its development from the initial reference version of the architecture to the final specifications achieved in iteration IT-2. This iterative process was essential to ensure the successful deployment of the HORSE platform. Additionally, advancements in driving applications, emerging technologies, and relevant standards are reviewed to reassess and validate the decisions underpinning the proposed architecture.

Section 3 of this deliverable presents the final HORSE architectural design developed during IT-2, highlighting the key modifications made from the IT-1 version. The architecture preserves its three core components, the Intent-based Interface (IBI), Platform Intelligence (PIL) and the AI Secure and Trustable Orchestration (STO). The IBI simplifies network configuration and operation by receiving high-level intents from the network manager or software agents, and leveraging advanced AI techniques, proposes policies to optimize network performance. The PIL module enhances network management by adding intelligence capable of detecting and predicting network security threats. Finally, the STO module ensures reliable network operation by enforcing the policies proposed by the IBI and effective network resources orchestration. The final version of the HORSE architecture introduces two new functionalities, including the monitoring and tracking of connected devices; as well as providing an interactive overview of the 5G/6G network status and security-related events. This includes details on detected and predicted attacks, implemented mitigation and preventive measures, impact analysis, and actionable recommendations. The final HORSE architecture will act as a unifying framework for task coordination within other technical work packages, which will implement the functional components envisioned in this architectural design.

The canonical HORSE workflows for threat detection and prediction have been enhanced to define the complete process, incorporating all the modules of the architecture. The first workflow is designed to illustrate the detection of network threats and the immediate reaction of the HORSE platform. In contrast, the second workflow makes use of Network Digital Twin (NDT) techniques to predict threats and assess the impact of the preventive actions in the NDT, before being enforced in the real infrastructure.

The deliverable provides an updated architectural design that improves upon previous versions of the framework, while ensuring alignment with the latest advancements in 6G, cybersecurity, and AI.

Table of contents

List of figures

List of tables

Abbreviations

3GPP	3rd Generation Partnership Project
5G	Fifth Generation of Wireless Cellular Technology
6G	Sixth Generation of Wireless Cellular Technology
AE	AutoEncoder
AFs	Autonomic Functions
AI	Artificial Intelligence
API	Application Programming Interface
AR	Augmented Reality
B5G	Beyond Fifth Generation of Wireless Cellular Technology
CAD	Computer-aided Design
CAS	Compliance Assessment
CISO	Chief Information Security Officer
CKB	Common Knowledge Base
CRA	Cyber Resilience Act
CSP	Communication Service Providers
CSV	Comma-Separated Values
DDoS	Distributed Denial-of-Service
DEME	Detector and Mitigation Engine
DevOps	Development Operations
DFP	Dynamic Function Placement
DTE	Distributed Trustable AI Engine
Dx.x	Deliverable x.x
E2E	End-to-end
EM	Early Modelling
ENISA	European Union Agency for Cybersecurity
ePEM	End-to-end (E2E) Secure Connectivity Manager
FL	Federated Learning
GANA	Generic Autonomic Networking Architecture
GUI	Graphical User Interface
HRF	HORSE Reference Framework
HTTP	Hypertext Transfer Protocol

IBI	Intent-based Interface
IT-1	Iteration 1 of HORSE Architecture Task
IT-2	Iteration 2 of HORSE Architecture Task
IT-X	Iteration X
JSON	JavaScript Object Notation
KPIs	Key Performance Indicators
MEC	Multi-access Edge Computing
ML	Machine Learning
MSE	Mean Square Error
NDT	Network Digital Twin
NFV	Network Function Virtualization
NOCs	Network Operations Centers
OSS	Operations Support System
PAG	Policies and Data Governance
PCAP	Packet Capture
DEME	Threat Detector and Mitigation Engine
PIL	Platform Intelligence
PoC	Proof of Concept
RAN	Radio access networks
REST	Representational State Transfer
RestAPI	Representational State Transfer (API)
RESTful API	Representational State Transfer (API)
RL	Reinforcement Learning
RTR	Reliability, Trust and Resilience
SAN	Sandboxing
SDK	Software Development Kit
SDN	Software Defined Network / Networking
SDO	Standard Development Organizations
SM	Smart Monitoring
STIX	Structured Threat Information Expression
STO	AI Secure and Trustable Orchestration
VIM	Virtual Infrastructure Manager
VNF	Virtual Network Function

WPx	Work Package x
XR	Extended Reality
YANG	Data Model for Network Topologies

1 Introduction

1.1 Purpose of the document

This document presents the final architectural design of the HORSE framework, building on the initial design conducted during the first iteration (IT-1), as documented in “D2.2 HORSE Architectural Design (IT-1)” [1], and incorporating the IT-2 requirements specified in “D2.3 HORSE Landscape: Technologies, state of the art, AI policies and requirements (IT-2)” [2]. It outlines the key updates made to the initial architectural design and to the HORSE framework components.

Leveraging insights gained during IT-1 from the practical implementation of HORSE framework components in WP3 and WP4, along with the integration efforts conducted in WP5, this document proposes revised solutions for individual components. These updates include enhancements to communication protocols and interface designs. Furthermore, the canonical workflows for threat detection and prediction, initially defined in IT-1, have been refined to align with the improved functionalities and interactions introduced in this iteration (IT-2).

The present document outlines the final version of the HORSE architecture, designed to deliver cutting-edge security solutions. These include predictive threat detection, impact analysis, and proactive threat mitigation, all while addressing disaggregation, software-based paradigms, and incorporating elements of automation and intelligence. The proposed architecture serves as a reference framework to guide activities in the technical work packages.

1.2 Structure of the document

This document is organized into several sections to present the collected information clearly and effectively. This section provides an overview of the document's structure, making it easier to navigate and locate specific content.

The document is structured as follows:

- Section 2: This section analyses the evolution of the HORSE architectural design and its limitations in IT-1. Furthermore, it reviews the advancements in driving applications, emerging technologies, and relevant standards over the past 15 months, which have motivated the decisions behind the proposed architecture.
- Section 3: This section presents the final HORSE architecture for IT-2, highlighting the key modifications made from the IT-1 version. It also provides a detailed description of the main modules within the architecture, along with an overview of their internal components.
- Section 4: This section outlines the final canonical workflows for threat detection and prediction, to be aligned with the improved functionalities and interactions introduced in the final version of the HORSE architecture.
- Section 5: In this section, two use cases are mapped to the final HORSE architecture, illustrating its interaction with the HORSE components.

2 HORSE system description

This section provides a comprehensive overview of the evolution of the HORSE architecture, starting with the initial version outlined during the proposal phase of the project. This is followed by a discussion of the architecture's refinement in D2.2, leading up to the final version in IT-2, which is presented in section 3. In this context, we also highlight the key limitations identified in the version released in IT-1. Specifically, we revisit and expand upon the critical areas of concern introduced in IT-1, providing further insights into how these issues have been addressed in the final version of the architecture.

Additionally, this section describes the key applications, technologies, and organizational aspects of the infrastructure that influenced the design of the final HORSE architectural blueprint, as well as the cyber resilience requirements and constraints that guided its development. Finally, it presents the relevant standards to be integrated into the HORSE framework

2.1 Evolution of the architectural design in IT-1

The preliminary version of the HORSE architecture presented in the project proposal is shown in Figure 1. It consisted three building modules which included: i) the AI Secure and Trustable Orchestration (STO); ii) the Platform Intelligence (PIL), and iii) the Intent-based Interface (IBI).

The STO module is responsible for endowing the 6G infrastructure with the performance, reliability and trust functionalities necessary to correctly orchestrate resources and deploy smart services. The PIL module comprises the whole set of intelligent strategies and mechanisms responsible for both supporting the predictive approach objective of HORSE and serving as interface to existing orchestration solutions. Finally, the IBI module is responsible for guaranteeing easy user engagement into the overall landscape.

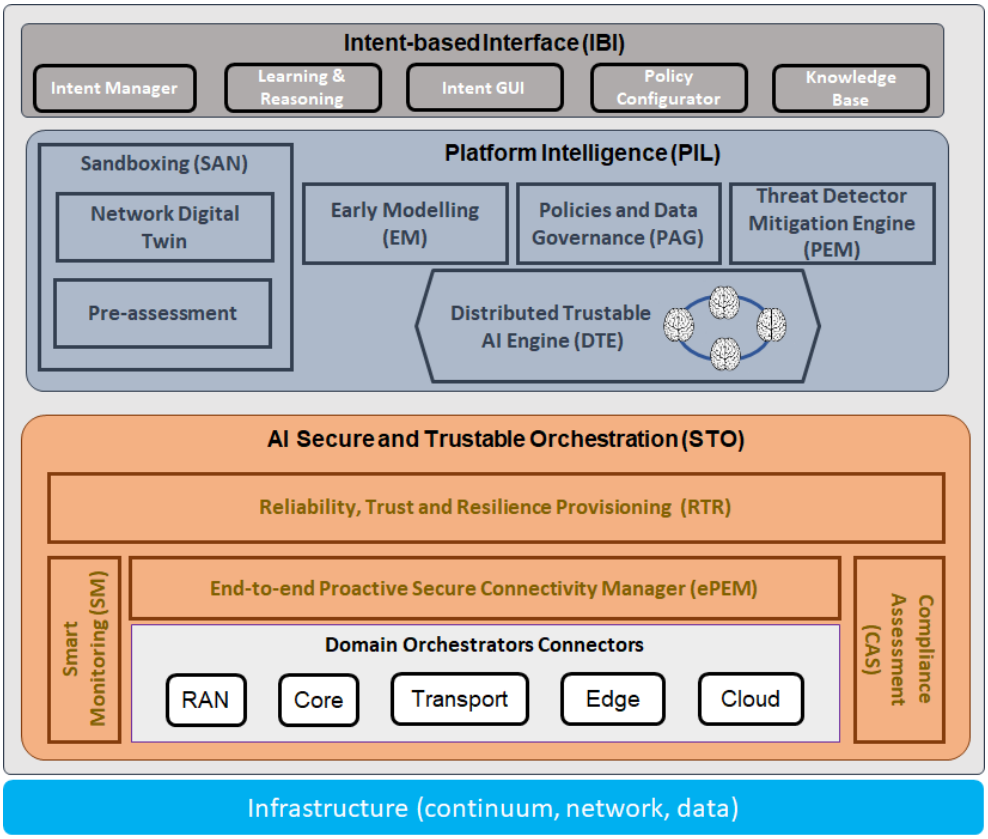


Figure 1: HORSE reference architecture

During the first revision of the architecture in IT-1, some initial changes were made from the original proposal, resulting in the architecture shown in Figure 2. The main changes can be summarized as:

- The Pre-processing component is added into the STO module to unify and standardize the data collected by the SM component. This ensures that the data meets the necessary requirements for proper handling before being fed into the smart modules.
- Two "contexts" are defined in the PIL module, one real and another one emulated. The DEME component works in the real context, being responsible for detecting threats in real time. While the SAN works in an emulated environment being responsible for predicting threats.
- Two complementary NDTs are considered in the SAN. The Prediction & Prevention NDT predicts anomalies and threats in the emulated context and the Impact Analysis NDT determines the impact of applying the mitigation and preventive measures in the different 6G/5G emulated components.
- The DTE component includes a Recommender element to create the intents, i.e., high-level descriptions of the actions to be taken.
- The IBI module process the received high-level intents and generates the corresponding workflow (lifecycle) to be taken by the RTR, first double checking whether the estimated impact is acceptable, and the policies are aligned with the decision to be taken.
- The RTR component is responsible for defining the set of actions to be executed by the ePEM that trigger the final execution to the connectors.
- The CAS component is responsible for verifying on the real infrastructure that the set of actions to be taken are aligned to the policies defined by the IBI.

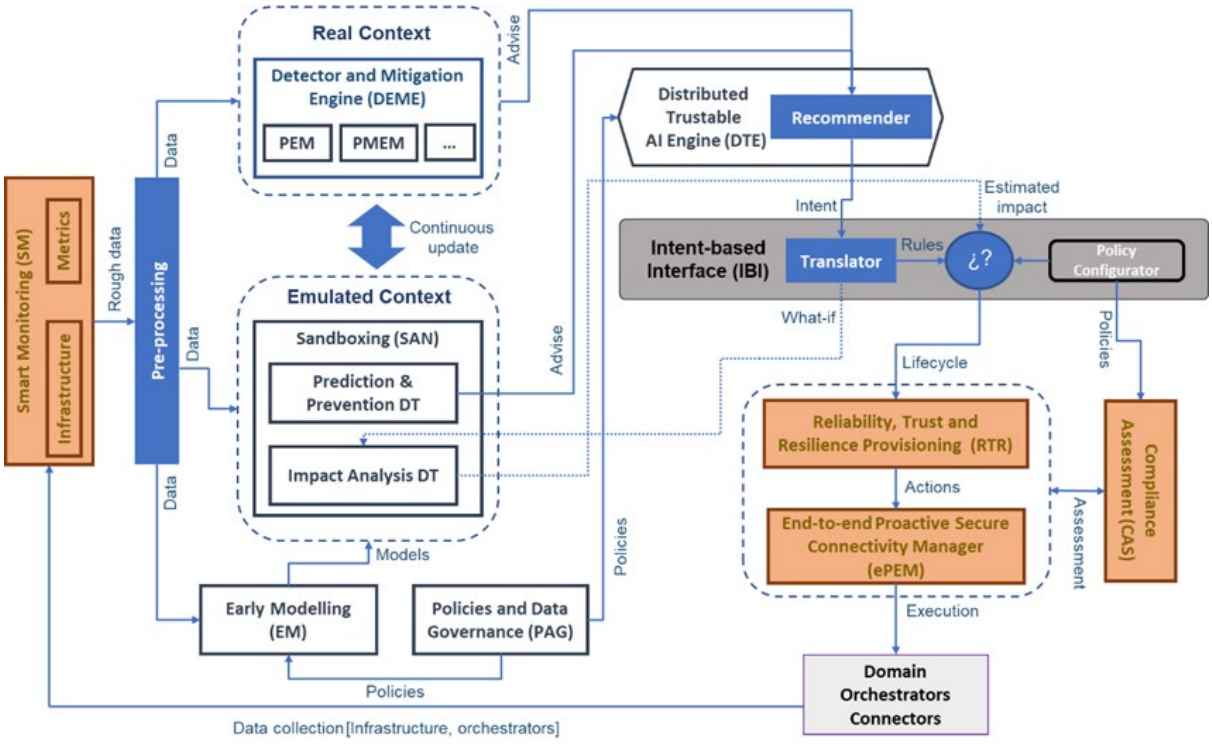


Figure 2: HORSE architecture IT-1 version 1

The activities conducted in WP3 and WP4 focused on the development of the HORSE components, together with the integration efforts in WP5, resulted in updates to the architecture. The main changes can be summarized as follows:

- The SM component includes the Elasticsearch module to efficiently store the data collected by the SM and pre-processed by the Pre-processing module. This data feds the AI-based HORSE components.
- A centralized database, so-called Common Knowledge Base (CKB), stores and provide essential information on attack mitigations and preventive actions. This database feeds the IBI and EM modules.

Figure 3 presents a detailed view of the HORSE architecture in IT-1 after the development and integration process, while Figure 4 provides a high-level overview of the final version of the architecture in IT-1, highlighting its main components.

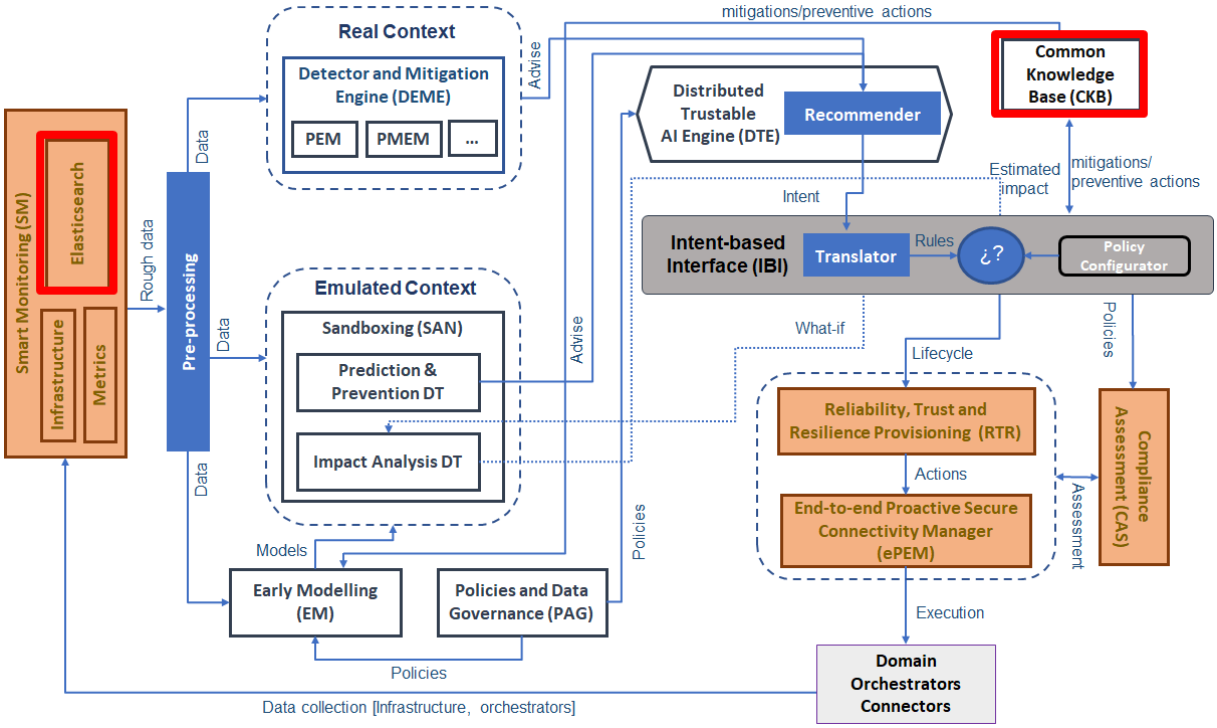


Figure 3: HORSE architecture IT-1 version 2

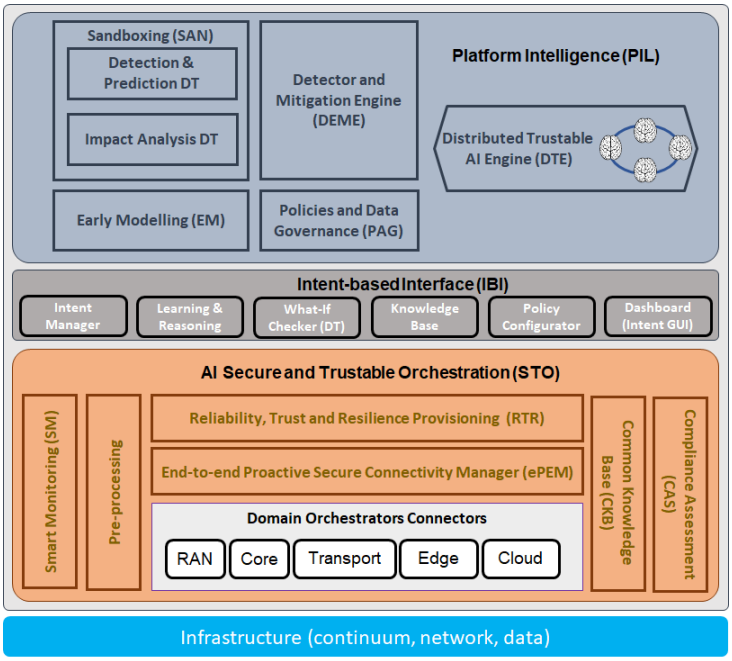


Figure 4. HORSE architecture IT-1 version 2 – Main components

2.2 Limitations of IT-1 implementation and architectural design

Certain components were not fully implemented during IT-1, so that their final implementations and modifications were left for the second iteration of the project. These final implementations complete the functionalities offered by the HORSE framework, being the main changes with respect to IT-1 described next.

The HORSE platform will provide mechanisms to determine the location of UEs within the network and share this information through a centralized API, while also monitoring the connectivity of the connected devices to the managed. This functionality is addressed by the new component, UEs Tracking, defined within the SM.

Access control mechanisms will be defined by the PAG component to ensure that access to the datasets stored in the Elasticsearch is granted exclusively to authorized HORSE components.

The HORSE platform will display the overall network status, including key metrics such as latency, throughput, and packet loss. Additionally, it will provide insights into detected or predicted attacks and anomalies, along with the status of ongoing mitigation and preventive actions. This functionality is provided by the new Dashboard component defined within the IBI.

The HORSE platform incorporates the capability to learn and reason from decisions made by human operators. When a decision is escalated to a human operator for resolution, the system will learn from the decision taken and apply the same reasoning when same situation repeats.

The DTE component provides functionalities for decentralized training of ML models, and it enables the dynamic update of the ML models.

In the second version of the IT-1 HORSE architecture, the CKB component was integrated to manage the mitigation and preventive strategies. In IT-2, it will leverage advanced generative AI techniques to automatically generate new mitigation and preventive strategies, which will enhance the CKB. In addition, the HORSE platform will enable the prioritization of these strategies through ranking mechanisms.

Finally, the HORSE policies will be rigorously evaluated to ensure compliance with the criteria established by relevant regulatory frameworks and standards. These assessments will consider guidelines and requirements set forth by authoritative bodies such as 3GPP and ENISA.

2.3 Platform structure and action areas

As a first approach to integrate the HORSE topology and the HORSE modules, we have developed three testbeds with 5G and 6G capabilities able to deploy the components of the HORSE project. These testbeds have been explained previously in Work Package 5 deliverables and are the ones owned by UMU, CNIT and UPC. Indeed, this last one is being promoted thanks to the project.

Once the deployment of the first topology has succeeded for the IT-1, a new topology approach has been agreed for the IT-2, including edge capabilities an application server and an extra gNode B for the users. This enhanced topology is the one shown in Figure 5.

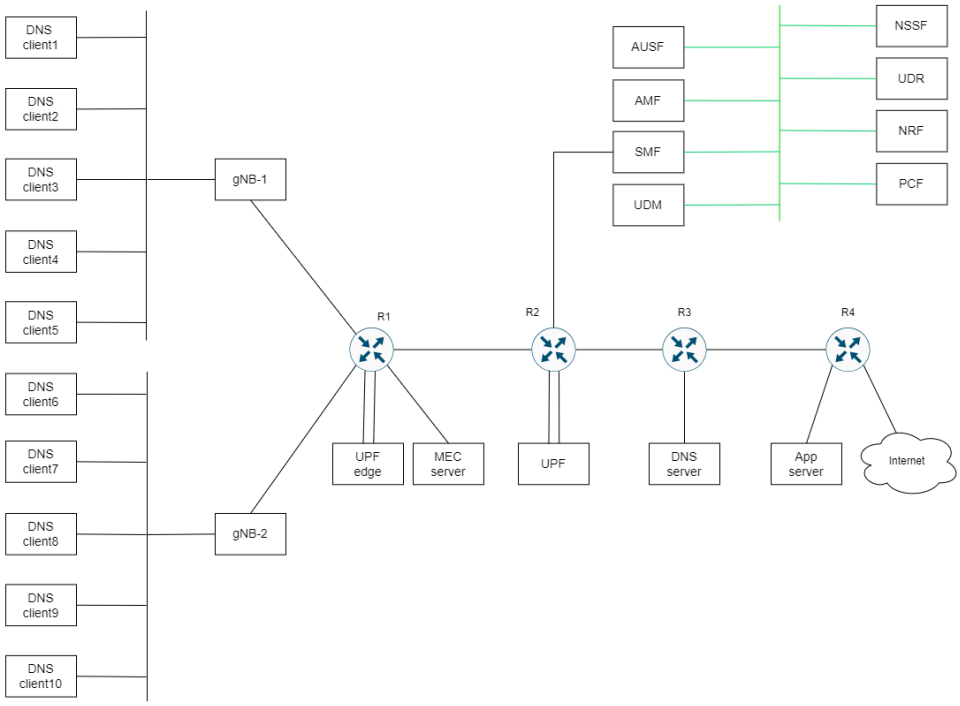


Figure 5. New network topology for IT-2.

Also, the uses cases will be integrated in the testbed to have a fully integrated scenario with the network, the use cases and the HORSE modules. Regarding use case 1 (Secure Smart LRT Systems), it will be deployed in the UMU testbed, and it will be connected to the UPC testbed, which will deploy the network and the HORSE modules. For the use case 2 (Remote Rendering to Power XR Industrial), it will be integrated in the CNIT testbed, that will deploy the network, the use case and the HORSE modules. The specifications and requirements associated to the use cases are described in deliverable 2.3 [2].

2.4 Related standards

In an effort to quickly benefit from their 5G investments, communication service providers (CSPs) often opt for tailored solutions, believing this will expedite market entry. However, this

strategy can lead to accumulating technical debt and result in a more complex and less manageable system. To counter these challenges, guidelines from Standard Development Organizations (SDOs) are invaluable, as they facilitate synergies and support plug-and-play capabilities, leading to innovative business models and creating ecosystems that are flexible, scalable, and interoperable. Looking forward to a sophisticated, adaptable, and expandable architecture prepared for 6G, it is essential to follow the key SDOs outlined in the initial version of this document, which are conveniently listed in Table 1 for easy reference.

SDO or Forum	Full name	Link
IEEE	Institute of Electrical and Electronics Engineers	https://www.ieee.org/
IETF	Internet Engineering Task Force	https://www.ietf.org/
ETSI	European Telecommunications Standards Institute	https://www.etsi.org/
ITU	International Telecommunication Union	https://www.itu.int/
3GPP	3rd Generation Partnership Project	https://www.3gpp.org/
NGMN	Next Generation Mobile Networks Alliance	https://www.ngmn.org/
BBF	Broadband Forum	https://www.broadband-forum.org/
TMF	TM Forum	https://www.tmforum.org/
ENISA	European Union Agency for Cybersecurity	https://www.enisa.europa.eu/

Table 1: SDOs

Table 2 provides an overview of the primary activities within the scope of the Horse project, emphasizing the differences between the current deliverable submission and the previous one:

SDO or Forum	Working Group or Framework	Activity	Updates
IEEE	NGSON WG (P1903 standards)	Service overlay networks as the main abstraction level for autonomics via embracing context awareness and self-organization capabilities.	No significant update
	INGR SysOpt WG	Outlines standardization items and approach for enhancing standards on autonomics in other SDOs/fora.	

IETF	ANIMA (e.g., RFC 8993)	Defines a reduced-scope Autonomic Networking (AN) with progressive introduction of autonomic functions (AFs). No implementation specifications for coordination among AFs.	New drafts recently edited and other in progress about: Extensions to the ANI, including variations of ANI deployment (e.g. in virtualised environments), information distribution within an AN, ANI OAMP interfaces (Operations, Administration, Management, Provisioning), interaction with YANG-based mechanisms, defining the domain boundary and membership management of the domain. Support for Autonomic Service Agents, including design and implementation guidelines for ASAs, life cycle management, authorization and coordination of ASA. BRSKI features, including proxies, enrollment, adaptions over various network protocols, variations of voucher formats. Generic use cases of Autonomic Network and new GRASP extensions/options for them, including bulk transfer, DNS-SD interworking, autonomic resource management, autonomic SLA assurance, autonomic multi-tenant management, autonomic network measurement. Integration with Network Operations Centers (NOCs), including autonomic discovery/connectivity to NOC, YANG-based ANI/ASA management by the NOC and reporting AF from node to NOC.
	NTF	Architectural framework for network telemetry. Protocols to gather monitoring data for full visibility.	
	AINEMA	Architectural framework for integrating AI in network management operations. Algorithms to operate AI, information model to represent AI data and decisions, and protocols to exchange them.	Activities are progressing. Eg. draft-pedro-nmrg-ai-framework-05 “Artificial Intelligence Framework for Network Management “ 24 April 2025 involves D.R. Lopez (TID) among authors
ETSI	TC NTECH/AFI WG and TC INT/AFI WG (e.g., ETSI TS 103 195-2 and White Paper #16)	GANA model and its instantiations onto various types of fixed, mobile and wireless networks. Running a 5G PoC to implement some GANA aspects.	Please see below
	ETSI TC CYBER, ETSI NFV SECURITY	Cross-domain cybersecurity, Mobile/Wireless systems (3G/4G, TETRA, DECT, RRS, RFID...), IoT and Machine-to-Machine (M2M), Network	

		Functions Virtualisation, Intelligent Transport Systems, Maritime Broadcasting, Lawful Interception and Retained Data, Digital Signatures and trust service providers, Smart cards / Secure elements, Exchangeable CA/DRM solutions, Security algorithms	
	ENI ISG	Defines an AI-based architecture to help external systems improve their environmental awareness and adapt accordingly. Envisions the translation of input data as well as output recommendations/commands.	
	ZSM ISG	Reuses existing standards and frameworks into a holistic design to achieve E2E automation in multi-vendor environments using AI-based data collection and closed-loop control.	
	SAI	Creates standards to preserve and improve the security of AI technologies, whether used in small and personal devices, as when AI is used to optimize complex industrial processes. The standards aim to secure AI from attacks, mitigate attacks created by AI (when AI is used to improve conventional attacks), and use AI to enhance security actions.	
ITU	SG13	Rec. ITU-T Y.3324: defines the functional and architectural requirements of autonomic management and control (AMC) for IMT-2020 networks. Rec. ITU-T Y.3177: specifies a high-level architecture of AI-based automation of future networks including IMT-2020. FG-AN: builds upon existing standards' gaps to standardize autonomous networks.	Activities of SG13 are in progress covering deep packet inspection, distributed SDN, Quantum Key distributions, Cloud Computing, Future Networks, Fixed, mobile and satellite convergence etc.
3GPP	Release 16 (e.g., TR 28.861)	Introduction to 5G NR-SON and further slicing management.	No significant update

	Release 18	Enhancement of data collection for 5G NR-SON.	Rel 18 finalized in mid 2024
NGMN	5G E2E architecture framework v3.0.8	Describes a high-level vision of architecture principles and requirements to guide other SDOs/Fora and promote interoperability. Its automation capabilities are based on the ETSI GANA model.	New activites mainly related to Automation and Autonomous Systems amd Green Future Networks
BBF	AIM	Builds on GANA and ITU Rs. to define autonomic functions (AFs) for access and E2E converged fixed/mobile networks.	Two key documents on AIM and disaggregation published with more specifications close to fruition for SDN/NFV
TMForum	ODA (e.g., IG1167 and IG1177)	Mapping of the ETSI GANA framework to the ODA intelligence management model.	No significant update

Table 2: SDOs Activities in the HORSE scope

2.4.1 ETSI Working Programme 2024-2025

The ETSI provided its 2024 work programme on Cybersecurity [3], offering valuable insights into how Standard Development Organizations (SDOs) are tackling the impending challenges posed by new technologies and related threats.

Summarizing, it covers the integral need for security and privacy in our digital lives, highlighting the increasing complexity of maintaining ICT security amidst evolving threats. The focus is on challenges posed by technologies like IoT, virtualization, cloud computing, and generative AI. Privacy concerns are rising, leading regulators like the EU to impose stringent requirements, with the upcoming EU Cyber Resilience Act (CRA) mandating standards for manufacturers and service providers. ETSI's Cybersecurity Technical Committee (TC CYBER) [4] is crafting standards and guidance in collaboration with international and regional bodies, preparing to finalize their support for the CRA by early 2024. The committee also emphasizes consumer IoT security, planning an updated standard in 2024. It addresses expanded guidance on cybersecurity controls via the revised Network and Information Security (NIS2) Directive. Furthermore, the rise of quantum computing, with its implications for cryptography, is being tackled by the CYBER QSC Working Group, which is developing quantum-safe solutions and aiming to complete several related deliverables in 2024.

2.5 CyberSecurity Specifications

In the specific context of CyberSecurity, comprehensive analysis [5] of the most important standards and frameworks can be summarized in Table 3.

Name	Stands for	Scope
ETSI TC CYBER	European Telecommunications Standards Institute Technical Committee on Cybersecurity	Cybersecurity Framework

NIST CSF	The National Institute of Standards and Technology-Cybersecurity Frameworks	Cybersecurity Critical Infrastructures
NIST SP800-207		Zero Trust Architecture
NIST 5G/6G		5G/6G Core Networks and Services
CISA	Cybersecurity and Infrastructure Security Agency	Zero trust maturity model
O-RAN Alliance WG11	O-RAN Alliance	ORAN security requirement specification
3GPP SA3 and SA5	Third Generation Partnership Project	Security & Privacy Management & Orchestration
IETF	Internet Engineering Task Force	Certificates management, Data transit protection, AAA
ENISA	5G Cybersecurity Standards	Cybersecurity threats and vulnerabilities standards.
	5G Security Controls Matrix	Recommendations for 5G telecommunication networks operation as part of 5G toolbox.
	NFV Security in 5G - Challenges and Best Practices	Security challenges and attacks to the Network Function Virtualization (NFV) in the 5G network.
	Threat Landscape for 5G Networks Report	Analyzes 5G network security challenges with input from industry experts and public sources.
	EU Cybersecurity Act	Activities related to the setting up and maintaining the European cybersecurity certification framework.
ISO/IEC 27032 27001	ISO/IEC	Guidelines for Cybersecurity Information Security Managements Systems
NIS 2 Directive	Network and information systems	Cybersecurity risk management
CSA CCM	Cloud Security Alliance's Cloud Controls Matrix	Cloud Security

Table 3: CyberSecurity Standards and Frameworks

For more references about CyberSecurity local regulations and CyberSecurity for Industry-Specific standards please refer to [5].

2.6 Datasets

This section outlines the HORSE Datasets creation process. It starts by detailing how data is collected in the three testbeds of the project: UMU, CNIT and UPC. Next, it explains how the collected raw data is pre-processed by the Pre-processing component of the HORSE platform, which unifies and standardizes all collected data. Finally, the section describes the procedure for storing the pre-processed data in the Elasticsearch database, which is managed by the Smart Monitoring component. This database serves as a central repository, which feeds the AI-based HORSE components.

2.6.1 Data collection

2.6.1.1 UMU Testbed

UMU's testbed data collection is done through an API, where an administrator can schedule a capture of a particular 5G interface (N2, N3, N4 or N6) in two different ways:

- Capturing a given time on that interface, generating a .pcap file with that type of traffic. For example, a twelve hours capture on the N3 interface.
- Capturing by specifying an interval, in seconds, with a maximum number of files. For example, a capture with one-minute intervals, with a maximum number of files of 600.

Regarding the type of data, we always work with the .pcap format for compatibility and the type of traffic varies depending on the interfaces:

- The N2 interface captures control plane traffic transported over SCTP, such as NGAP/NAS-5GS.
- The N3 interface captures user plane traffic encapsulated over GTP prior to arrival at any UPF.
- The N4 interface captures control plane traffic with PFCP protocol, between the SMF and the UPFS.
- The N6 interface captures unencapsulated user plane traffic, after traversing the UPF to reach an infrastructure service or the Internet.

Regarding the traffic pattern, we have real user traffic, redirected to virtual UEs, which would simulate a normal traffic pattern. Moreover, it is also possible to capture at a certain time when an attack has been carried out with the UEs.

N2 Initial UE Message NGAP/NAS-5GS packet (see Figure 6):

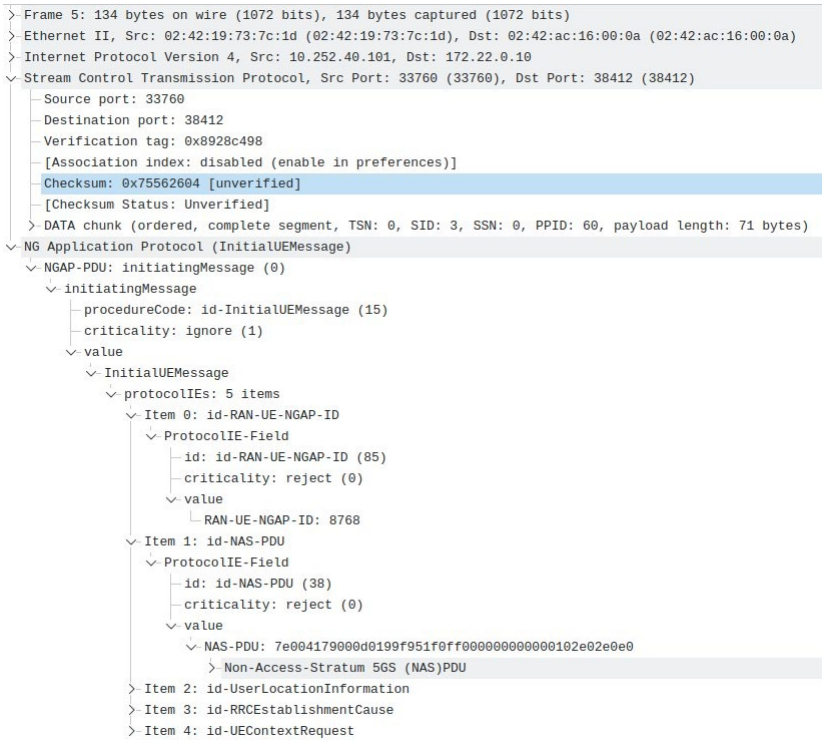


Figure 6. N2 Initial UE Message NGAP/NAS-5GS packet.

N3 GTP-encapsulated ICMP packet (see Figure 7):

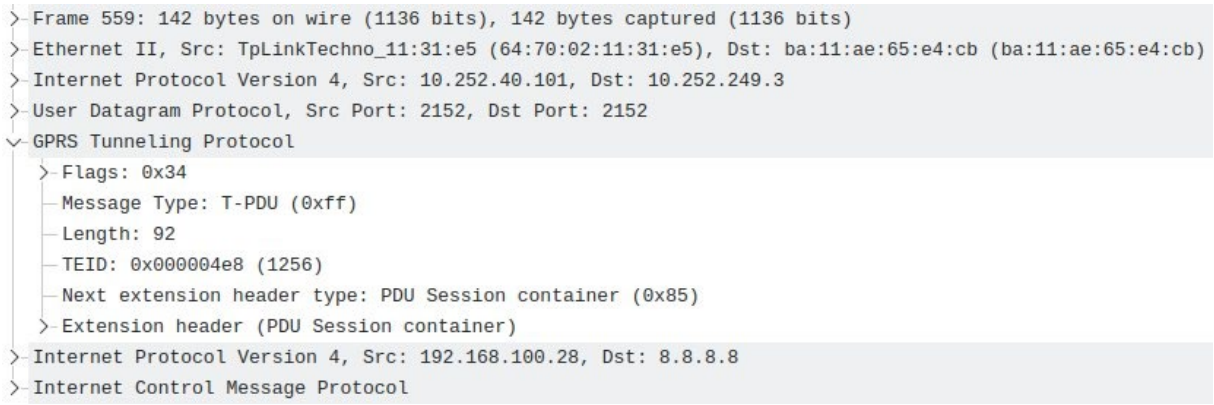


Figure 7. N3 GTP-encapsulated ICMP packet.

N4 PFCP Heartbeat packet (see Figure 8):

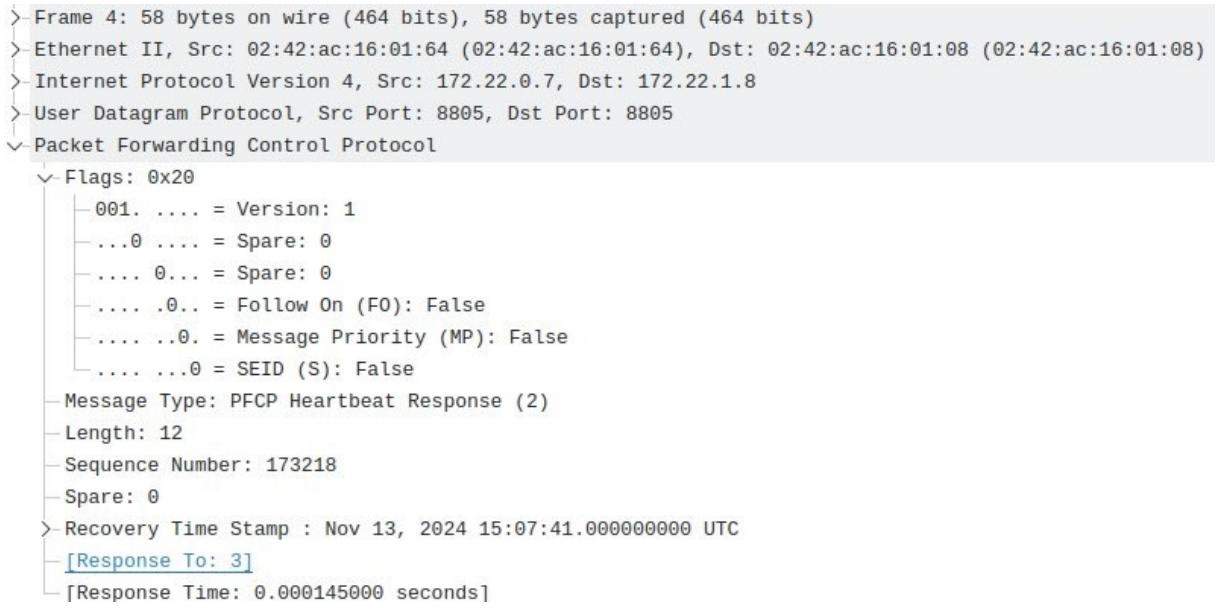


Figure 8. N4 PFCP Heartbeat packet.

N6 ICMP packet (see Figure 9):

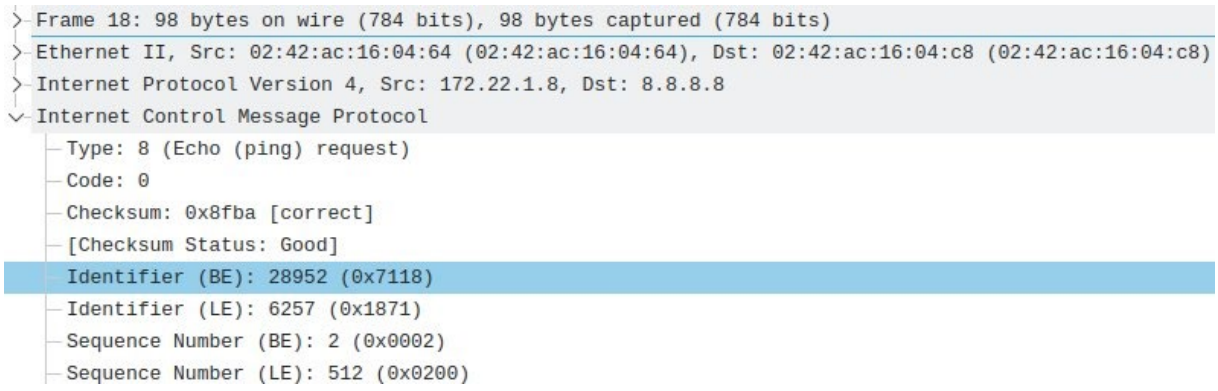


Figure 9. N6 ICMP packet.

2.6.1.2 CNIT Testbed

CNIT's testbed data collection is done through a python script where an administrator can schedule a capture of a particular 5G interface (N2, N3, N4 or N6) in the following way. The specific interval selected by the administrator defines when a new packet captured will be saved on the same host where the traffic is captured. When a PCAP file is saved automatically another packet capture starts. The script runs indefinitely. The parameters of this scripts are:

- host name (IP address) where to capture the traffic;
- 5G Interfaces (optional, default: all interfaces);
- output directory where to store the PCAP files;
- capture interval (in seconds, default: 60).

There is another script that is used to periodically remote the oldest captured pcap files. Also, this script runs indefinitely. The parameters of this scripts are:

- hostname (IP address) to monitor for oldest pcap files;
- PCAP files directory to monitor;
- maximum number of PCAP files that can be present in the PCAP file directory.

For what regards data, the traffic varies depending on the interface types:

1) N2 interface:

- Connects the gNB (gNodeB) to the 5GC (5G Core Network).
- Primarily handles control plane traffic, including signaling messages, configuration information, and user plane control information.

2) N3 interface:

- Connects the 5GC to the external network, such as the internet or other 5G networks.
- Handles both user plane and control plane traffic, including user data, signaling messages, and control plane data related to user sessions.

3) N4 interface:

- Connects different 5GC functions, such as the SMF (Session Management Function) and the UPF (User Plane Function).
- Primarily handles control plane traffic related to user session management and policy control.

4) N6 interface:

- Connects the 5GC to external networks, such as other 5G networks or legacy networks.
- Handles both user plane and control plane traffic, similar to the N3 interface.

2.6.1.3 UPC Testbed

In the UPC testbed, data is being captured on the following nodes highlighted in red in Figure 10:

- 10 UEs
- gNodeB
- UPF
- DNS Server
- Router Gateway
- Preprocessing Module
- ePEM Module

Monitoring of these nodes is carried out using tcpdump, a tool that captures all network traffic passing through the device interfaces, providing raw data in real-time. The tool generates data files every minute in PCAP format, with filenames following the pattern YYYY_MM_DD-HH_MM.pcap. These files are stored in the /packet-capture directory on each node and are automatically deleted after one minute.

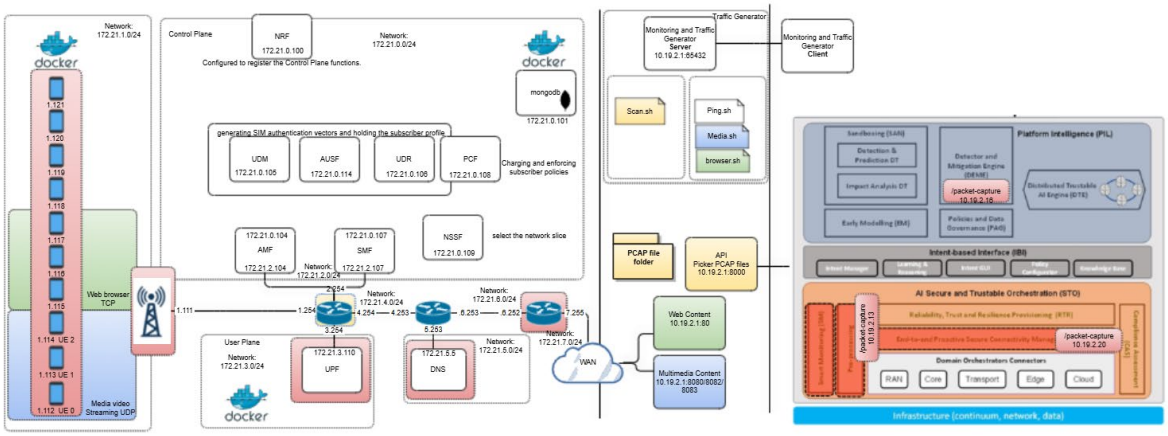


Figure 10. HORSE topology - UPC testbed

The types of data captured include:

1. Network Traffic Data:

- Source and Destination IP Addresses: Identify the origin and destination of the packets.
- Source and Destination Ports: Correspond to the specific applications or services in use.
- Network and Transport Protocols: Include protocols such as IPv4/IPv6, TCP, UDP, ICMP, and others.

2. Packet Metadata:

- Capture Times: Precise timestamps indicating when each packet was captured.
- Packet Sizes: The total length of each packet in bytes.

3. Packet Contents:

- Headers: Contain details from the network, transport, and link layers.
- Payload: The data transmitted by applications, which varies depending on the protocol in use.

4. Network Events and Errors:

- Includes data on retransmissions, fragmented packets, and anomalies in data transmission.

Future plans include implementing more specific filters for the data captured and to expand monitoring to other network nodes. The goal is to enable users to select which nodes to monitor, for how long, and the type of monitoring desired.

Currently, the nodes marked in yellow in the diagram are set up for monitoring for a specific duration. Once the monitoring period ends, the files are stored in a directory, and an API allows interaction with them.

2.6.2 Data pre-processing

The preprocessing of data is essential for transforming raw, heterogeneous input into structured and actionable formats suitable for further analysis within the system. This process begins with the periodic aggregation of data from the central Elasticsearch (ES) database, utilizing a sliding time window approach. The time window defines the interval of data being

processed, covering the period between the current query time ***t*** and ***t−tw,*** where ***tw*** is the predefined duration.

Within the specified time window, the preprocessing steps include:

- **Filtering:** Removing irrelevant or redundant data to ensure that only pertinent information is retained for analysis.
- **Structuring:** Organizing and formatting the data to match the specific input requirements of downstream workflows.
- **Aggregation:** Calculating metrics and summarizing data to provide high-level insights. For example, metrics such as the number of NTP or DNS packets observed within the time window are aggregated to support cybersecurity analysis.
- **Feature Extraction:** Deriving additional attributes from the raw data, which may involve identifying patterns or anomalies relevant to the use case.

The processed data is then outputted in formats optimized for subsequent analysis stages. It is transmitted via API to other system components and stored in the ES database for both real-time use and future reference. This ensures data consistency, accessibility, and readiness for integration into the larger analytical workflows.

2.6.3 Datasets Storage

In our data management system, plane traffic data, captured in .pcap format, is systematically stored in Elasticsearch for efficient querying and analysis. The .pcap files contain various types of network plane traffic, including NGAP/NAS-5GS over SCTP, GTP-encapsulated plane traffic before reaching a UPF, PFCP protocol exchanges between SMF and UPFs, and traffic traversing the UPF towards infrastructure services or the Internet. To ensure these files are transformed into a structured, searchable format, we employ a shell script that monitors the directory */packet_capture* for new .pcap files. Upon detection, the script uses tshark (the command-line interface for Wireshark) to convert the .pcap files into JSON format, preserving all relevant packet-level details.

Once converted, the JSON data is uploaded to Elasticsearch via its bulk API, ensuring high-performance ingestion into a dedicated index named *pcap_data*. This indexing allows us to organize and retrieve data efficiently, leveraging Elasticsearch's powerful search and aggregation capabilities. After the upload process, the script moves the processed .pcap file to an archival directory for long-term storage or further analysis. This workflow ensures that raw traffic data is seamlessly transformed, ingested, and managed, enabling robust visibility and insights into 6G network traffic dynamics.

3 Architectural Design

This section presents the final HORSE architectural design for IT-2. First, it details the principles applied during the review of the HORSE architecture defined in IT-1 [1]. Next, it introduces the final HORSE architectural design, emphasizing the key modifications done compared to the IT-1 architecture. The main modules of the architecture are described, including an overview of their internal components. Section 5 further explores the interactions between these modules, presenting two reference workflows that illustrate the functionality of the HORSE architecture.

3.1 New architecture design principles

In the second phase of the project, the HORSE architecture is being enhanced with new design principles to address new requirements for sustainability, flexible deployment and streamlined operations. These new design requirements focus on critical areas related to generative AI (GenAI), trust and user equipment (UE) mobility. Each of these design principles significantly improves the HORSE platform's functionality and robustness, enabling efficient management, intelligent system automation, and improved traceability throughout the entire process.

The use of GenAI through Large Language Models (LLMs) is transformative, enabling advanced automation, adaptive decision-making, and personalized interactions for complex processes. LLMs are powered by advanced natural language processing (NLP) [6] and machine learning (ML) techniques [7], offering a new frontier in the fight against cyber threats. LLMs can excel in various domains within cybersecurity to identify potential weaknesses and exploitable vulnerabilities, significantly accelerating the recovery process. The domains and applications of LLMs in cybersecurity include threat detection and analysis, security automation, phishing detection and response, cyber forensics, penetration testing, security protocols verification, incident response, security training, and awareness [8]. Furthermore, LLM models have the capabilities to classify and forecast malware variants, facilitating the implementation of proactive defense strategies. LLM models like GPT3 and GPT4 are also revolutionizing the cybersecurity domain by improving threat detection and incident response. GPT4 has demonstrated the capability to exploit one-day vulnerabilities, when provided with Common Vulnerabilities and Exposures (CVE) descriptions, showcasing its potential applications for both defensive and offensive strategies [9].

Trust [10] is considered as one of the main pillars of HORSE architecture, addressing the critical aspects of security, privacy and reliability. In HORSE, we are developing an AI-assisted, human-centric platform designed to ensure seamless device connectivity, optimize resource and data utilization, and strengthen security and trust capabilities for 6G-enabled smart devices. The HORSE architecture ensures that all the communications and operations are trustworthy, by incorporating the robust security mechanisms for end-to-end data encryption, as well as security policies to ensure that data access is only granted to the authorized entities. Moreover, the HORSE platform implements a mechanism to verify the integrity of information exchanged between the different HORSE modules. The integration of intent-based networking (IBN) with security architectures, such as zero-trust models, creates a more secure environment by continuously validating access and restricting lateral movement within the network, thereby enhancing overall security [11].

As mobility becomes a defining characteristic of modern systems [12], the HORSE platform emphasizes seamless support for dynamic user equipment. This includes adaptive resource allocation, low-latency handover, and consistent service quality across different cells. By prioritizing user equipment mobility, the architecture ensures reliable connectivity, enabling users to monitor the status and summary of all AI pipelines.

3.2 Reference architecture in IT-2

This section presents an overview of the HORSE architectural design conducted in IT-2 (see Figure 11). The main changes regarding to the IT-1 version 2 HORSE architecture, described in Section 2.1, can be summarized as:

- The SM component has a new element, so-called UEs tracking, to monitoring the connectivity of the connected devices and to determine the location of the UEs within the network.
- The PAG component provides access control to the data collected by the SM module stored once pre-processed in the Elasticsearch. To enable this functionality a connection has been added between the PAG and the SM components.
- The IBI component tracks the status of the attacks, mitigations, preventive actions, as well as the status of the 6G network. To this end, the Dashboard element has been added in the IBI component. In addition, an asynchronous flow has been defined between the ePEM, RTR, and IBI, to maintain the IBI continuously informed about the status of the execution of the mitigations and preventive actions.
- The Dashboard element will provide an interactive view of the status of the 5G/6G network and security-related events, including the detected and predicted attacks, the mitigations and preventive actions enforced, the impact analysis, and the recommendations. More specifically, this component integrates the outputs (dashboards) from the different HORSE components.
- The DTE supports decentralized training of ML models, and it enables the dynamic update of the ML models.
- The CKB component is integrated in the HORSE framework to manage the mitigation and preventive strategies. This component interacts with the IBI and EM components.
- The CAS component ensures that the enforcement policies defined by the ePEM are aligned with the HORSE policies defined by the IBI and with the regulatory frameworks, such as those established by 3GPP and ENISA.

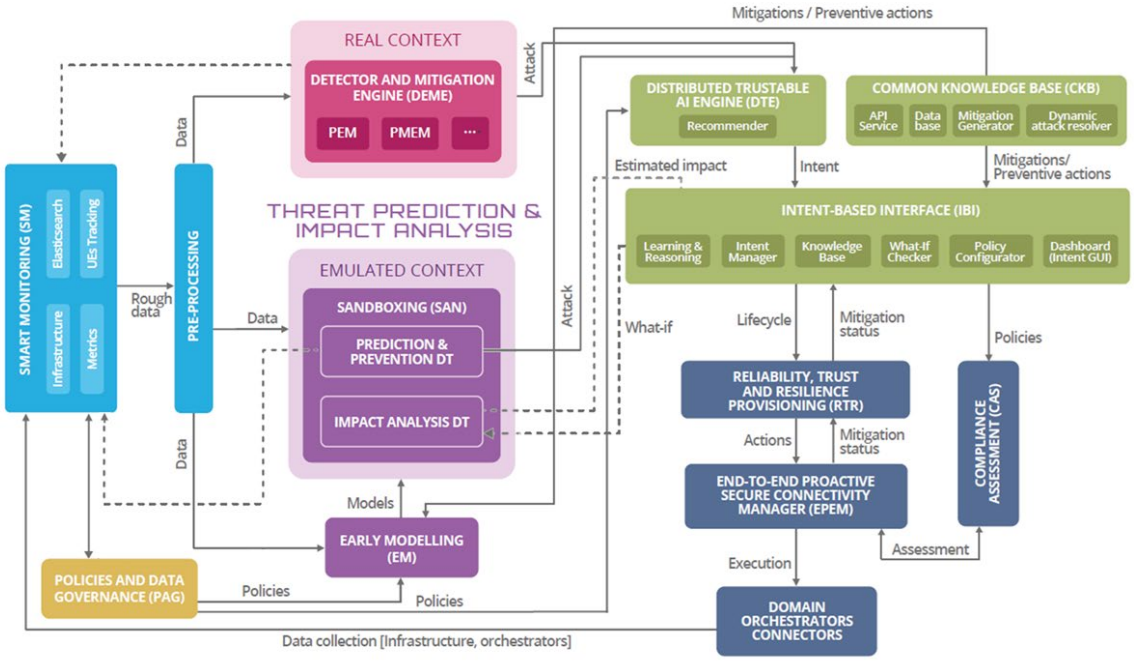


Figure 11: HORSE final architecture (IT-2)

Next, we provide an overview of the HORSE main modules, shown in the architectural design for IT-2 (see Figure 12).

The STO module is responsible for providing security and reliability in the HORSE architecture. It consists of the following seven components. Note that the Common Knowledge Base component has been added to the HORSE IT-1 version 1 architecture to provide essential information on attacks, mitigations and preventive actions to the HORSE components.

- The Smart Monitoring (SM) component collects data from the infrastructure, domain orchestrators, and resource usage information related to the lifecycle management of 6G services. In addition, it tracks the status and location of the connected UEs.
- The Pre-processing component unifies and standardizes all collected data. It will orchestrate and manage large-scale, structurally diverse data sources within a common and expandable data framework.
- The Common Knowledge Base (CKB) stores and provides essential information on, attacks, mitigations and preventive actions. It benefits from advanced generative AI techniques to automatically generate new mitigation and preventive strategies, which will enhance the database. It also prioritizes the mitigation and preventive strategies through ranking mechanisms.
- The Reliability, Trust and Resilience (RTR), provides the set of tools and technologies to ensure a secure performance. It defines the mitigation and preventive actions to be enforced by the ePEM component, in terms of Ansible security playbooks, according to the workflow defined by the IBI.
- The end-to-end (E2E) secure connectivity manager (ePEM) works as an Operations Support System (OSS) executing the mitigations and preventive strategies stablished by the RTR over the available infrastructure. Additionally, it manages and maintains information on deployed applications, network services, and available resources.
- The Domain Orchestrator Connectors integrates management and orchestration functionalities across all network segments, including RAN, transport, core, near edge, far edge and cloud.

- The Compliance Assessment (CAS) component ensures that the enforcement policies defined by the ePEM are aligned with the HORSE policies and fully aligned with the applicable regulatory framework, focusing on 3GPP and ENISA.

The IBI module aligns the received high-level intents with the configured policies and translates them into workflows using appropriate ML techniques. Prior to generating the workflow, the policy configurator validates the intent requirements against the applicable policies ensuring alignment and consistency. Simultaneously, the translator collaborates with the Impact Analysis NDT to assess whether the estimated impact of the mitigation or preventive strategies defined in the workflow is acceptable. In addition, the IBI shows in the Dashboard, the status of the network; the risks, in terms of detected or predicted attacks or anomalies; and the status of the enforced mitigation and preventive actions.

The PIL module comprises advanced strategies and intelligent mechanisms to support HORSE's predictive approach and serves as the interface for domain orchestration. It is composed of the following five components.

- The Sandboxing (SAN) environment operates within an "emulated context," enabling the simulation of multiple realistic scenarios using a "network-in-network" approach. It provides a secure, controlled, and realistic environment for emulating and experimenting with various services, alternative connectivity topologies, traffic paths, and the deployment of specific security network functions across different networks. The environment includes the following two NDT components:
  - Detection & Prediction NDT predicts anomalies and threats in the emulated environment.
  - Impact Analysis NDT estimates the impact of enforcing mitigation actions and preventive strategies in the emulated environment previous being enforced in the 6G infrastructure.
- The Early Modeling (EM) component provides all the information required by the SAN to successfully perform. It models potential threats and attacks in the 6G infrastructure, their impact, and the impact of enforcing the mitigation and preventive actions on 6G components.
- Detector and Mitigation Engine (DEME) works in the "real context" providing threat detection in the real infrastructure. It focusses on threat detection and high-level mitigation advise with a special attention to the most dangerous attack cases, able to impact, and often paralyze, whole portions of the network for a long amount of time. In the IT-1 architectural design, the DEME provides as output a high-level advice according to the threats detected. It should be noted that the IT-2 architectural design may also consider threats that might potentially require immediate mitigation actions to be applied. In this case, the DEME would be the responsible for defining the appropriate mitigation strategies, which will be enforced by the ePEM over the infrastructure.
- Policies and Data Governance (PAG) integrates the tools and services needed to define and enforce data policies for all types of data stored and handled by the HORSE platform in the Elasticsearch. This includes access controls, privacy preservation rules, encryption rules and data retention policies.
- Distributed Trustable AI Engine (DTE) component gathers the outcomes from both real and emulated context and generates a high-level description of the mitigation and preventive strategies to be enforced in the different 6G components, expressed in the form of intents.

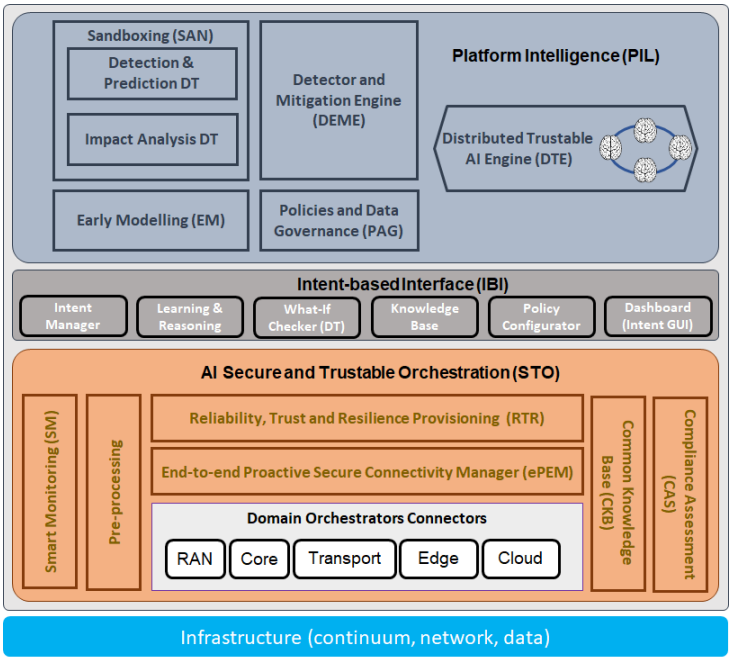


Figure 12: HORSE final architecture (IT-2) – Main components

3.2.1 Data collection

3.2.1.1 Smart Monitoring

This module will be responsible for the collection of data from the various and diverse sources of the HORSE infrastructure. Data will be collected from all VNFs in order to provide feedback to the Distributed Trustable AI engine. Figure 13 shows the logical position of the Smart Monitoring module within the HORSE architecture.

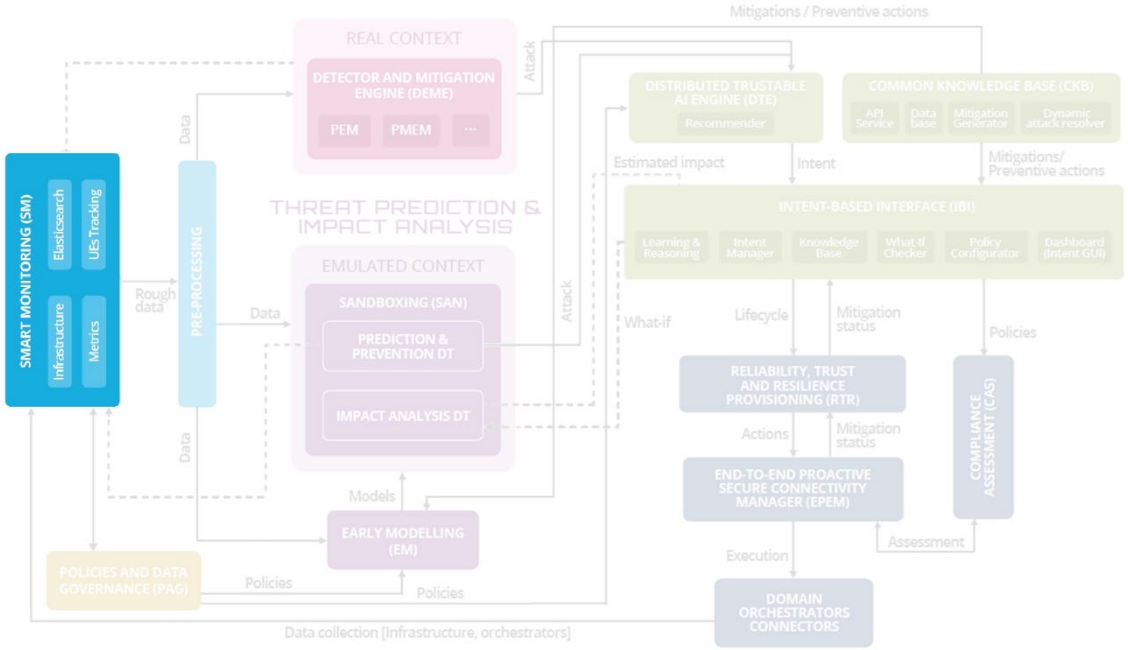


Figure 13: The logical position of the Smart Monitoring module within the HORSE architecture

In the intricate landscape of 6G networks, the functional capabilities of smart monitoring take on paramount significance. The 6G paradigm, being based on highly heterogeneous infrastructure, as already stated, introduces a multitude of complexities, ranging from diverse network domains to the virtualization of resources as well as varying hardware abstraction levels. The multi-domain nature of 6G necessitates a unified approach to the management of networks and observability across administrative boundaries. Smart monitoring in 6G extends beyond traditional monitoring for cellular networks, reaching into the realm of advanced analytics, machine learning, while heavily utilizing real-time processing of generated control-plane data. This holistic observability encompasses the aggregation and analysis of performance metrics, security incidents, and information regarding resource utilization. It is thus challenging to account for all security functionalities and their corresponding functional blocks' interfacing. However, it's evident that the concepts of network disaggregation, virtualization, and cloud-native principles will remain central. In this context, SPHYNX's Event Reasoning Toolkit (EVEREST) emerges as a solid choice for the HORSE 6G case. Log analytics and its adaptabilities align perfectly with the dynamic and agile requirements of 6G networks. HORSE's intentions of extending and adapting EVEREST to suit the specifics of 6G telecommunications, position it as an ideal candidate. By integrating EVEREST, HORSE can not only monitor but also intelligently manage the intricate 6G network environment, optimizing resource allocation, ensuring robust security, and meeting the rigorous demands of the relevant use case. This underscores EVEREST's pivotal role in shaping the future of 6G-ready network architectures, offering a powerful solution to the challenges of observability, security, and resource management in the next generation of telecommunications networks.

An Event Captor is a tool that aggregates log and event information from the targeted infrastructure and encapsulates it in a specific format that can be consumed by the HORSE analytics. Logs and events can be collected in two modes. The former mode is based on the ELK solution. More specifically, Elasticsearch [13] and some lightweight shippers (namely Beat [14]) are utilised to forward and centralize log data. The latter makes use of SPHYNX's Native Event Captors, i.e., captors that cannot utilise the logging capabilities of the ELK stack

HORSE will support three testbeds and each one will consume data from nine nodes. Event captors will capture network data from the nodes and store them to the Elasticsearch installed in each testbed.

In HORSE, EVEREST will assume the role of the Smart Monitoring Module which will be responsible for: i) retrieving data and security logs from running services and software packages, physical servers and SDN controllers running on different administrative domains, ii) enabling flexible management and processing of the collected data in a homogeneous manner, and iii) permanently storing data in a metrics database which is accessible by analytics tools to perform intelligent resource management and orchestration. To realize this goal, the monitoring component will rely on a high-performance, distributed, and scalable message queue that would allow exchange of monitoring information between publishers (running services) and subscribers (analytics tools that consume monitoring metrics). The monitoring architecture of the monitoring component is depicted in Figure 14.

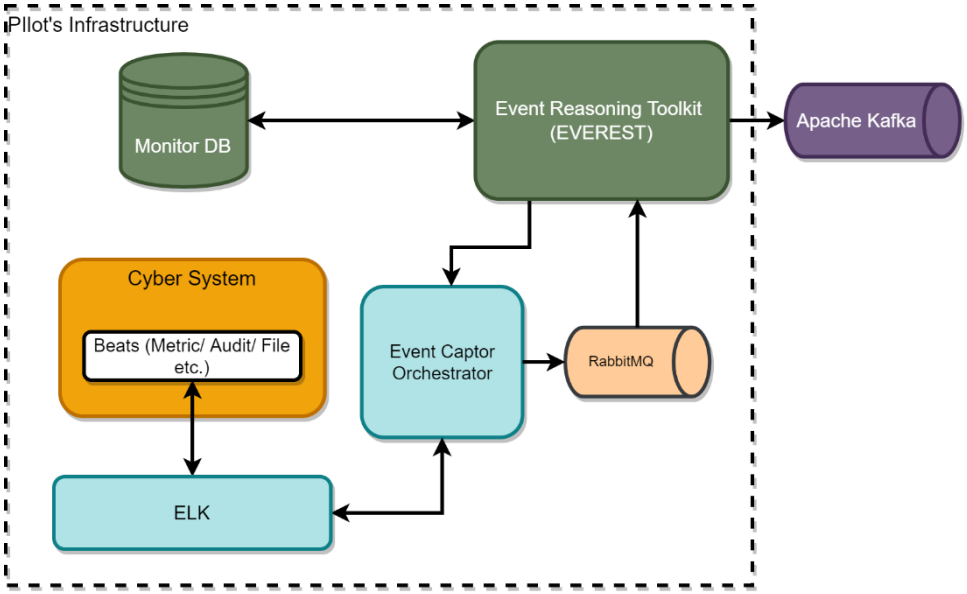


Figure 14: Internal architecture of the monitoring component

3.2.1.2 Pre-processing

The Pre-processing module, a pivotal addition to the HORSE architecture, assumes the role of harmonizing and standardizing the data accumulated by the Smart Monitoring (SM) component. Designed to bolster the efficacy of the system, this module serves as a bridge between data collection and subsequent analysis. By unifying data from diverse sources - ranging from infrastructure components to domain orchestrators - the Pre-processing component contributes to the creation of a unified and coherent data landscape.

As shown in Figure 15, Pre-Processing component is responsible for frequently aggregating data (based on a stable time interval) from an Elastic Search (ES) database which is based in the center of SM component and subsequently process those initial raw data in order to transform them into a form that is usable by components like DEME and EM. The processing of the data consists of multiple data pipelines that input data form the SM component within a time window (tw). This tw covers the time period from the moment of querying the DB (t) until the moment of querying the DB minus the time window (t – tw).

Within this time frame, the pipelines filter out irrelevant elements and structure the remaining data to align with the input requirements of downstream components. For example, one pipeline might construct data suitable for the DEME component, aggregating metrics like the number of NTP and DNS packets observed in the last two minutes that meet specific criteria for detecting potential cybersecurity threats.

The Pre-Processing module forwards its results, tailored to the input needs of subsequent HORSE workflow components, via API POST requests. Additionally, these results are stored permanently in the SM component's Elasticsearch database, ensuring they are available for future analysis.

Figure 15 below shows the logical position of the Pre-processing module within the HORSE architecture.

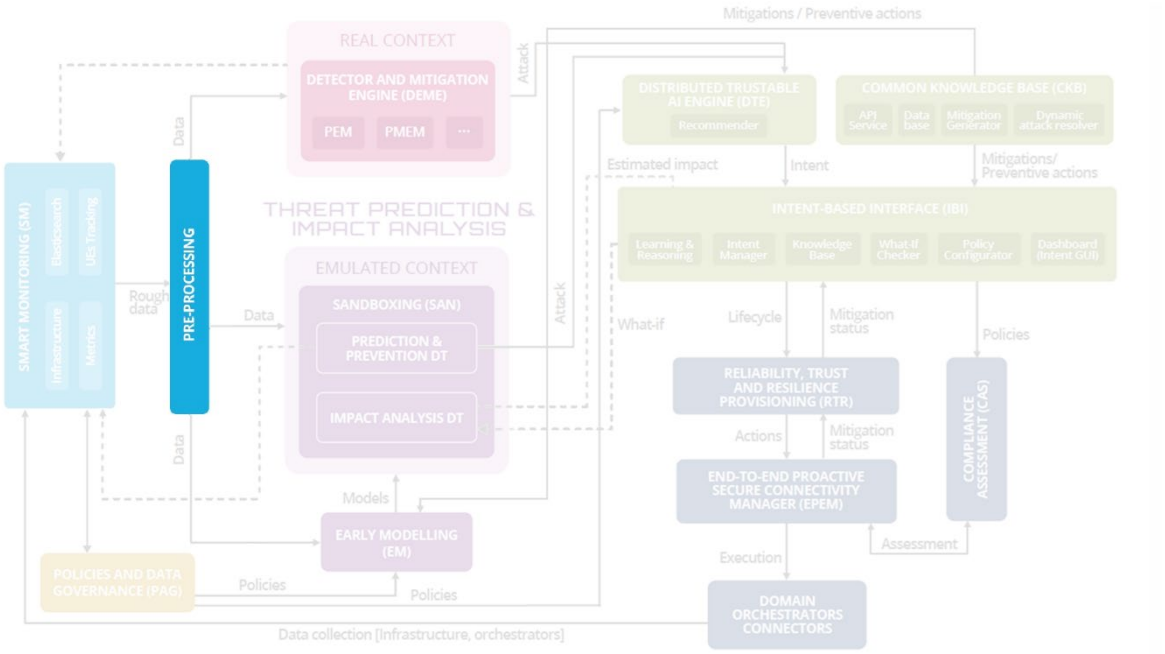


Figure 15: The logical position of the Pre-processing module within the HORSE architecture

One of its core functions lies in orchestrating the aggregation of data from various ES indexes that contain raw and heterogenous data, into cohesive, manageable data spaces. This orchestration is implemented by creating and sending adaptable API queries to indexes of the ES and waiting for a response in an asynchronous manner. Once the ES replies to a request, the data are processed by a workflow tailored to the needs of a connected component. This procedure fosters the integration of structurally varied datasets, ensuring that the subsequent analysis benefits from consistent and comprehensible data structures.

Through its capacity to homogenize data, the Pre-processing module effectively contributes to the optimization of subsequent analysis processes. By standardizing data and providing a consolidated foundation, this module prepares the collected information for further evaluation and utilization within the HORSE architecture. In essence, the Pre-processing module's role is not only in data harmonization but also in enabling efficient, accurate, and unified analysis across the entire spectrum of the HORSE platform.

3.2.2 Platform Intelligence

The Platform Intelligence (PIL) module comprises five key components:

- Detector and Mitigation Engine (DEME)
- Sandboxing (SAN)
- Early Modelling (EM)
- Policies and Data Governance (PAG)
- Distributed Trustable AI Engine (DTE)

This module is responsible for integrating various methodologies, procedures, and tools, allowing systems and machines to function with a superior intelligence.

3.2.2.1 Detector and Mitigation Engine

Figure 16 shows the logical position of the Threat Detector and Mitigation Engine module within the HORSE architecture.

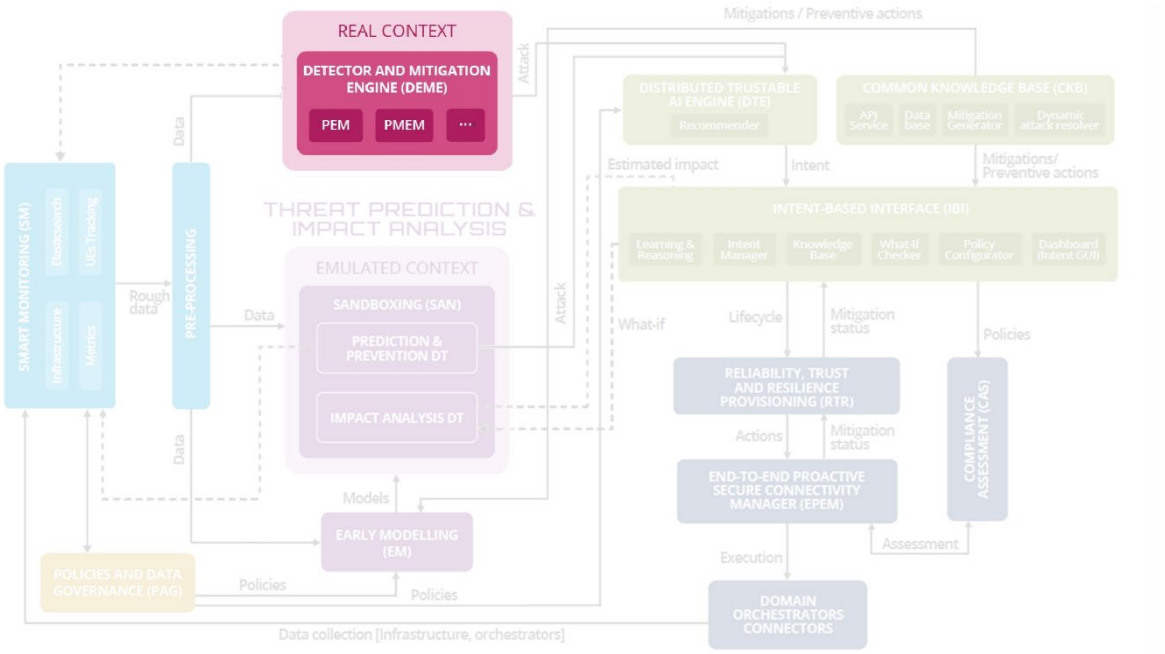


Figure 16: The logical position of the Threat Detector and Mitigation Engine module within the HORSE architecture

3.2.2.1.1 DEME final version

Compared to the first iteration of the threat detection circuit, which was still rough, the solution was completed and refined in the second iteration, resulting in the final version. Here, we describe the main architectural features of this version.

In Figure 17 we see the innovative Multistage Pipeline.

The first stage collects all the data gathered in real-time by the smart monitoring blocks, as shown, and applies predictive machine learning to develop corresponding predictive models. These models are initially derived from historical data analysis and are continuously updated in real-time with each new sample received. The expected values, or forecasts generated by the models, are then compared with the actual values received to calculate the observed differences. These differences are represented in the figure using Delta symbols (e.g., DeltaA, DeltaB, up to DeltaM).

In the second stage, these difference values are processed mathematically, allowing for normalization, data adaptation, and other applications. Finally, the egress stage (which, in this three-stage implementation, may evolve in the future with the addition of further stages) enables the simultaneous processing of all deviations of every parameter within the observation space. This enhances overall visibility, facilitating the learning and recognition of new patterns, emerging attack forms, zero-day attacks, and so on. These findings are reported, indicating the type of recognized or most similar attack and the confidence level of the detection. The complete architecture is presented in Figure 18.

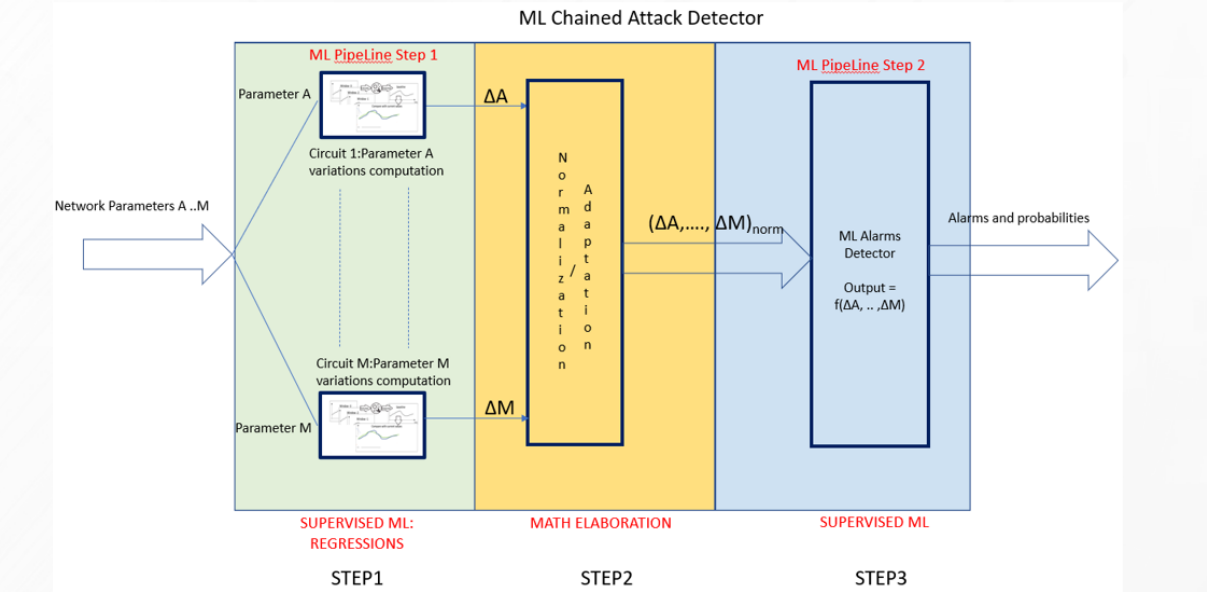


Figure 17: Multistage pipeline block diagram.

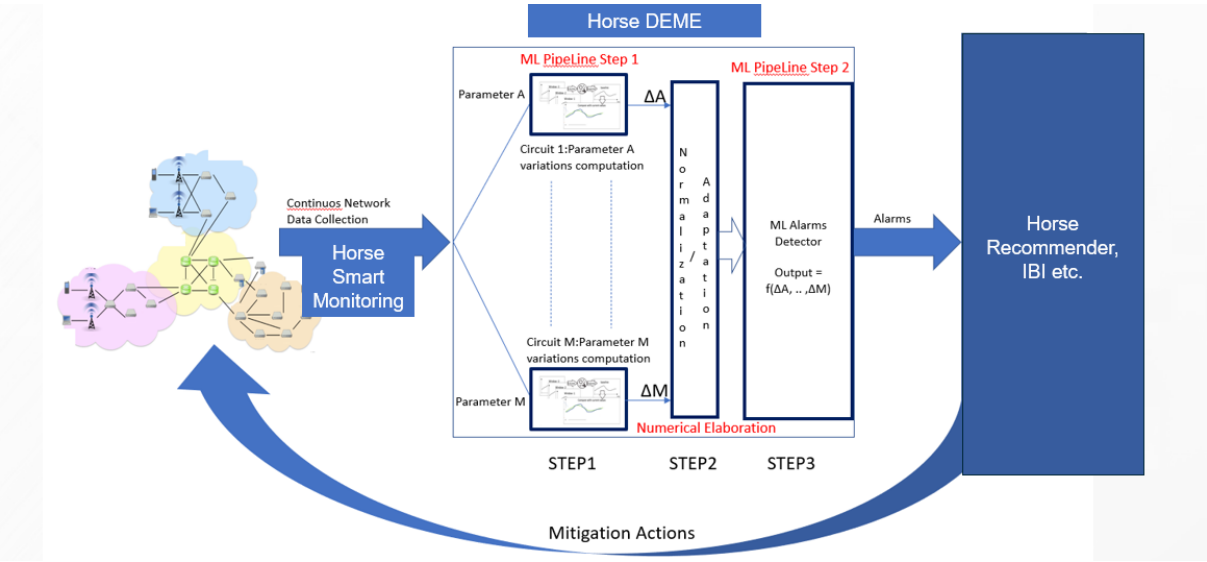


Figure 18: DEME integrated in the HORSE framework

In the following Figure 19, a detailed breakdown of the processing phases carried out in the first stage is presented. This stage has been refined to integrate with the Smart Monitoring and Preprocessing modules to receive the necessary inputs via pkfiles for the initial learning phase, development of the predictive models, comparison of real-time received data with expected values, data parsing, and the updating of the predictive model. This updating occurs purposefully, meaning it is based on specific conditions aimed at maintaining the consistency of the model itself.

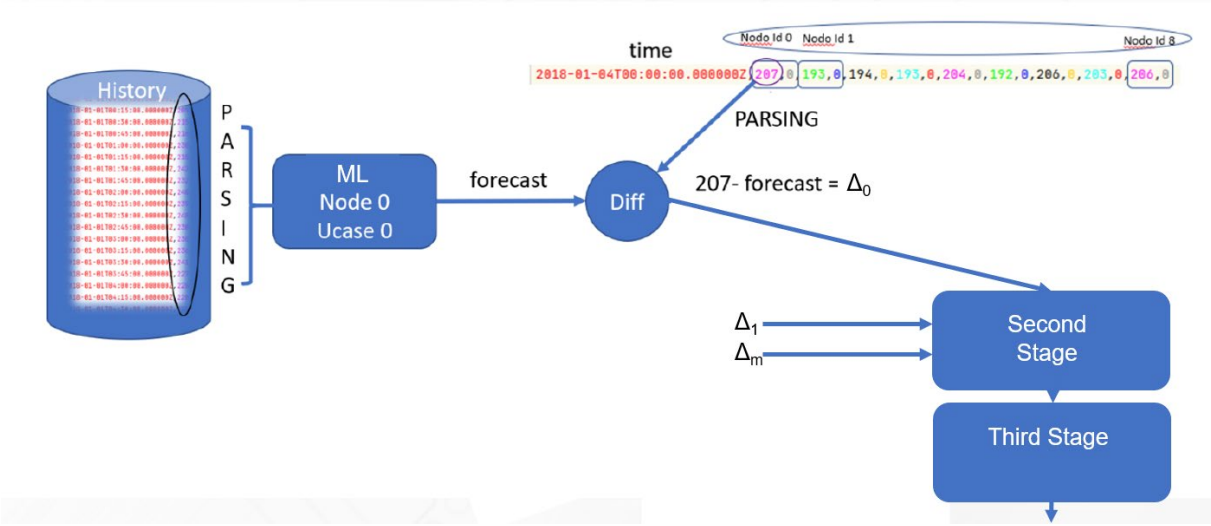


Figure 19: First stage processing

3.2.2.2 Sandboxing

The Sandboxing (SAN) module is part of the Platform Intelligence (PIL) module and is designed to facilitate the emulation of multiple realistic situations in a "network in network" environment. This approach will be achieved by applying the concepts of Network Digital Twin [15], by emulating diverse network scenarios.

By using Network Digital Twins, different network configurations and changes can be tested and validated before being deployed to the real environment, reducing the risk of network outages or failures and also, the cost is reduced. The data generated by the Digital Twins will be accessible to the rest of security components for intelligent analyses and predictions.

Focusing on the sub-modules of the Sandboxing, we have two different approaches:

- The Prediction and Prevention NDT will predict anomalies in the emulated environment and will be able to propose mitigation actions.
- The Impact Analysis NDT will be responsible for the "what-if" question, this is a pure experimentation environment to test the behavior of the different modules deployed on it.

3.2.2.2.1 Prediction and Prevention DT

The Prediction and Prevention Network Digital Twin will be a tightly coupled Digital Twin of the Network, responsible for supporting prediction and prevention of relevant network events (e.g., presence of new flows, or expected congestion raise) and security treats (e.g., DDoS attacks). Figure 20 shows the logical position of the Prediction and Prevention DT module within the HORSE architecture.

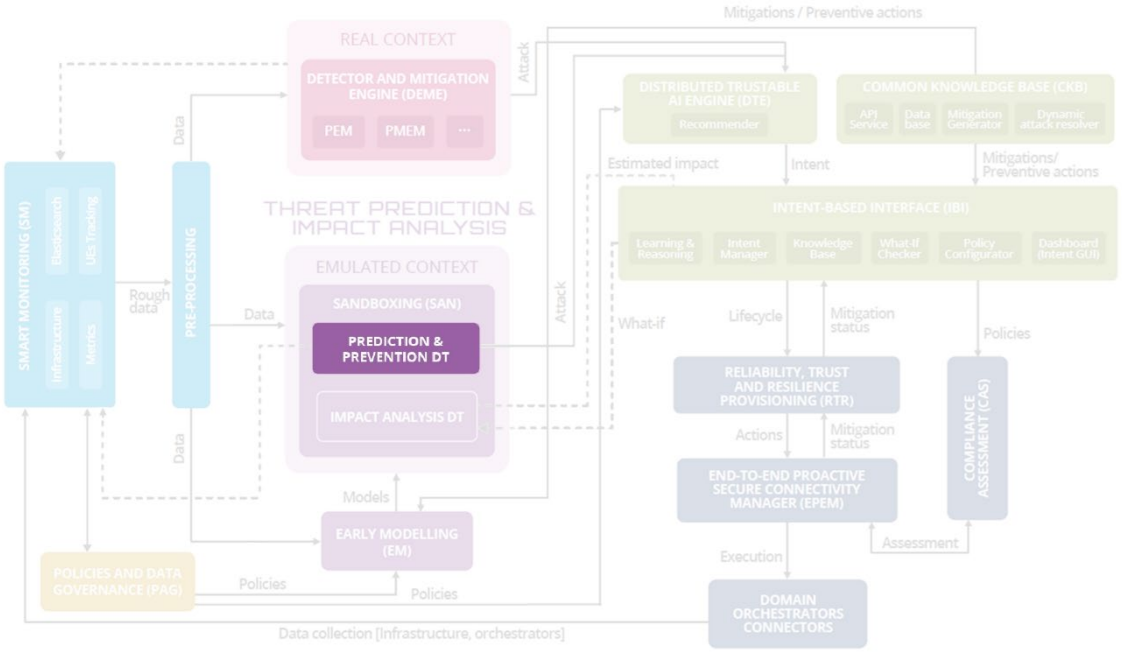


Figure 20: The logical position of the Prediction and Prevention DT module within the HORSE architecture

The Prediction and Prevention Digital Twin offers two services:

1. Construction of the Digital Twin of the 6G network: this service will enable the generation of the Digital Twin of the Network in an emulation environment. The module will offer two alternative functionalities: (i) automated runtime construction of the NDT (by exploiting interfaces with the Smart Monitoring / Pre-processing modules, the system will automatically detect network topology, traffic flows and traffic matrix and running services); and (ii) offline construction of the NDT (the network setup, traffic and services will be pre-defined through configuration scripts). As described during the previous iteration, the Prediction & Prevention Network Digital Twin will be built based on the Comnetsemu network emulator, an SDN/NFV-powered emulation environment capable of running a complete 5G network in a single laptop [16] [17].
2. Execution of the Network Digital Twin and generation of predictions and warnings: the Prediction and Prevention NDT will be executed while maintaining tight coupling with the actual infrastructure and services in order to provide predictions to support DTE or IBI decisions. In the second phase of HORSE development, the NDT will be upgraded in order to continuously analyze incoming data about the status of the network to identify anomalies and inform DTE accordingly (e.g., detecting or even predicting a security treat in order to automatically trigger mitigation actions). Exchange of data between the Physical and the Digital Twin will be achieved periodically, and supported by interaction with the Smart Monitoring module. Multiple instances of the Prediction and Prevention NDT might be allocated and executed to support different scenarios and study different strategies, if required.
3. Data collection and analysis: the Network Digital Twin will store data about traffic flows and state of the emulated network. Prometheus and Grafana softwares will be used to store and visualize data for further analysis by the HORSE network manager. Data collected by the Digital Twin will include: pcap traces in strategic links, predicted traffic patterns, docker containers status information (CPU utilization, memory and network resources) and log files.

The block structure of the Prediction and Prevention Network Digital Twin is the following:

- Digital Twin Modeling block: it is responsible for generating the DT based on the input data (traffic and topology information, orchestrated services, etc.)
- Digital Twin Engine block: it will run the DT in the Comnetsemu emulation environment
- Digital Twin-based Prediction block: it will analyze the output of the DT Engine block using AI/ML algorithms to perform predictions and identify anomalies
- I/O Interface block: interface with DTE / IBI for receiving requests and providing the related outcomes

Figure 21 shows the conceptual diagram on the Prediction and Prevention NDT module.

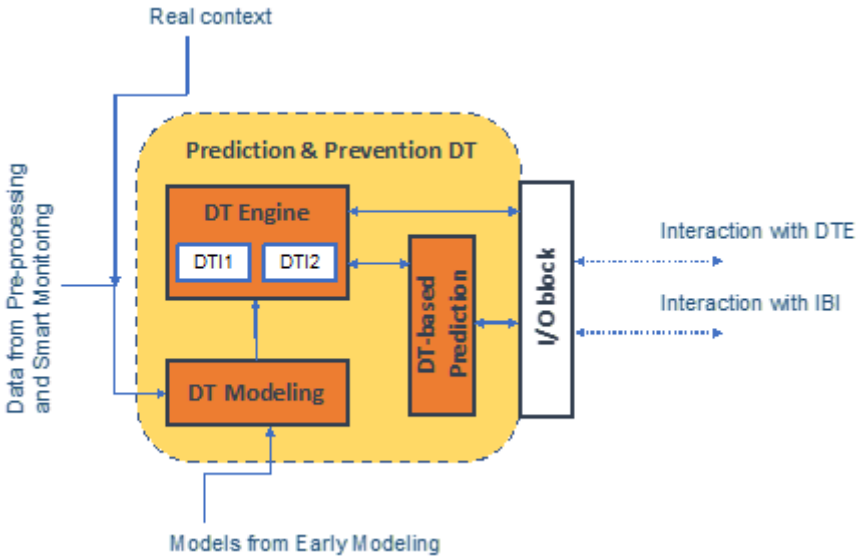


Figure 21: The block structure of the Prediction and Prevention DT module

The module will offer RestAPI interfaces with respect to the other modules in the HORSE system, and it will support existing standards in the field of topology representation, security attacks description and Digital Twin architecture, including RFC 8345 - A YANG Data Model for Network Topologies, STIX - Structured Threat Information Expression and IRTF Digital Twin Network: Concepts and Reference Architecture [15].

3.2.2.2.2 Impact Analysis DT

Figure 22 shows the logical position of the Impact Analysis NDT module within the HORSE architecture.

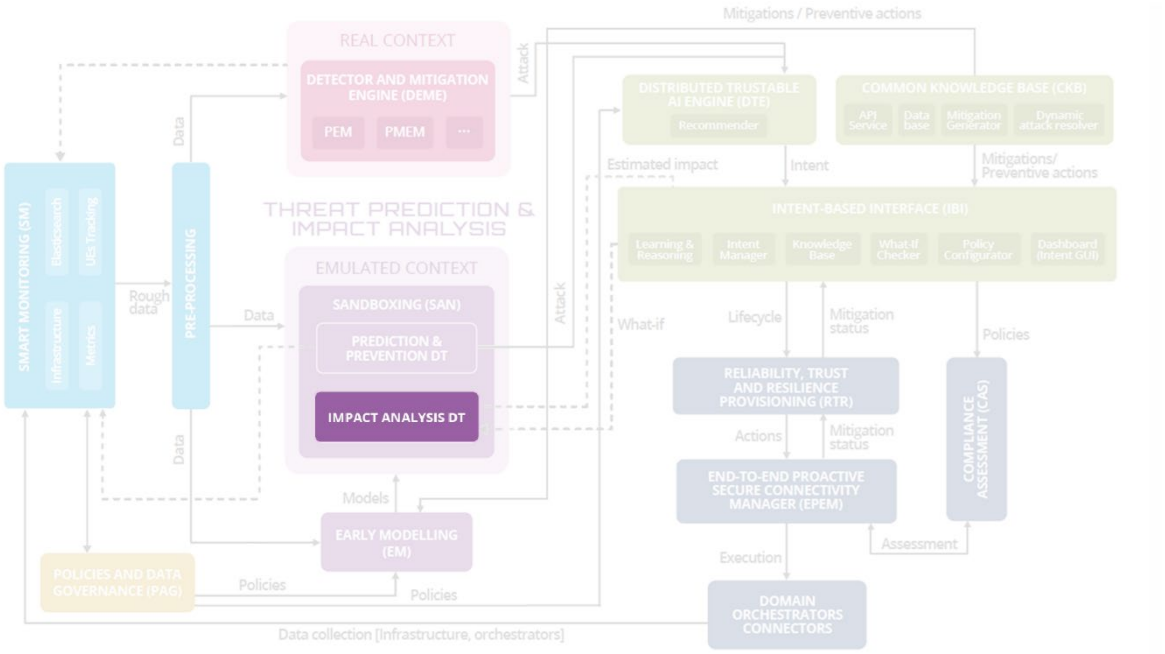


Figure 22: The logical position of the Impact Analysis DT module within the HORSE architecture

The Impact Analysis Network Digital Twin, as part of the Sandboxing, is responsible for providing reports to the Intent-Based Interface regarding experiments about attacks and mitigations of the network. To make that, this module emulates the network in a Kubernetes environment, ready to execute the concrete attacks and mitigations requested.

The idea behind this module follows the what-if loop, this concept is enclosed in a scenario where the goal is to mitigate attacks on the network. This flow starts in the IBI, which sends a request about any countermeasure that wants to apply to a certain attack, in a concrete point of the network and with a specific KPI. For example: "WHAT is the latency in eth1 of DNS client IF we apply a rate limit of 20 packets per seconds in the DNS server."

With these definitions, the Impact Analysis NDT can work by testing the concrete scenarios proposed by the IBI and then, the NDT will answer those questions by sending the KPIs that were specified on the what-if requests.

Finally, the Intent-Based Interface, with the information of the different impacts that has been emulated in the Impact Analysis NDT, can take the necessary decisions to be later on applied into the real infrastructure.

In this flow also appears the Model Translator, which is an intermediate module in charge of translating the requests sent by the IBI and also the response with the feedback to the IBI. So, the general flow of this what-if loop is the one showed in Figure 23.

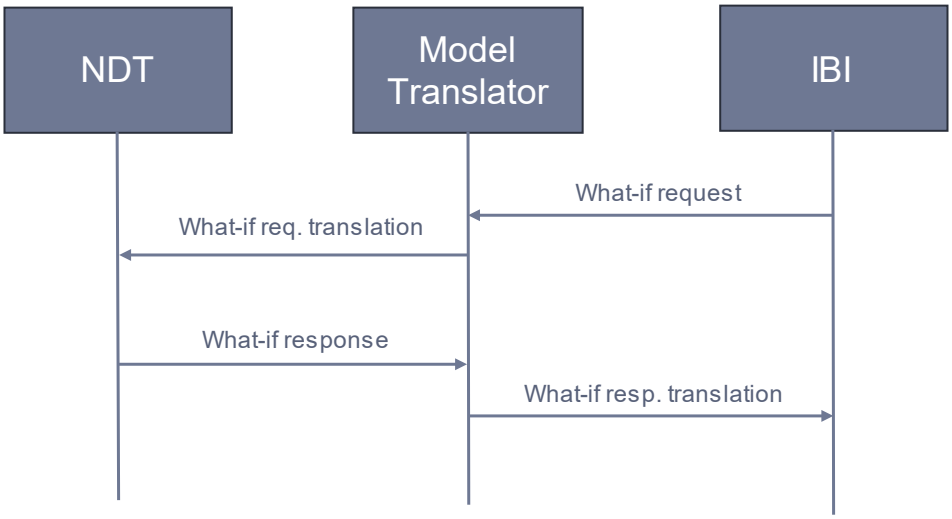


Figure 23. General flow of the what-if loop.

3.2.2.3 Early Modelling

Early modeling (EM) component is responsible for providing all information required by the Sandboxing (SAN) to successfully perform, including threat and impact models.

Figure 24 shows the logical position of the Early Modeling module within the final HORSE architecture (IT-2).

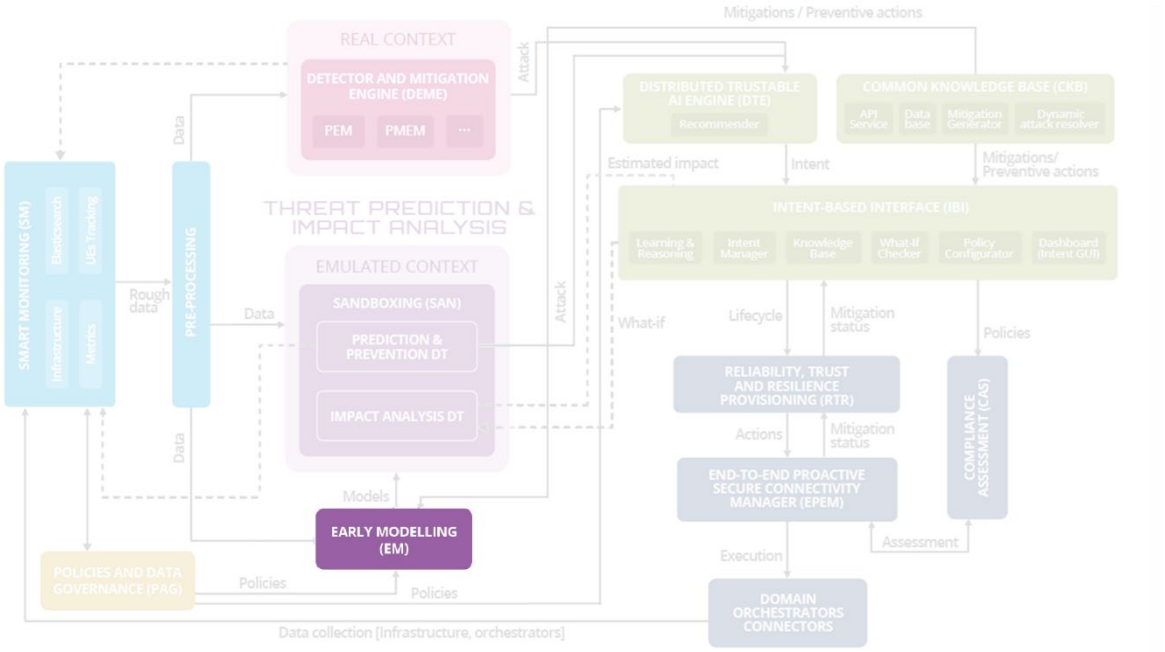


Figure 24: The logical position of the Early Modeling module within the HORSE architecture

The IT-1 version of the Early Modeling module focused on modeling attacks by incorporating the information about the attacks and their corresponding preventive and mitigation actions [18], which were extracted from the Common Knowledge Base (CKB). The output of this modeling process was provided to the SAN for further assessment.

In the initial phase, the modeling information was static. However, in this phase, we aim to provide dynamic modeling information, which will not only enhance the model but also validate the generated outputs. To achieve this, we will use a reinforcement learning (RL)-based approach [19], allowing the system to progressively acquire knowledge over time. This will enable the model to continuously update itself while simultaneously verifying the accuracy of the generated model.

Additionally, in this second period of the project, we will focus on assessing the impact of attacks within the SAN environment. This phase is critical, as it seeks to understand how such attacks could compromise the performance metrics related to confidentiality, integrity, and availability in a SAN environment. The goal is to develop a methodology that incorporates information about the attacks, mitigation strategies, and the overall impact on critical 6G and B5G infrastructure.

Cyberattacks and their effects on system performance and operational efficiency are inherently uncertain and unpredictable. These attacks exhibit considerable variability in both their occurrence and impact. In the current state of the art, this inherent uncertainty is represented by treating the state of the system as a stochastic entity [20], effectively represented as a random variable. This random variable can be modeled using a finite state machine, which defines the various possible states in the system's operational environment. The transitions between these states, which reflect the dynamics of the system under various conditions, can be simulated using a Markov chain model.

During an attack scenario in the SAN environment, the system's performance can be categorized into different states, such as fully functional, partially functional, barely functional, and non-functional. This categorization helps in understanding the system's resilience and its ability to adapt to changing conditions. The different states are defined as follows:

- Fully functional: System performance is unaffected.
- Partially functional: System performance is reduced by 50%.
- Barely functional: System performance is reduced by 75%.
- Non-functional: System performance is completely disrupted.

To evaluate the impact of the attack, we need to identify attack-specific evaluation metrics. For example, NIST performance metrics [21] can be used as a baseline to assess the impact of the attack. Additionally, domain-specific metrics such as throughput and latency can be employed to measure the attack’s effects on the architectural level. We will use metrics tailored to the nature of the attack to assess its impact. Figure 25 illustrates how modeling information can be incorporated into the XML schema, including details about the attack, mitigation actions, and the overall impact.

<?xml	version="1.0" encoding="UTF-8"				
ThreatModel	@id	tm_8b2c65n3dr87345s54gd746b7h83t82k			
	@xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance			
	@xsi:noNames...	file:///C:/Users/Saman%20Tariq/Downloads/threatModel%20(3).xsd			
	ThreatMod...	@id	tme_6d3c88w5rt45873g98el723f7g92j631		
		CyberAttack	Type	Network Denial of Service	
			Pattern		
			Vector		
			ATT_CK	Type	Network Denial of Service: Reflection Amplification
				ID	T1498.002
			Estimated_impact	System_parameter	Server-side processing time
				System_components	Server
				System_services	End user Application
		ControlAc...	Mitigation	MitigationAction	Type FilterNetworkTraffic
					ATT_CKID M1037
				MitigationCondition	@type FilterNetworkTraffiCondition
					FilterCondition
					isCNF false

Figure 25: Threat model integrating the estimated impact

3.2.2.4 Policies and Data Governance

According to the architecture, Policies and Data Governance (PAG) will integrate all required functionalities for establishing and applying data policies concerning all types of data stored and handled by the HORSE platform in the Elasticsearch. This includes access controls, privacy preservation rules, encryption rules and data retention policies. Figure 26 shows the logical position of the Policies and Data Governance module within the HORSE architecture.

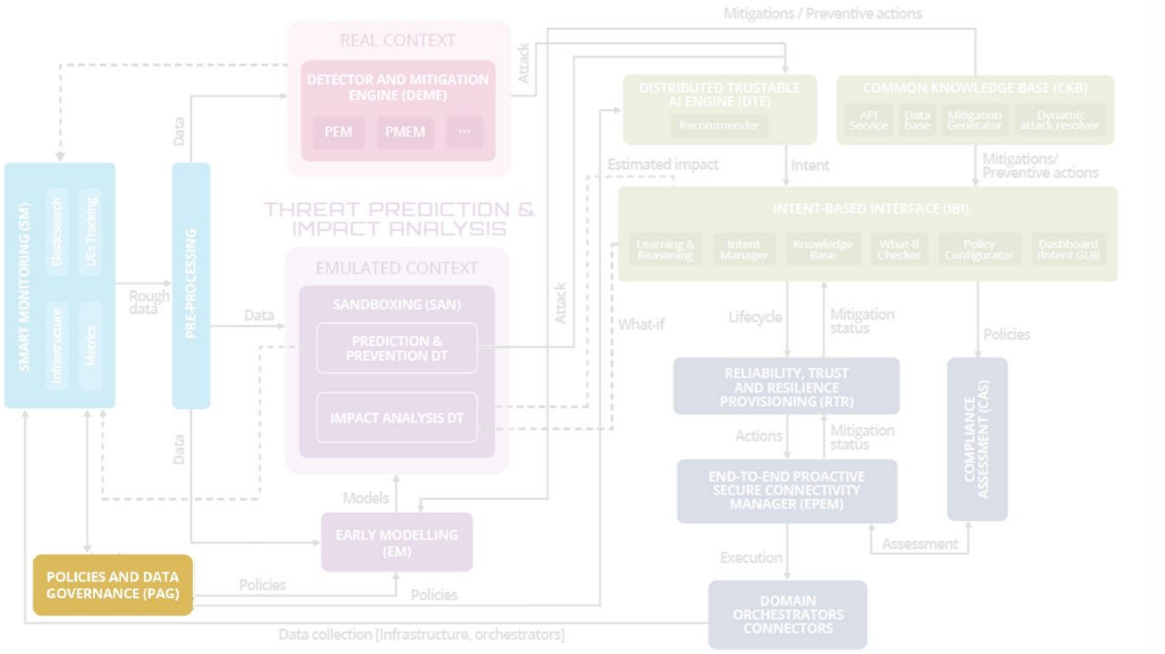


Figure 26: The logical position of the Policies and Data Governance module within the HORSE architecture

PAG mechanisms include access policies, privacy preservation rules for preventing unintended disclosure of personal or corporate information, encryption rules that need to be applied over the data when in-transit and data retention policies.

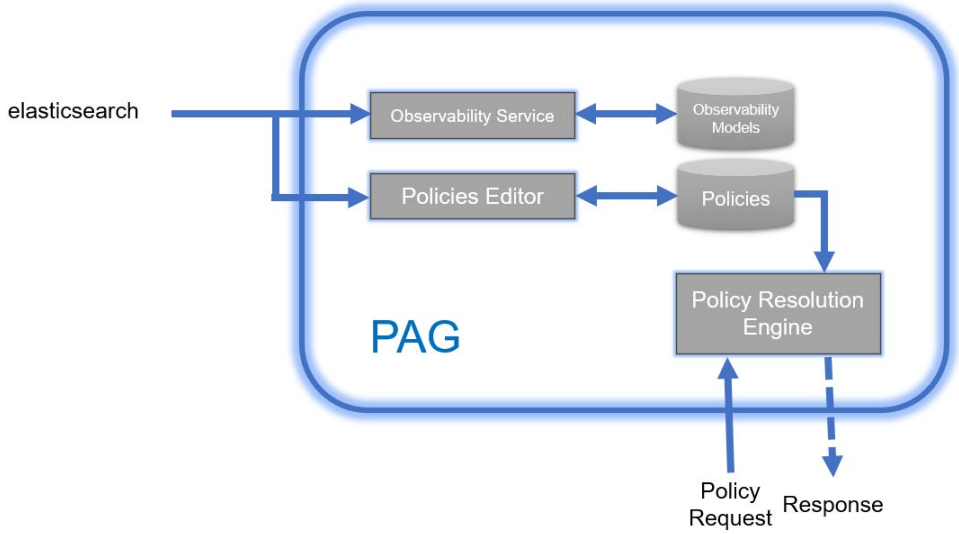


Figure 27: PAG component design

The PAG component, illustrated in Figure 27, will use a User Interface in the form of a Policies Editor. The Policies Editor will enable the user to define and update the access control policies which apply on the collected datasets. Such policies shall include data access policies based on the requestor's attributes at two different levels: per individual user and per component. Therefore, the data collected by the SM module and, once pre-processed, stored in the Elasticsearch, will be only accessible by authorized HORSE components.

Additionally, the PAG shall implement privacy policies based on sensitive and/or potentially identifying information inside datasets, encryption preferences and data retention rules (e.g., deletion of a dataset after a certain period).

Furthermore, the PAG component will continuously examine the collected datasets and provide information to the user regarding the freshness (e.g., date/time of last update) and the quality (high/low) of the collected datasets, through the Observability Service. The results of the Observability Service will be based on pre-defined rules expressed as observability models.

Once defined by the user, the policies will be stored in the Policies repository. The Policies repository does not store datasets per se; datasets are stored in Elasticsearch.

Finally, the PAG component will implement a Policy Resolution Engine which will run in the background and will be responsible for resolving the defined policies. The Policy Resolution Engine will communicate via REST APIs with other components (DTE, EM) in order to send or receive data.

3.2.2.5 Distributed Trustable AI Engine

The Distributed Trustable Engine is part of the intelligence module of HORSE. Its main goal is to analyze all related data from the Detection and Mitigation Engine (DEME) and define the appropriate set of actions that are later passed to the IBI in the form of high-level intents. To this end, DEME sends periodically to the DTE network measurements with a confidence interval, in the case where a new threat has been detected. Afterwards, two types of intents can be generated: a) predictive intents, where network reconfiguration policies are defined to avoid the appearance of a specific type of attack and b) mitigation intents, where the goal is now to fully mitigate a new threat.

The initial specifications of this module included the ability to train and execute various machine learning (ML) algorithms, to define the optimum remedy policy per case. In particular, during the first phase of implementation simple classification models were trained, to verify the correct functionality of the DTE. At a later stage, and in particular in IT-1, DNN models were used as well. In the final phase of the project (IT-2) the concept of Federated Learning (FL) has been introduced as well. In this case, and in order to speed up execution times, multiple network instances are used for model training and corresponding updates.

Moreover, model re-training is also possible until certain ML key performance indicators (KPIs) such as F1-score, mean square error (MSE), etc. are reached, as well as catalogue services where selected ML models per service or application are stored. Similar to FL, split learning techniques will be also adopted and combined, if possible, with FL in order to reduce the communication and computation cost without reductions on the accuracy of the calculations involved.

Figure 28 shows the logical position of the Distributed Trustable AI Engine module within the HORSE architecture.

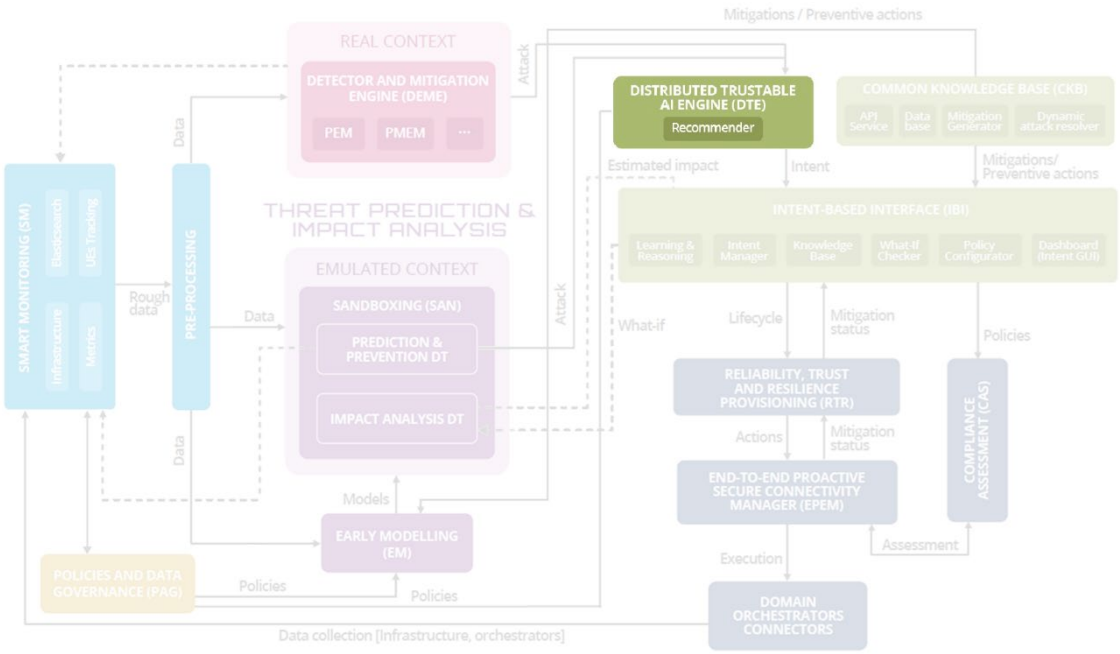


Figure 28: The logical position of the Distributed Trustable AI Engine module within the HORSE architecture

The implementation of the DTE will be in compliance with the latest 3GPP specifications defined in 3GPP TR 23.700-80 [22], where among others single slice optimization of network resources is assumed.

3.2.2.6 AI Secure and Trustable Orchestration

This section focuses on the part of the architecture referring to the STO module, addressing security and low-level orchestration topics. The subsections to be discussed are the following:

- Intent-based Interface (IBI): It is responsible for creating the policies that will form the HORSE lifecycle. It is also responsible of tracking the status of the attacks, mitigations, preventive actions, as well as the status of the 6G network.

- Common Knowledge Base (CKB): Centralized repository designed to store, manage, and enhance information about network vulnerabilities, attack patterns, mitigation strategies, and associated deployable actions.
- Compliance Assessment (CAS): This module verifies the correct compliance with the HORSE policies and the regulatory framework.
- Reliability, Trust and Resilience Provisioning (RTR): The RTR proposes an environment of reliability and trust in the system, implementing and applying mitigation and prevention actions.
- End-to-end Proactive Secure Connectivity Manager (ePEM): The ePEM module coordinates and manages actions on the HORSE service artifacts. To do this, it is in constant communication with the DOC.
- Domain Orchestrator Connectors (DOC): Domain Orchestrator Connectors are the intermediate point responsible for the communication and monitoring of the modules previously seen with various types of clusters, controllers and virtualization environments.

3.2.2.6.1 Intent-based Interface

The Intent-Based Interface (IBI) is responsible for receiving intents from human operators, for example, network administrators and other HORSE modules. In HORSE IT-1, IBI was integrated to handle QoS, mitigation, and prevention intents coming from human operators. The implementation involves translating user intents to network configurations. The configurations are applied in real time, while other HORSE modules handle the enforcement of the configurations. Regarding integration with other modules, IBI is also integrating with the DTE to receive mitigation and prevention intents.

Figure 29 below shows the logical position of the Intent-based Interface module within the HORSE architecture.

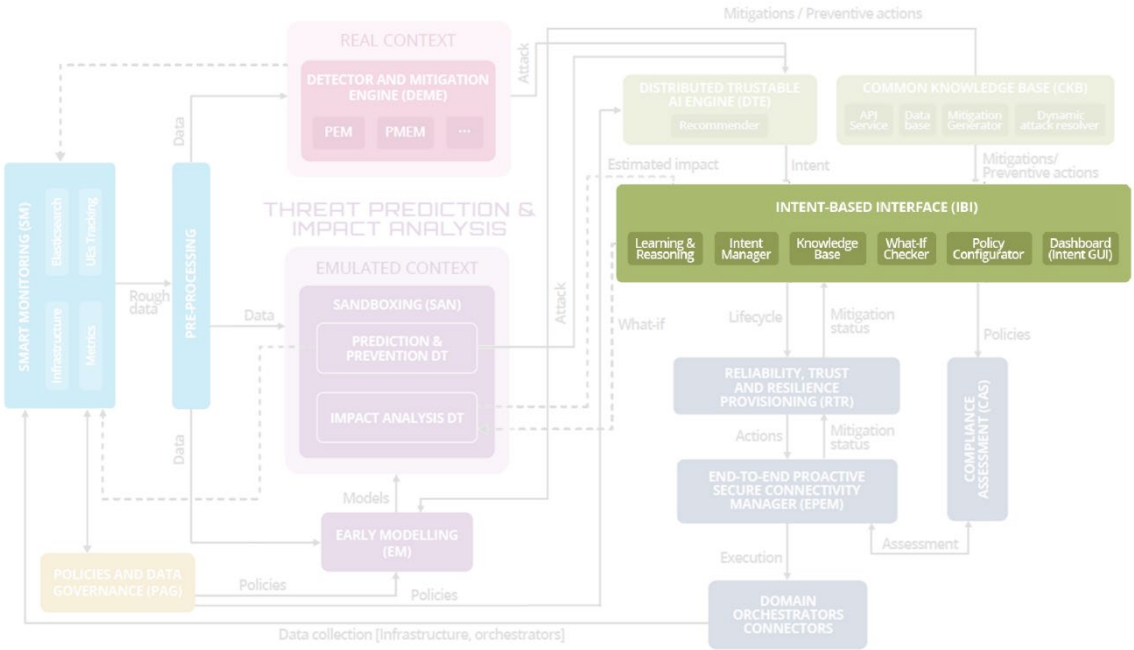


Figure 29: The logical position of the Intent-based Interface module within the HORSE architecture

The IBI module was designed to receive intents as structured JSON documents, and its functionalities are exposed through RESTful-based interfaces.

The IBI already includes the implementation of five out of the six components planned, namely the Dashboard, Intent Manager, Policy Configurator, What-if Checker and Common Knowledge Base. The Learning and Reasoning component will be the main focus of the IBI development in HORSE IT-2. The component will employ Q-Learning, a model-free reinforcement learning algorithm that is able to recommend optimal mitigation and prevention actions based on the previous performances of the actions. The IBI could receive feedback from the monitored infrastructure of the digital twin to update its Q-Values. It also considers the input of the DTE module as feedback. For example, an action will be considered successful if, during the period that the migration action is in place, no report or similar threat is received.

Regarding the integration of the IBI with other HORSE modules, new interfaces will be defined to allow connection with the Common Knowledge Database (CKB). The integration aims to allow the IBI to continuously update the list of possible mitigation and prevention actions from an external source for information. New interfaces will also be exposed to the Compliance Assessment (CAS) module to ensure that the selected policies by IBI will only be applied in the network if they align with regulatory frameworks. The integration with CAS will mainly affect the Policy Configurator design. However, other IBI components will need to be adjusted to deal with the situation when a policy is refused, implying the refining of the current policy or the selection of a new policy.

The IBI also currently integrates with the Impact Analysis Network Digital Twin, allowing the IBI to test effects of prevention policies before sending them for enforcement in the infrastructure. This is achieved with the component called "What-If Checker." However, due to the project's evolution, new test and scenarios will be supported, allowing the IBI to have better understanding of effects of proposed policies or actions using an isolated and sandboxed environment.

3.2.2.7 Common Knowledge Base

The Common Knowledge Base (CKB) in the HORSE architecture (see Figure 30) serves as a centralized repository designed to store, manage, and enhance information about network vulnerabilities, attack patterns, mitigation strategies, and associated deployable actions. It provides the foundation for automated threat detection and proactive threat mitigation in the HORSE platform.

The CKB is equipped with advanced functionalities, including integration with generative AI technologies, ensuring it evolves dynamically and remains a reliable resource for decision-making within the HORSE ecosystem.

The Common Knowledge Base encompasses the following features:

- Data Storage and Management: Centralized database for attacks, vulnerabilities, and mitigations.
- Knowledge Query Service: REST API-based interface enabling seamless integration and retrieval of CKB data. Supports high-level queries from HORSE components such as IBI and EM.
- AI-enhanced Insights: Incorporates generative AI models to synthesize and correlate data across multiple sources, generate mitigation strategies tailored to specific attack patterns, augment and update CKB entries automatically based on evolving threat landscapes.
- Threat Prioritization and Ranking: it utilizes machine learning algorithms to analyze patterns and prioritize mitigation actions based on severity and impact.

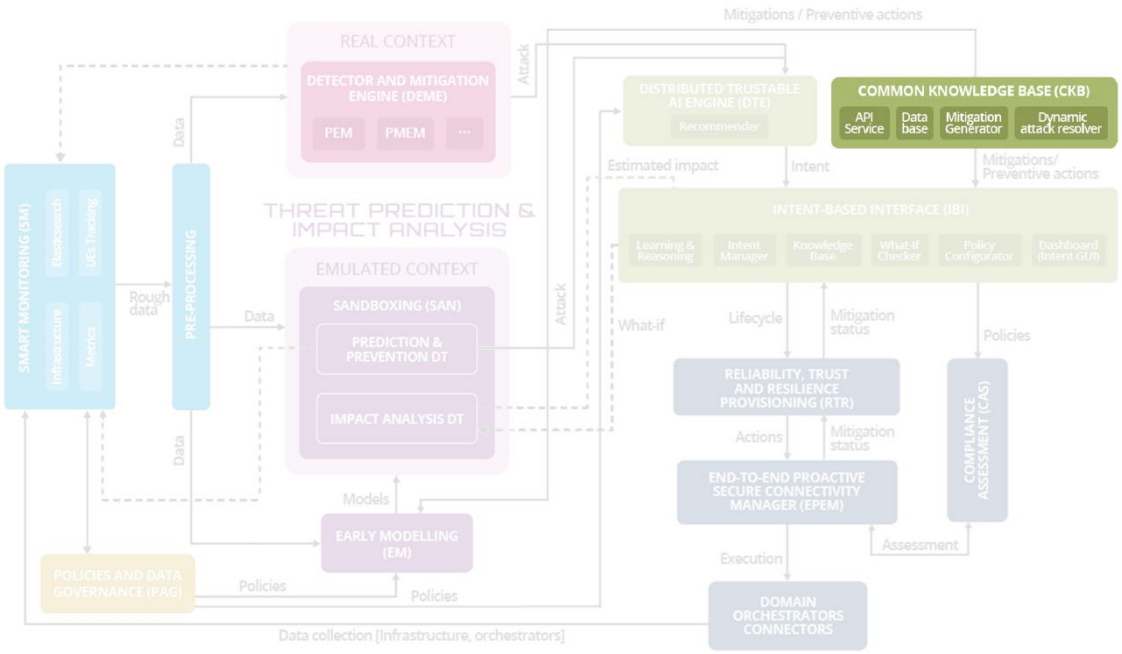


Figure 30: The logical position of the Common Knowledge Base module within the HORSE architecture

3.2.2.8 Compliance Assessment

The Compliance Assessment (CAS) module within HORSE holds a crucial role in ensuring the harmonization of security policies and solutions generated by the Trustable AI engine with the relevant regulatory framework. The Intent-based Interface (IBI) within HORSE proposes high-level network policies based on given requirements and intent, while CAS will validate above-mentioned policies against regulatory standards to ensure proper alignment. In the event of a non-compliant policy, CAS communicates back to IBI and creates a feedback loop for further refinement. This cross-module coordination ensures that HORSE's network operations can strike a balance between efficiency and compliance.

Figure 31 shows the logical position of the Compliance Assessment module within the HORSE architecture.

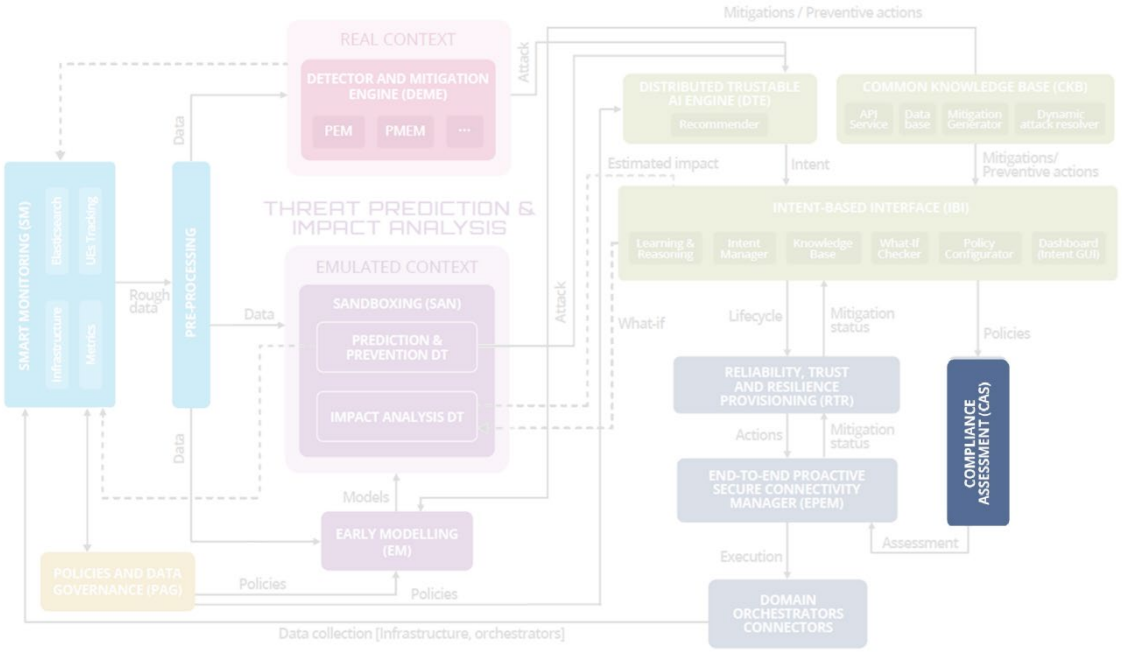


Figure 31: The logical position of the Compliance Assessment module within the HORSE architecture

Specifically integrated within the STO (Security Trust Orchestrator) module, CAS takes on the responsibility of validating, on the actual infrastructure, that the planned actions are in accordance with the policies outlined by IBI. By effectively acting as a compliance gatekeeper, CAS not only confirms the alignment of IBI-defined policies with the regulatory framework as mentioned earlier, but also acts as a communication bridge with the Policy Configurator sub-module. For example, CAS may verify that inputted security policies adhere to international standards such as the 3GPP's security specifications, safeguarding against threats like network attacks and data breaches in a 5G context [23]. The core functional idea will be exploited to pave the path towards regulatory compliance assessment in 6G networks. This collaboration between CAS and the Policy Configurator ensures that the policies selected for deployment are not only matched with the given intents and requirements but are also vetted to guarantee compliance with the applicable regulatory standards, enhancing the overall security posture of the HORSE system.

3.2.2.9 Reliability, Trust and Resilience Provisioning

The Reliability, Trust, and Resilience (RTR) Provisioning component plays a pivotal role in the architecture, acting as the intermediary between the high-level intent-based instructions provided by the Intent-Based Interface (IBI) and the execution of these instructions by downstream components like ePEM and DOC It provides a secure and structured framework for translating user intents expressed in natural language into actionable, enforceable mitigation commands.

Figure 32 shows the logical position of the Reliability, Trust and Resilience Provisioning module within the HORSE architecture.

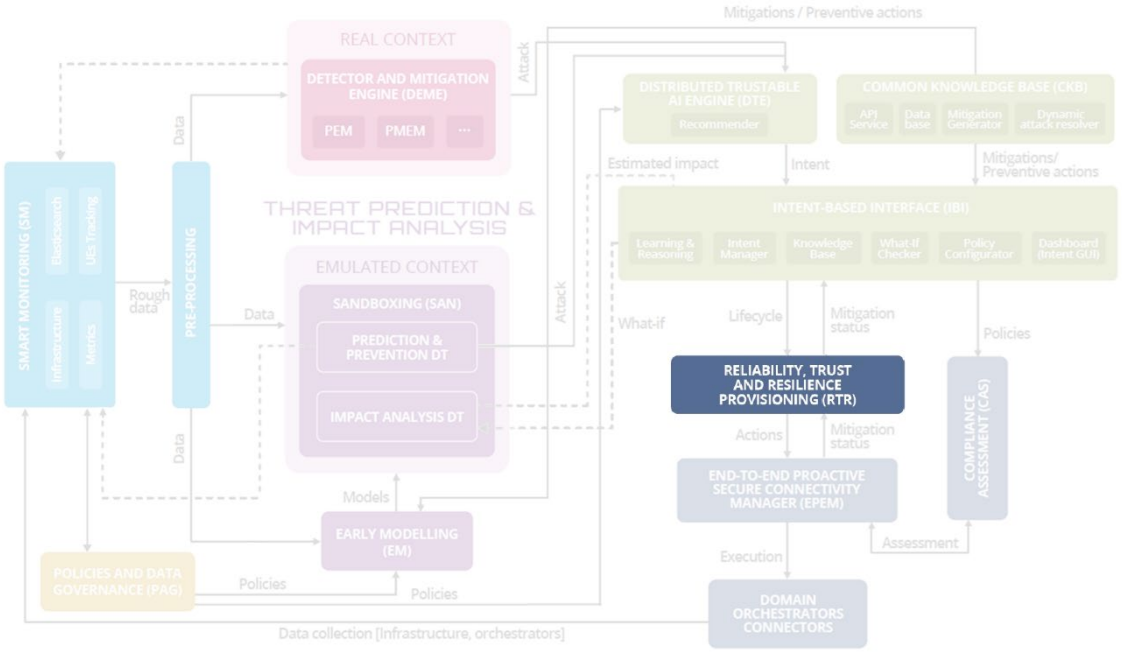


Figure 32: The logical position of the Reliability, Trust and Resilience Provisioning module within the HORSE architecture

The RTR component exposes its functionalities through an API endpoint, accessible via an API gateway. This endpoint is specifically designed to handle incoming mitigation actions from the IBI and transform them into a format that the subsequent components can enforce. This ensures seamless and automated translation of high-level user intents into network policies and actions.

Security is a key consideration in RTR's design. The component implements OAuth2 authentication to ensure that its API functions are securely accessed and used only by authorized actors. By leveraging OAuth2, RTR provides token-based access control, adding a layer of protection to its operations and safeguarding the integrity of the mitigation process. Internally, the component is capable of keeping track of mitigation actions at all stages of their lifecycle. These stages include actions in progress, successfully implemented actions, and actions that have failed to be enforced. This status tracking is critical for maintaining reliability and ensuring accountability, as it enables RTR to provide detailed feedback about the state of each mitigation action.

The RTR is composed of two containerized subsystems. The first container hosts the API application and its dependencies, encapsulating the logic for transforming and managing mitigation actions. This includes parsing the natural language input, applying transformation rules, and forwarding the formatted actions to downstream components for enforcement. The second container supports the OAuth2 functionality by hosting a containerized database, which manages the tokens and credentials required for secure API access. This separation ensures modularity, making it easier to scale or modify individual subsystems without disrupting the component's overall functionality.

As depicted in Figure 33 which displays the RTRs API endpoints functionality as well as in Figure 34 which describes the internal workflows overview of the RTR component, RTR's architecture includes detailed interactions between its API endpoints and other components. These interactions are orchestrated to ensure seamless communication and execution. For example, each mitigation action is assigned a unique action ID, enabling precise tracking and status updates. The ePEM component utilizes the RTR API to periodically update the status

of actions, ensuring that their progress is monitored and reflected accurately. Internally, RTR includes operations to transform mitigation actions into enforceable formats, which are then forwarded to the appropriate execution engines. This end-to-end flow of actions, from receipt to enforcement and tracking, highlights RTR's critical role in ensuring the reliability, trust, and resilience of the network management system.

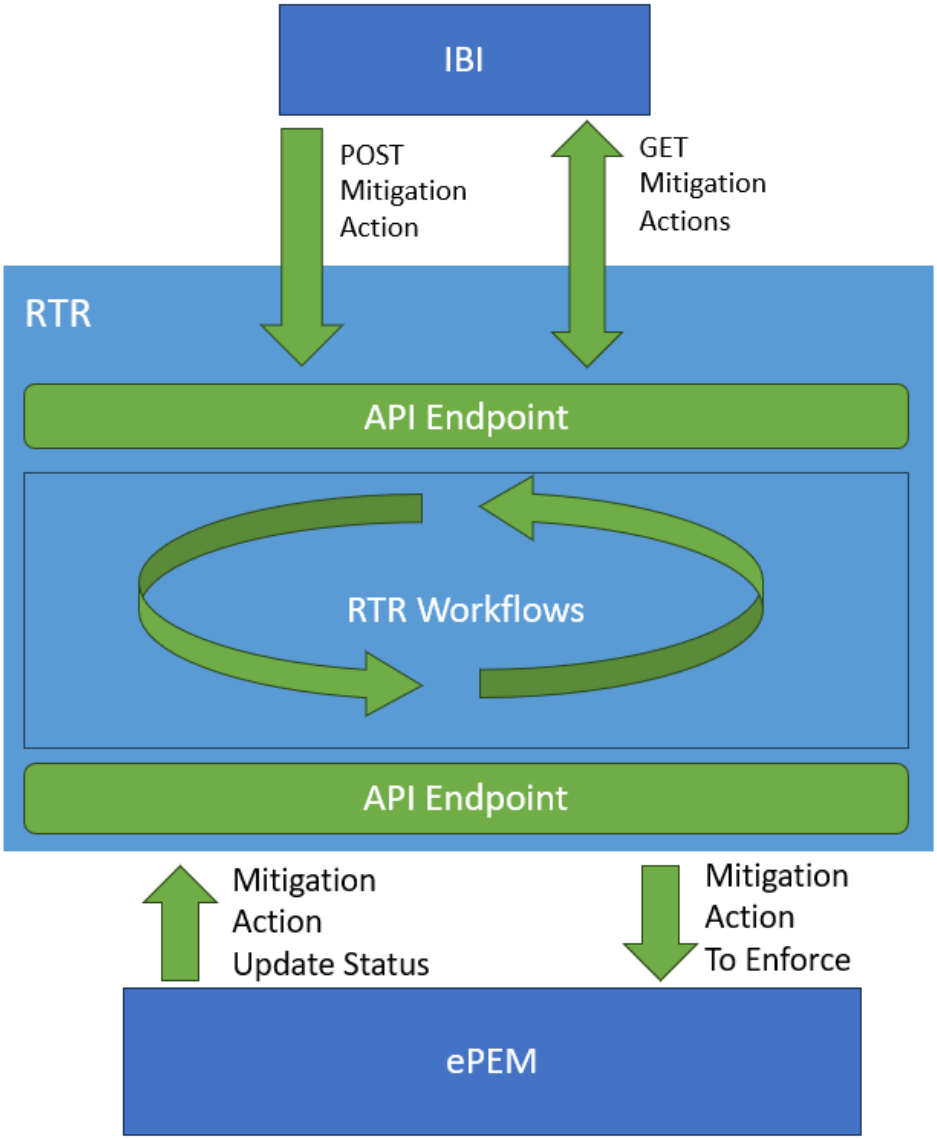


Figure 33: RTR API endpoints

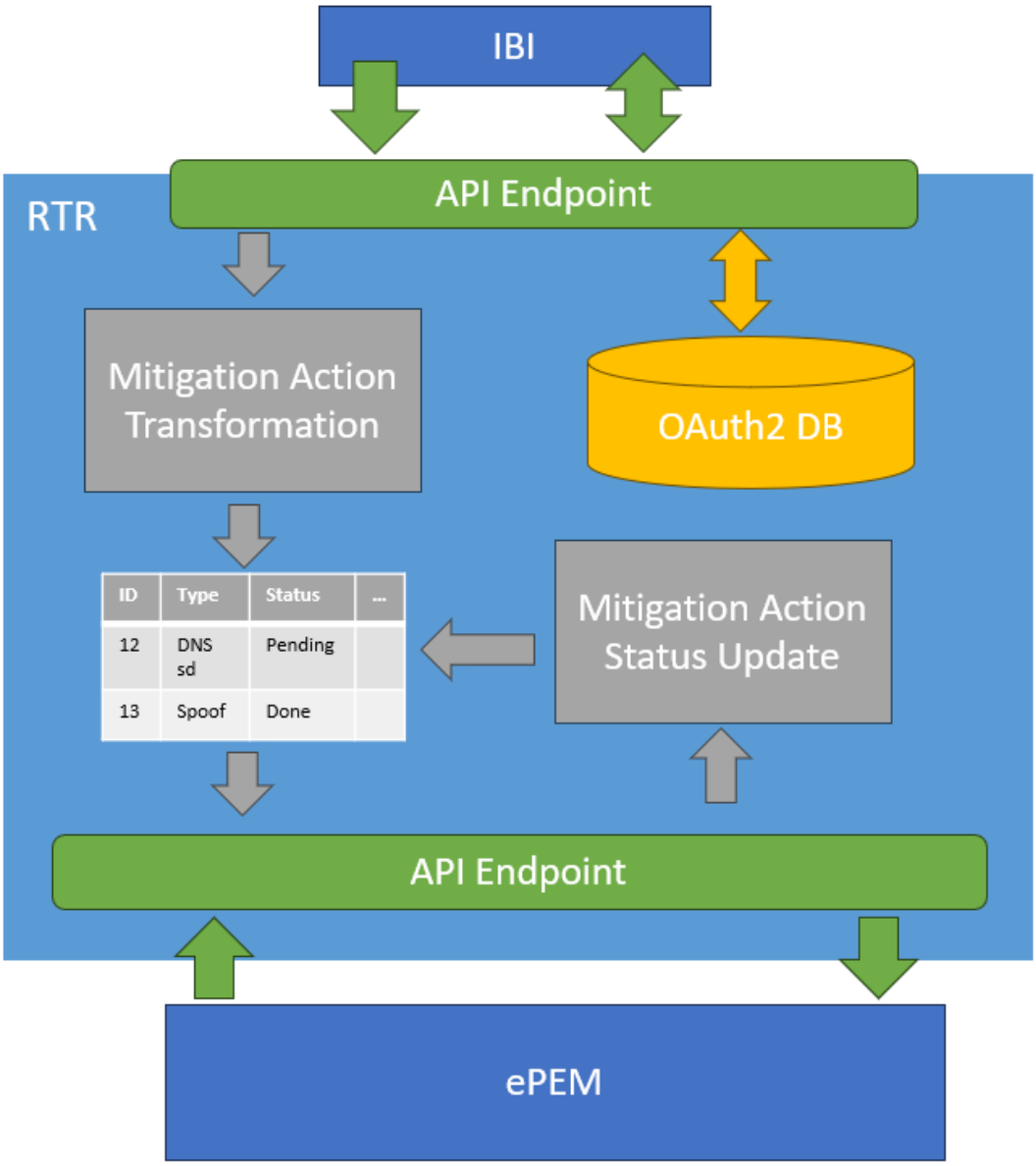


Figure 34:RTR internal workflows overview

3.2.2.10 End-to-end Proactive Secure Connectivity Manager

The End-to-End Proactive Secure Connectivity Manager (ePEM) emerges as a critical architectural component within the HORSE security perimeter, designed to orchestrate actions and provide comprehensive observability across heterogeneous and distributed network elements that collectively constitute end-to-end services. By seamlessly integrating with various domain orchestrators and controllers, the ePEM empowers the HORSE platform to proactively address security threats, optimize resource utilization, and enhance overall network resilience. Figure 35 shows the logical position of the End-to-end Proactive Secure Connectivity Manager module within the HORSE architecture.

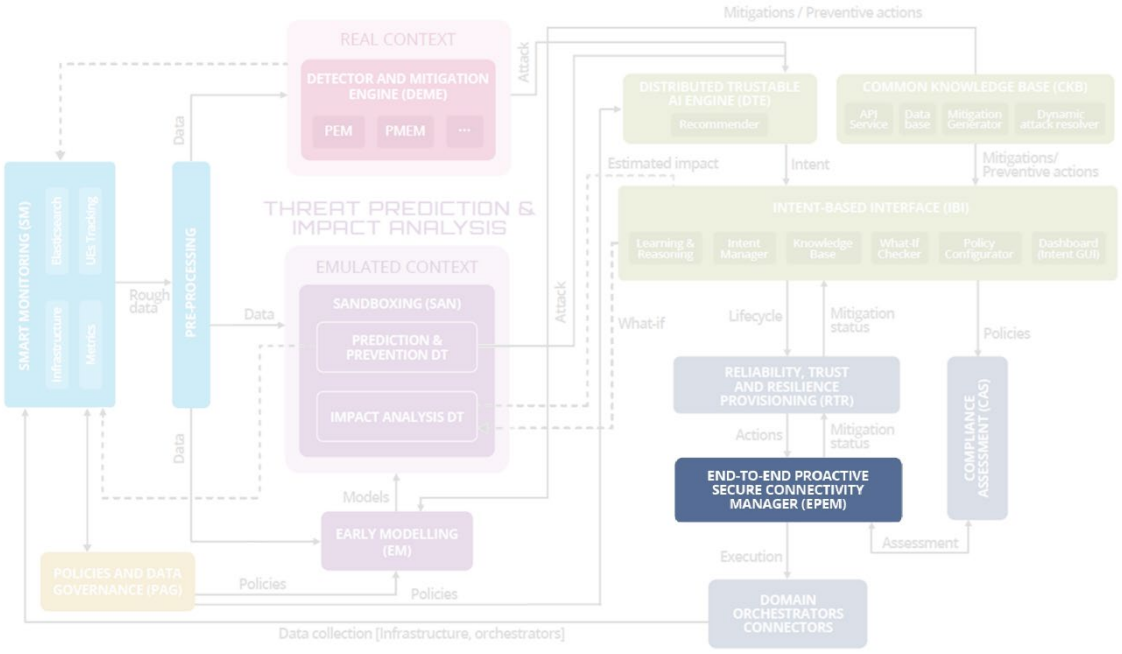


Figure 35: The logical position of the End-to-end Proactive Secure Connectivity Manager module within the HORSE architecture

At its core, the ePEM maintains a detailed understanding of the network's logical topology, encompassing both wide-area connectivity and VIM-level intricacies. This granular knowledge enables the ePEM to track the localization and degrees of freedom afforded to individual network functions and application components within the HORSE security perimeter. By leveraging this topological information, the ePEM can effectively manage the lifecycle of NFV services and resources, ensuring optimal performance and security.

Furthermore, the ePEM assumes a pivotal role in homogenizing and correlating diverse information streams from multiple sources. This harmonization process culminates in a unified and coherent end-to-end view of the services managed by the HORSE platform. By simplifying complex network topologies, the ePEM empowers network operators to gain deeper insights into their infrastructure, facilitating efficient troubleshooting and proactive maintenance.

A key aspect of the ePEM's functionality lies in its ability to autonomously acquire and process information about available actions and primitives from orchestrators and controllers. This knowledge empowers the ePEM to construct a comprehensive catalog of meta-actions, which can be leveraged by the HORSE Platform to formulate robust contingency plans in response to security breaches or network failures. By mapping services and artifact groups to predefined blueprint profiles, the ePEM can proactively identify potential vulnerabilities and devise targeted mitigation strategies.

In addition to its proactive security capabilities, the ePEM plays a crucial role in optimizing network resource utilization. By collaborating with energy consumption monitoring systems, the ePEM can map energy usage to specific topology elements and hosted artifacts. This granular visibility enables the ePEM to identify opportunities for energy savings and optimize resource allocation, ultimately reducing operational costs.

The ePEM's ability to dynamically inject and remove VIM-level operators and sidecar containers empowers the HORSE Platform to adapt to evolving security threats and performance requirements. By strategically deploying these components, the ePEM can enhance network observability, bolster security posture, and optimize resource utilization, ensuring the ongoing reliability and resilience of critical services.

The ePEM plays a crucial role in bolstering the security posture of the HORSE network. By continuously monitoring network traffic and analyzing security logs, the ePEM can detect anomalies and potential threats in real-time. This proactive approach enables the ePEM to initiate timely responses, such as blocking malicious traffic, isolating compromised devices, and deploying security patches.

Furthermore, the ePEM can leverage machine learning algorithms to identify emerging threats and develop adaptive security policies. By analyzing historical data and identifying patterns, the ePEM can anticipate future attacks and proactively implement countermeasures. This proactive approach significantly enhances the network's resilience against cyber threats.

Beyond security, the ePEM contributes to the overall performance and efficiency of the HORSE network. By optimizing resource allocation and traffic routing, the ePEM can minimize latency and maximize throughput. Additionally, the ePEM can automate routine network management tasks, reducing human error and freeing up valuable resources.

By leveraging advanced analytics techniques, the ePEM can identify performance bottlenecks and potential issues before they impact service delivery. This proactive approach enables network operators to take corrective actions and prevent service disruptions.

Finally, considering the role of ePEM inside the HORSE workflows, these are the main responsibilities:

- Policy Assessment and Execution: ePEM is responsible for assessing the "policies" generated during the threat prediction process. These policies likely outline preventive actions or countermeasures designed to mitigate the predicted threat.
- Execution of Actions: Once ePEM determines that the policies are suitable, it initiates the execution of the prescribed actions. This could involve various security measures, such as adjusting firewall rules, blocking suspicious traffic, or alerting security personnel.

3.2.2.11 Domain Orchestrators Connectors

The logical position of the Domain Orchestrators Connectors module within the HORSE architecture is depicted in Figure 36.

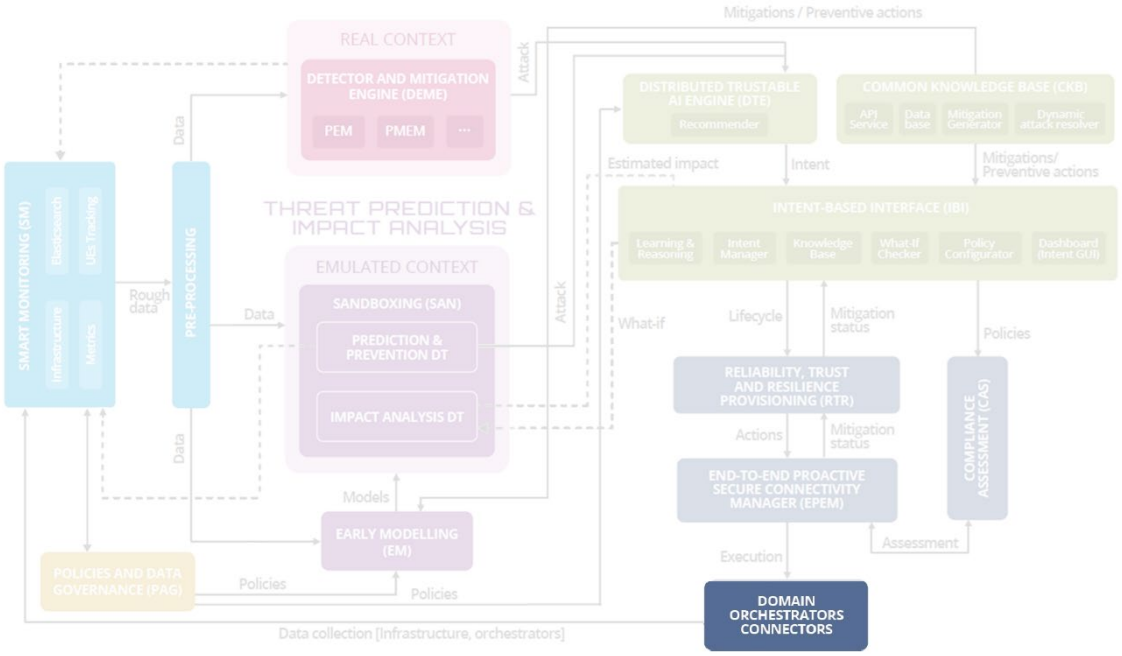


Figure 36: The logical position of the Domain Orchestrators Connectors module within the HORSE architecture

In the context of 6G networks, there is a great diversity of environments and infrastructures in which to make use of the wide connectivity that these networks will offer, this aspect hinders and makes it difficult to manage them due to their high heterogeneity. DOC seeks to simplify this task, unifying the different existing methods for the management of mitigation actions, with the main goal of providing an abstraction layer to the components of the upper layers. These components, which are explained in a deep way throughout this document, implement their functionalities in a simpler and independent way from the underlying infrastructure provider.

DOC module provides a unified interface to HORSE context, offering a way to enforce mitigation actions abstracting from existing infrastructure. In the case of HORSE project, there are three different infrastructure providers (UMU, UPC and CNIT) that are managed by different Northbound Interfaces, using the DOC's algorithm it is possible to manage all of them from a single entity, abstracting all their complexity to the upper layers.

That algorithm is easily extendable to other infrastructure providers due to its modular development, thus achieving a high reusability of the component for other projects and scenarios, which is vital in the 6G environment which is not defined in a definitive way and it is susceptible to relevant changes during the standardization period that will take place in the following years after the end of the project, reducing the risk of becoming obsolete in consecutive years.

4 "Canonical" Workflow: working together

This section presents the canonical workflows defined to test the core functionalities of the HORSE architecture, while also defining the operational data flow. To this end, two different workflows have been defined, for the detection and prediction of threats. These workflows build upon those defined for IT-1 [1], but have been extended to consider a complete scenario involving all HORSE components.

Both workflows are intended as templates to fuel more specific, lower-level workflows, as well as for creating workflows tailored to the use cases. The two proposed workflows, presented as sequence diagrams, are described next.

Additionally, to facilitate the understanding of the different steps in the workflows, they are presented with a distinction between communications based on the key functionalities provided by the HORSE framework, as depicted in Figure 37. These main functionalities include:

- Data collection and pre-processing
- Threat detection
- Threat prediction and impact analysis
- Recommendation
- Execution

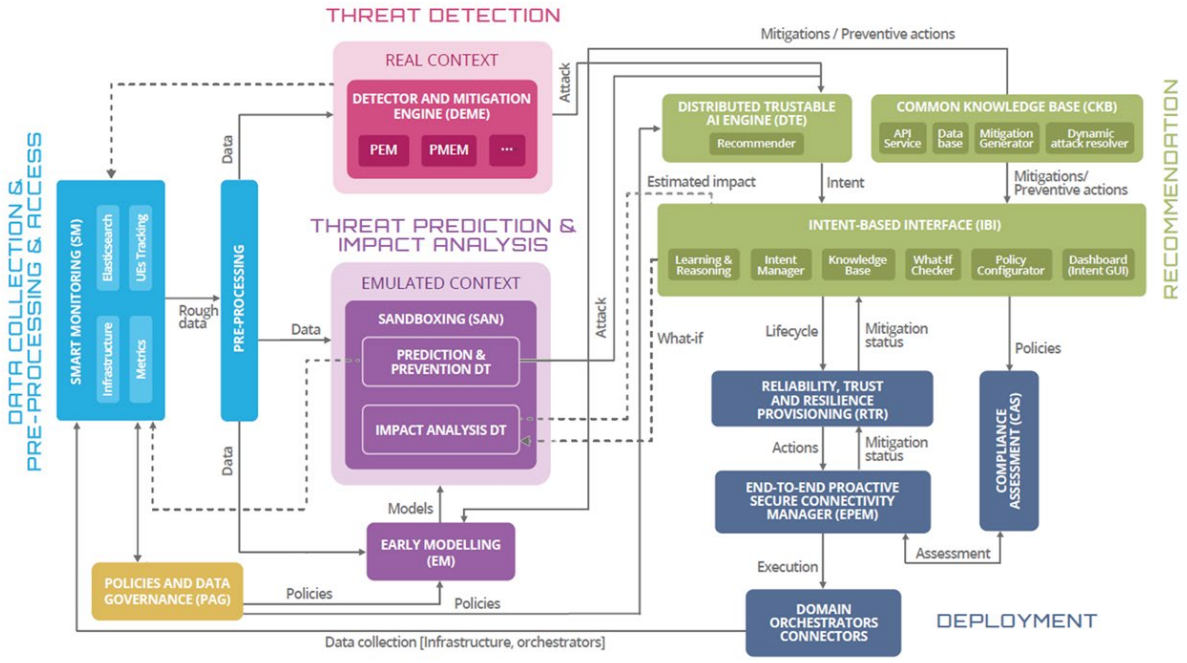


Figure 37. HORSE architecture – main building blocks

4.1 Threat Detection Workflow

Three different workflows are defined for threat detection, depending on whether the policies defined to mitigate the detected attack are compliant, non-compliant of partially compliant with the HORSE policies and the regulatory framework.

In the threat detection workflow, we can differentiate two phases: (i) the Preparation and retraining phase, that has as objective to train the models of the HORSE AI-based components, and (ii) the threat detection phase that illustrates the detection and mitigation of a network threat.

In the preparation and mitigation phases, see Figure 38, the workflow starts by gathering measurements from the infrastructure and/or orchestrators, in terms of rough data, which is sent to the Pre-processing module for normalization. The pre-processed data is stored in the Elasticsearch database of the SM component. The DEME then requests access to the datasets stored in the SM for training purposes. The PAG grants access to the datasets, and the SM sends them to the DEME for training purposes.

In the threat detection phase, the workflow starts by gathering measurements from the infrastructure. The SM module gathers the data from the infrastructure and/or the orchestrators and sends it to the Pre-processing module, which performs normalization tasks to unify all the received data. Once normalized, the data feeds the DEME module, where efficient threat and attack detection mechanisms are continuously running. In addition, the processed data is stored in the Elasticsearch within the SM component, which provides input for retraining the AI models used by the HORSE components. When a threat or an attack is detected, the DEME notifies the DTE, which generates the corresponding intent, transforming it into a readable layout.

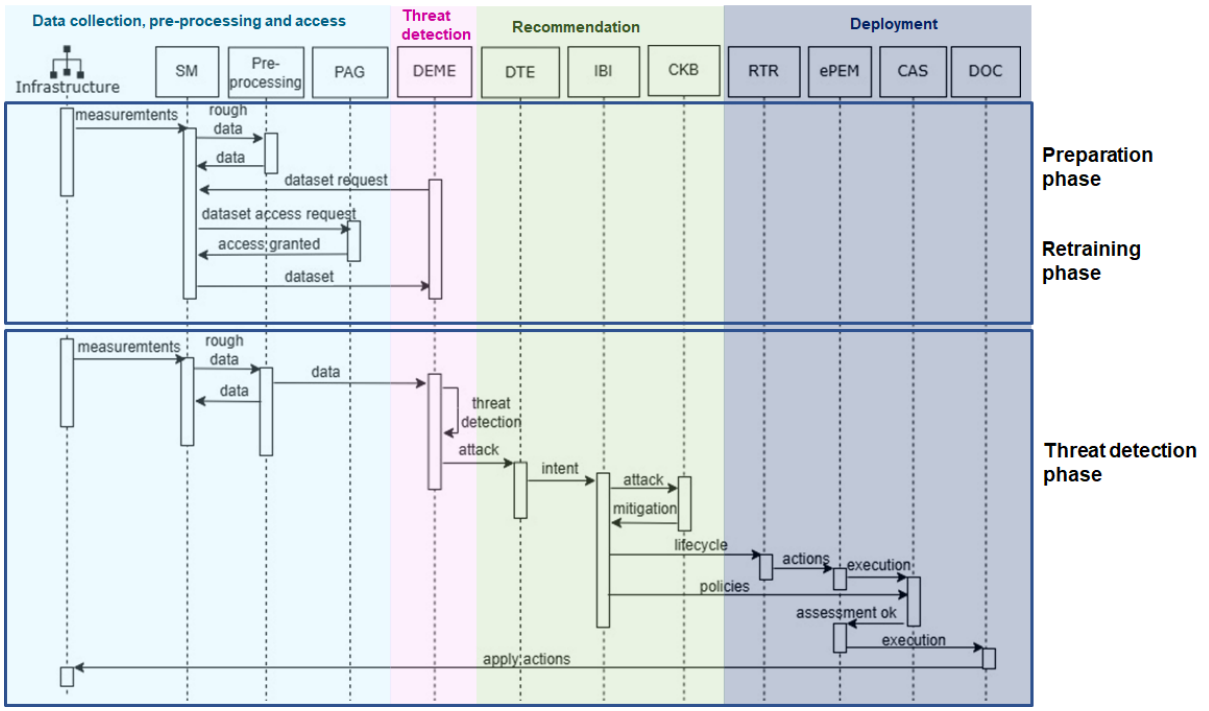


Figure 38: HORSE Threat Detection Workflow – Compliant policy

The intent is sent to the IBI, which queries the CKB for mitigation actions to be enforced in the infrastructure. It then generates the corresponding lifecycle of concrete actions, covering the whole set of steps to be taken to handle the detected attack or threat. This lifecycle is forwarded to the RTR, responsible for defining the concrete set of mitigation actions inferred from the previous lifecycle, to be deployed in the infrastructure. The set of actions is then sent to the

ePEM, which checks with the CAS if the execution policies align with the HORSE policies and the regulatory framework. If compliant, the DOC executes the required technologies and solutions in the infrastructure, as illustrated in Figure 38, to properly react to the detected attack triggering this workflow.

If the policies are not compliant with either the HORSE policies or the regulatory standards, as shown in Figure 39, the IBI is informed and generates an alternative lifecycle. This lifecycle is sent to the RTR to derive the necessary mitigation actions for deployment. The set of actions is then sent to the ePEM, which again checks compliance with the CAS. If compliant, the DOC executes the required technologies and solutions in the infrastructure, to properly react to the detected attack.

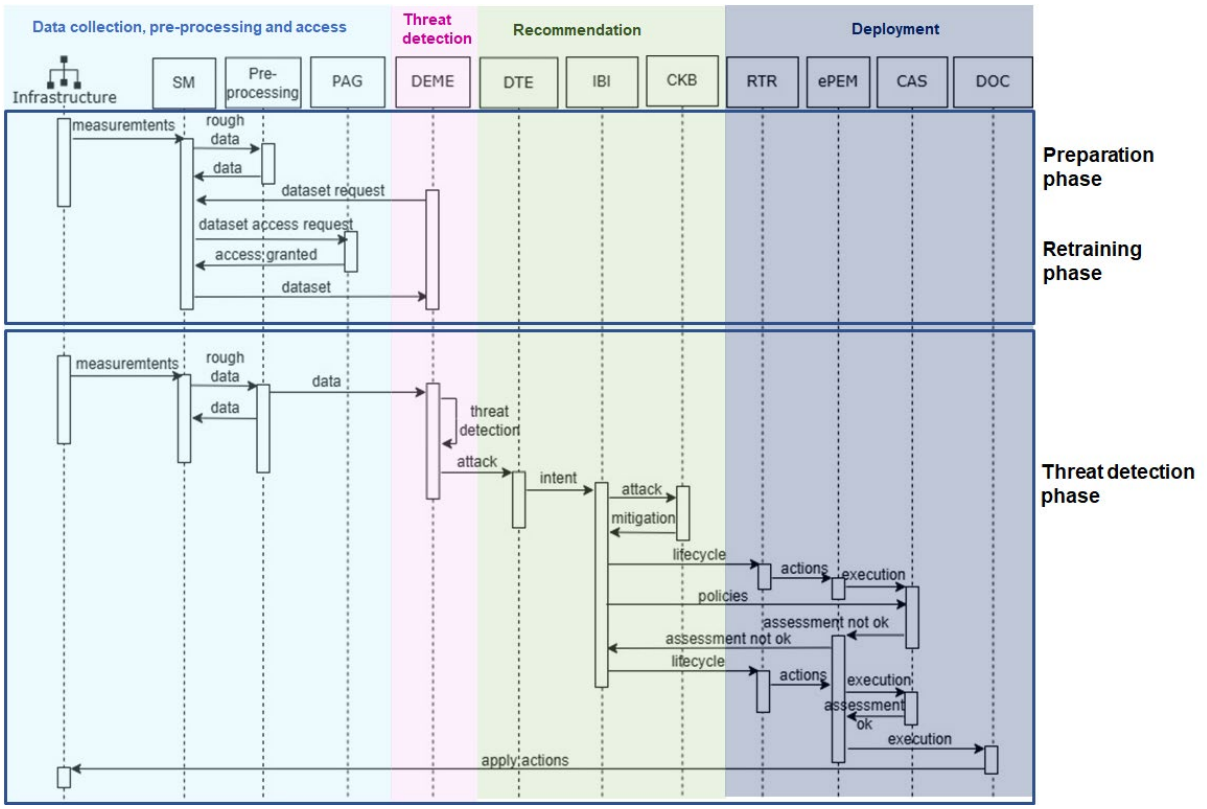


Figure 39: HORSE Threat Detection Workflow – Non-compliant policy

Finally, if the policies are partially compliant, as depicted in Figure 40, the IBI is notified and determines whether the policies can be enforced or if a refined version of the non-compliant policies is needed. If refinement is necessary, the non-compliant subset of policies is revised and sent to the RTR. The RTR generates the concrete set of mitigation actions and send them to the ePEM, which again checks compliance with the CAS. Once compliant, the DOC executes the necessary actions to mitigate the detected threat or attack.

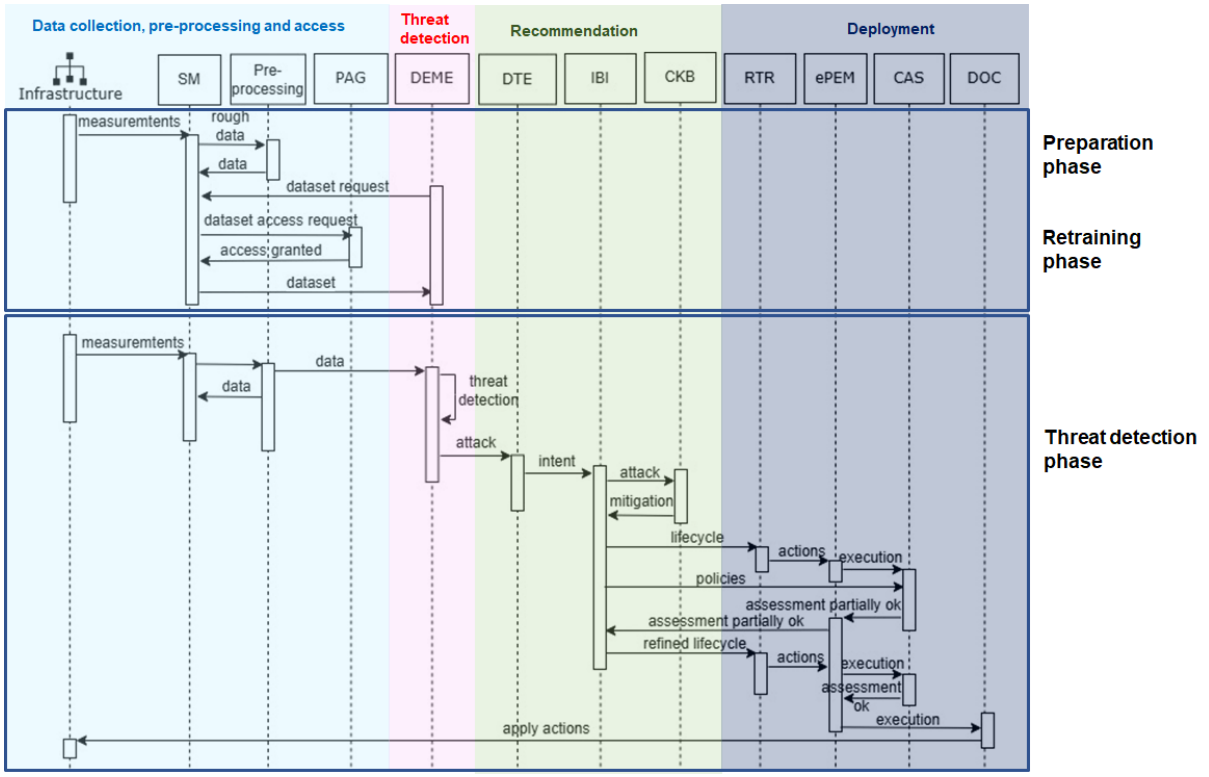


Figure 40: HORSE Threat Detection Workflow – Partially compliant policy

4.2 Threat Prediction Workflow

The HORSE Threat Prediction Workflow, presented in Figure 41, focuses on the prediction of attacks, and differentiates from the Threat Prediction Workflow in the assessment of the impact of the preventive actions in the NDT, before being enforced in the real infrastructure. Additionally, this workflow also illustrates the preparation and retraining phases for HORSE's AI-based components, such as the Prediction & Prevention NDT.

The workflow starts by gathering measurements from the infrastructure and/or orchestrators, in terms of rough data, which is sent to the Pre-processing module for normalization. The pre-processed data is stored in the Elasticsearch database of the SM component. The Prediction & Prevention NDT requests access to the datasets stored in the SM for training purposes. The PAG grants access to the datasets, and the SM sends them to the Prediction & Prevention NDT.

In the Threat Prediction phase, measurements from the infrastructure and/or the orchestrators are again gathered as raw data, which is sent to the Pre-processing module for normalization. Unlike detection workflow, where the data is used to detect attacks and threats, in this workflow the collected data is used to predict attacks or anomalies through the SAN module. Indeed, the normalized data received by the SAN module, is smartly processed by the Prediction & Prevention NDT. The main objective of this component is to predict that a threat or an attack are about to come with a certain probability. If a potential threat is predicted, the DTE generates the corresponding intent, and sends to the IBI. The IBI processes the intent and retrieves the preventive actions to be enforced from the CKB. As in the detection workflow, the IBI generates a lifecycle of specific preventive actions, containing the entire set of steps to be taken. However, unlike the detection workflow, recognized the fact that in this workflow the overall decision process will deal with estimated and non-completely accurate predictions, before being forwarded to the RTR, the lifecycle is sent to the SAN module, where the Impact Analysis

NDT runs the foreseen preventive actions into an emulated scenario, so a clearer overview of the real outcome of deploying such a lifecycle may be deeply observed. Indeed, the Impact Analysis NDT estimates the impact of executing the proactive actions (the lifecycle) in the emulated infrastructure, handling out the estimated impact to the IBI, which processes this estimation and evaluates if it would be acceptable, according to some specific and well-defined policies. In the case the impact is acceptable, the IBI sends the generated lifecycle to the RTR, which defines the concrete set of mitigation actions to be deployed in the infrastructure. The set of actions is then sent to the ePEM, which checks with the CAS to ensure the execution policies are compliant with the HORSE policies and with the regulatory framework. If compliant, the DOC executes the required technologies and solutions in the infrastructure, as illustrated in Figure 41, to proactively react to the predicted threat or attack.

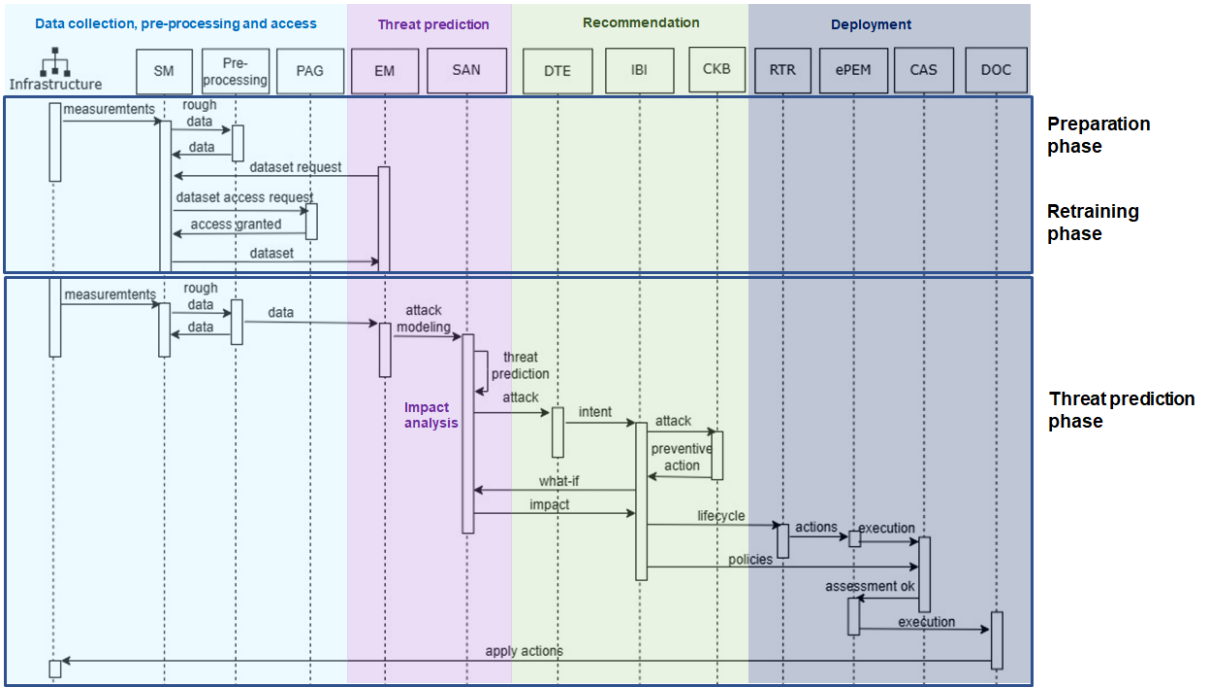


Figure 41: HORSE Threat Prediction Workflow

5 Use cases mapping to the HORSE architecture

The current section presents the two selected use cases from two project pilots, namely the Secure Smart LRT Systems (SS-LRT) and Remote Rendering to Power XR Industrial ($R^2$2XRI), integrated in the HORSE platform, allowing the validation of the requirements, assuring this way to realize the proper behavior of the HORSE components according with the defined architecture.

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

The first use case of the HORSE project takes advantage of the components described in previous deliverable D2.2 [1] and will allow the validation of several HORSE requirements as shown in Deliverable D2.3 [2].

The use case is based in Metro operations – Light Rail Transit and four scenarios related to specific Metro applications were selected, allowing the integration, in the HORSE environment, of these operations, such as Passenger Information, Automatic Vehicle Localization and video streaming from the tram stops/stations.

Therefore, since all scenarios are based on traffic network over a private 5G/6G network the HORSE architecture and the HORSE components such as IBI, PIL, SAN, EM, PEM, DTE, STO or SM must be validated to assure resilience and smart security capabilities to future networks allowing to Metro networks to take advantage of such benefits.

Figure 42 illustrates the integration of the use case 1, based on real scenarios of Metro systems, such as Dublin/LUAS (Ireland) and Bergen (Norway), by EFACEC, where is highlighted the communication between the tram stops and vehicles with the Operational Command Centre (servers and workstations)

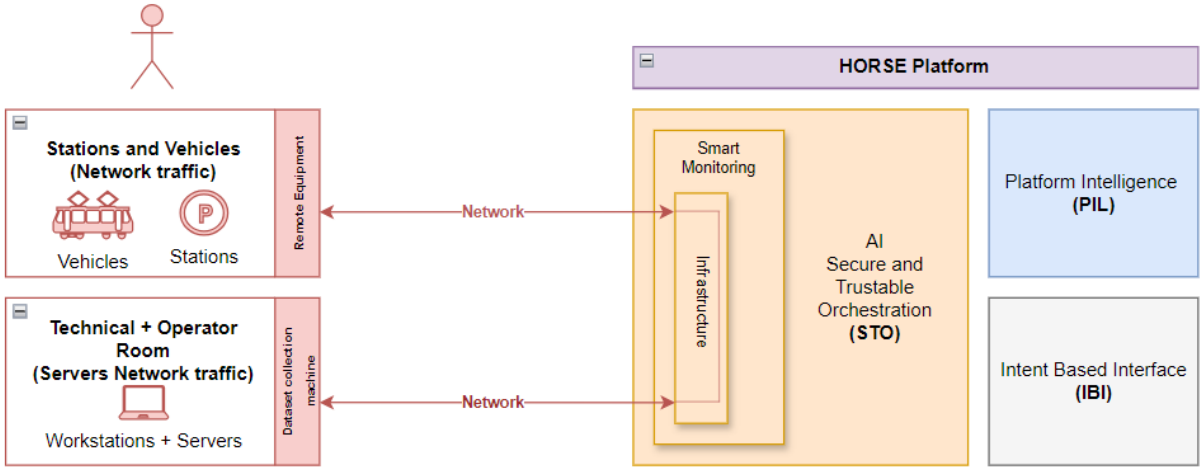


Figure 42: HORSE Architecture Mapping and Integration: use case 1 SS-LRT

As described in the Deliverable D2.3 [2] the HORSE demonstrations and validation will occur at laboratory level with an integration of three testbeds: i) UMU (tram stops), ii) UPC (HORSE components) and iii) EFACEC (Operational Command Center), using network traffic for three different services/applications: i) Video streaming from tram stops to OCC (surveillance/security application), ii) vehicle localization messages (Automatic Vehicle Localization System) and iii) Information messages to tram stop displays (PID-Passenger Information System). Figure 43 illustrates the reference solution for the Use Case 1, showing

the integration of the three environments: UPC testbed, UMU testbed, and EFACEC Operational Command Center.

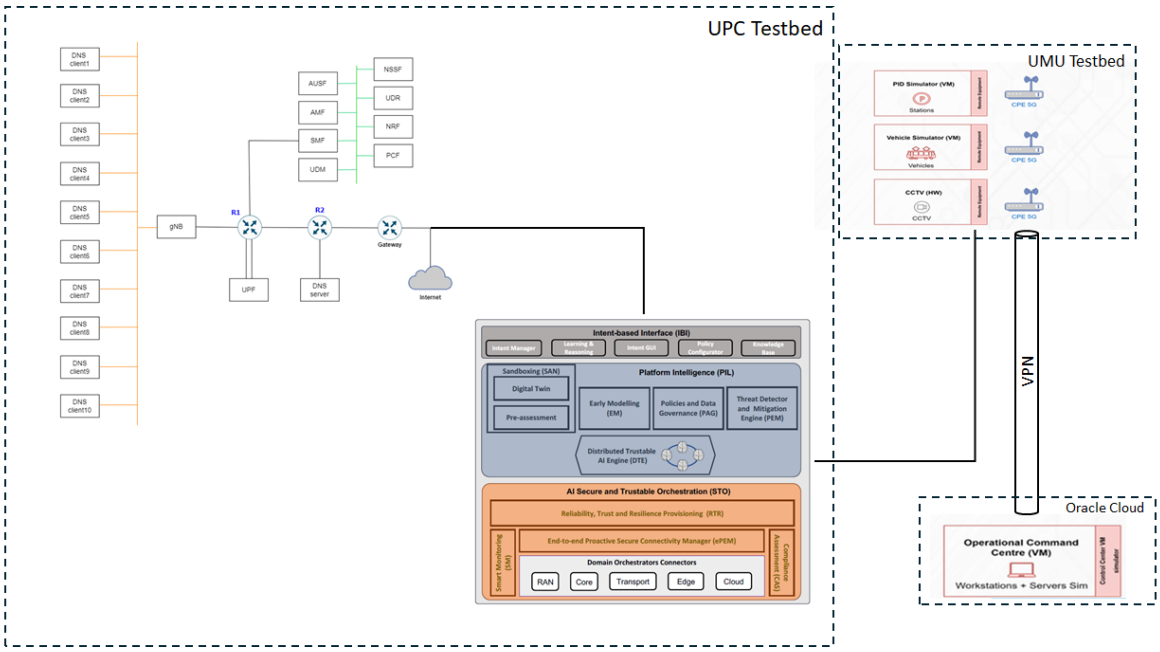


Figure 43: Use case 1 integration in the HORSE framework - UMU and UPC testbeds

Figure 44 shows the UMU testbed where the simulation of the described Metro applications, at tram stop levels, will be deployed,

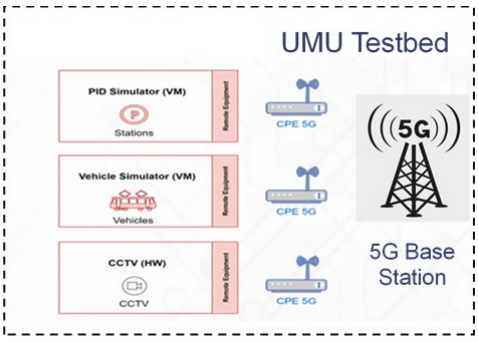


Figure 44: UMU testbed: use case 1 SS-LRT (tram stops)

Figure 45 illustrates the use case 1, showing the simulation of vehicle localization, using tools for simulation at vehicle level, allowing the vehicle representation in Metro Lines, as shown in the Command Center workstation. The information messages, exchanged between the tram stops and the OCC allows not only the vehicle localization but also to realize if the vehicles are delayed, in advance or due, according with the timetable defined for each Metro Line service.

Figure 45: OCC Vehicle localization: use case 1 SS-LRT

Figure 46 illustrates the use case 1, showing the normal behavior of the Passenger Information system where it is possible to realize the visualization of the scheduling for the vehicles in the OCC workstation and in a simulated display. The display messages are exchanged, using a network protocol, from the OCC to the tram stop's displays allowing the visualization of the estimated arrival time or messages sent by the metro operator to the tram stop display.

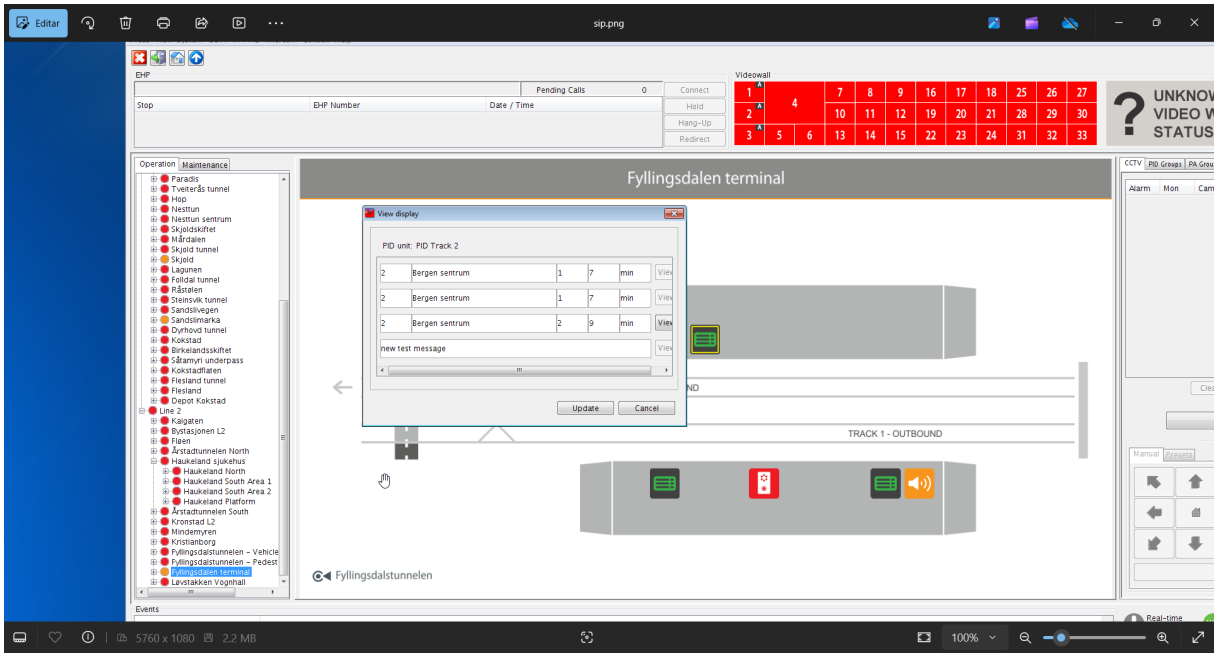
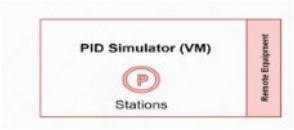


Figure 46: Passenger Information use case 1 SS-LRT (OCC visualization)

Using a developed tool as shown in Figure 46, it is also possible to visualize in a simulated display, the information sent by the OCC to the tram stop (manual messages from the operator or automatic messages of trams estimated arrival time). Figure 47 illustrates the Passenger Information simulated display.



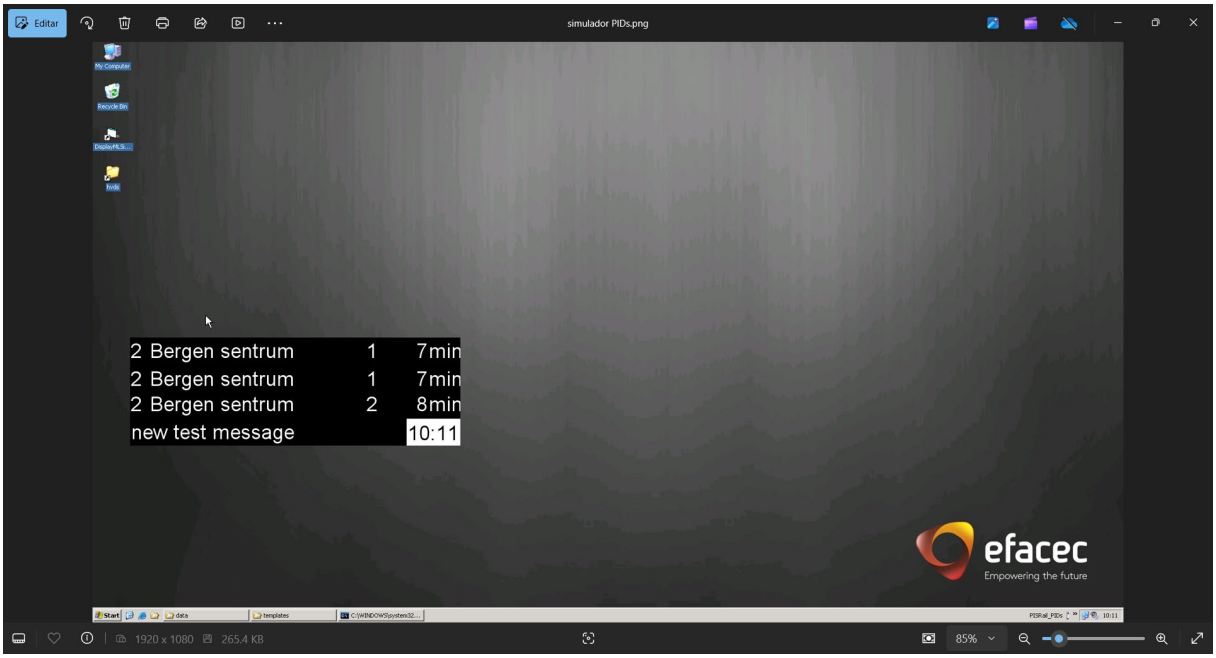


Figure 47: PID simulator: use case 1 SS-LRT (display simulation)

These scenarios intend to validate the HORSE platform behavior in the presence of cyber-attacks and the relevant impact in a Metro network services as well as the benefits of using the HORSE components to achieve resilience and security for future 5G/6G networks.

5.1.1 Data collection

In the scenarios described in previous section, first traffic captures were conducted during the 14 and 15th December 2024 in EFACEC premises. The captures were done using wireshark running on PID simulator virtual machine, that is present in the EFACEC laboratory. The traffic contains all packages exchanged between PID simulator application running on 10.110.0.196 and all necessary servers. Figure 48 shows relevant traffic related to PID simulator, this traffic has been filtered using the command ip.addr == 10.110.0.196.

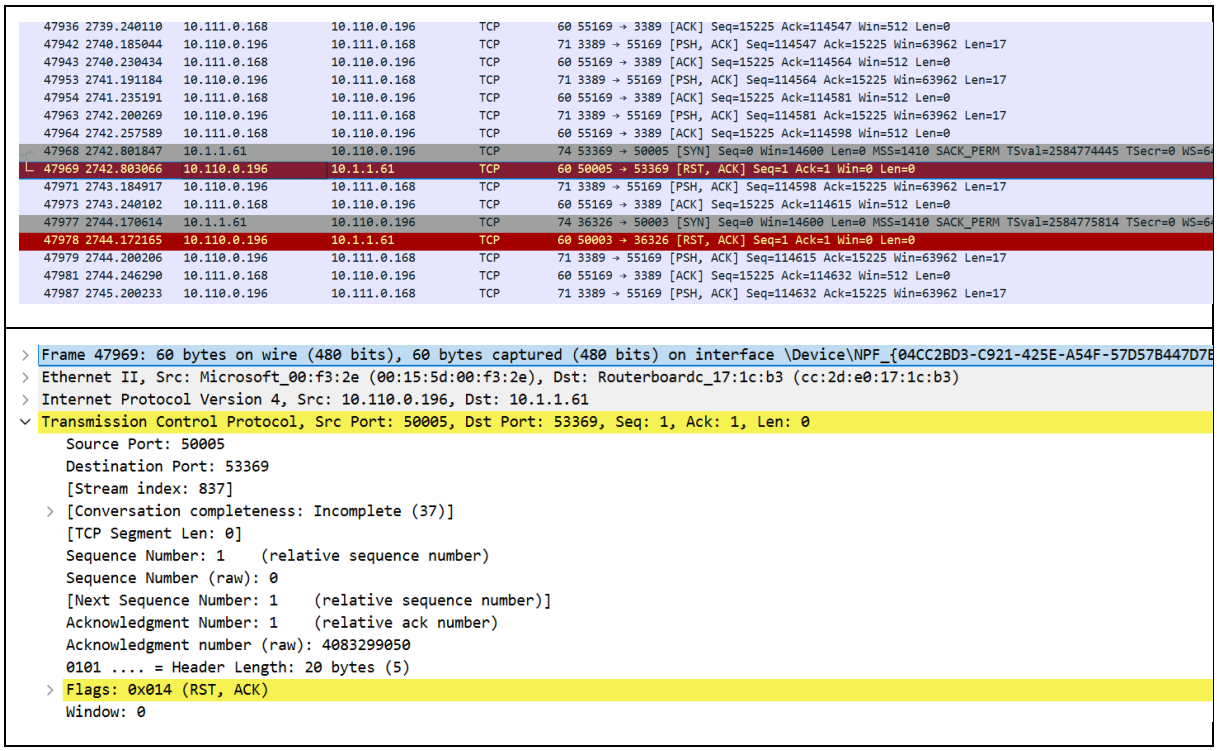


Figure 48. Traffic captures related to PID simulator

5.2 Use Case 2: Remote Rendering to Power XR Industrial

Use Case 2 focuses on leveraging remote rendering to enhance Extended Reality (XR) applications in industrial sectors. In today's rapidly evolving industrial landscape, organizations are turning to XR—encompassing Augmented, Virtual, and Mixed Realities—for immersive and cost-effective employee training in low-risk industrial environments. XR addresses the unique challenges faced by industries such as manufacturing, healthcare, construction, and aerospace, where traditional training methods like classroom sessions, simulations, and manuals often fall short. By offering interactive, hands-on learning experiences, XR allows employees to safely train in high-stakes scenarios involving hazardous materials or complex machinery without disrupting production. With features like rapid feedback, performance tracking, and scenario-based learning, XR ensures better knowledge retention, making it a critical tool for workforce development. However, while XR has a lot of potential, overcoming XR device limitations and scaling up the usage to becoming an industrial alternative can be difficult. Here's where XR streaming comes in.

This use case utilizes HOLO's Unity-based AR Engineering Application called Hololight Space to visualize 3D CAD designs on XR mobile devices like HoloLens. With Hololight Stream, the content is hosted on a workstation and rendered remotely and then the application is streamed to the XR device to overcome device limitations, while sensor data (e.g., head pose), audio input, etc. is sent from the client to the server.

The Hololight Stream Software Development Kit (SDK) is a remote rendering solution that enables real-time streaming of entire XR applications. By streaming entire applications, Hololight Stream enables the visualization and interaction with high polygonal, data-intensive content such as graphics-intensive 3D objects, 3D Computer-Aided Design (CAD) models or Building Information Modelling (BIM) data which would otherwise be unlikely on native

applications due to the limitation in the processing power of the Extended Reality (XR) devices.

The Hololight Stream SDK can be integrated into any Unity-based XR application. Its multi-device support and native Unity 3D integration streamline application development, saving time and effort while increasing the security and scalability of AR and VR applications. With Hololight Stream, user can run their Unity-built XR application on a powerful workstation, local server, or cloud-based infrastructure. It enables the secure streaming of AR/VR applications to all major AR/VR glasses and iOS devices in the market to visualize high fidelity content without down-sampling data. In summary:

- Streaming data to AR/VR devices at its original complexity, size, and quality, eliminating the need for extensive data preparation.
- Secure streaming of XR applications over networks to control and protect critical and sensitive data by never storing it on endpoints.
- Creating and deploying a wide range of XR applications across multiple devices, increasing user engagement and delivering more dynamic experiences.
- Ensuring to efficiently render demanding and resource-intensive XR contents.
- Speeding up XR application development with a device-agnostic approach, native Unity 3D integration, and rapid application deployment.

Figure 49 below shows an example of streamed data on Hololight Space as compared to the data on the native application on the AR glasses.

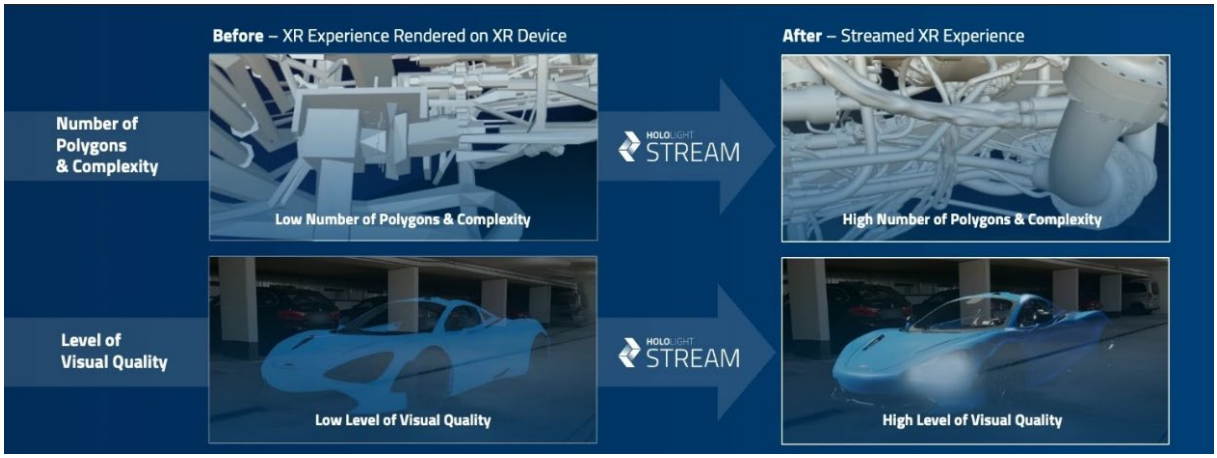


Figure 49: High polygonal 3D CAD data visualization on Hololight Space that is streamed to the AR glasses

The advanced functionalities of streaming technology and seamless remote rendering are critically dependent on a robust and efficient network infrastructure. The state-of-the-art HORSE network infrastructure is designed to enhance these capabilities by empowering Hololight Stream, thereby maximizing the performance and functionality of Hololight Space.

This use case highlights HORSE's ability to address network challenges and elevate XR experiences across industries by addressing the growing demands for low latency, high throughput, and secure connectivity. In the use case, the user/users visualize and interact with high-polygonal 3D CAD models on Hololight Space that is streamed to the AR glasses.

During the data exchange between the AR glass streaming Client Application and the server AR Application, network traffic data will be monitored by the HORSE platform. Specifically, the Smart Monitoring component of the STO will collect this data from the network infrastructure.

Following, pre-processing of the data will ensue before the data is sent toward the DEME, SAN, and EM, which allow for the real contextual detection of threats, the emulation of realistic situations, and provide missing information with which to feed into the sandbox, respectively. Both the SAN and DEME will pass advise to the DTE, which in turn generates an AI-based actions which are provided to the IBI. The IBI generates workflows which can be applied to the HORSE infrastructure, which are passed to the RTR and ePEM respectively for definition and coordination of the actions. Finally, the domain orchestrator connectors ensure that all relevant infrastructure elements are appropriately orchestrated per these defined actions/workflows. The efforts of these modules ensure that the XR technologies in this use case can begin to appropriately leverage the advanced network infrastructures particularly in the context of 6G.

The use case supports 4 different scenarios:

1. Rendering XR in Local Networks: The implementation of remote application rendering within local networks is achieved through the integration of HORSE's advanced 5G/6G infrastructure. This process involves the interaction between a Server Application and a Client Application, interconnected via WebRTC. The HORSE platform's robust connectivity ensures optimal performance for XR rendering in localized environments.

Figure 50 is an example of the information flow which will be present in this scenario, which encompasses remote rendering for one end-user in an XR session. The detailed connections between the modules within the HORSE platform components are defined in previous deliverables. Information which is important to be monitored in the data flow consists of network traffic data which will allow monitoring of network availability and security.

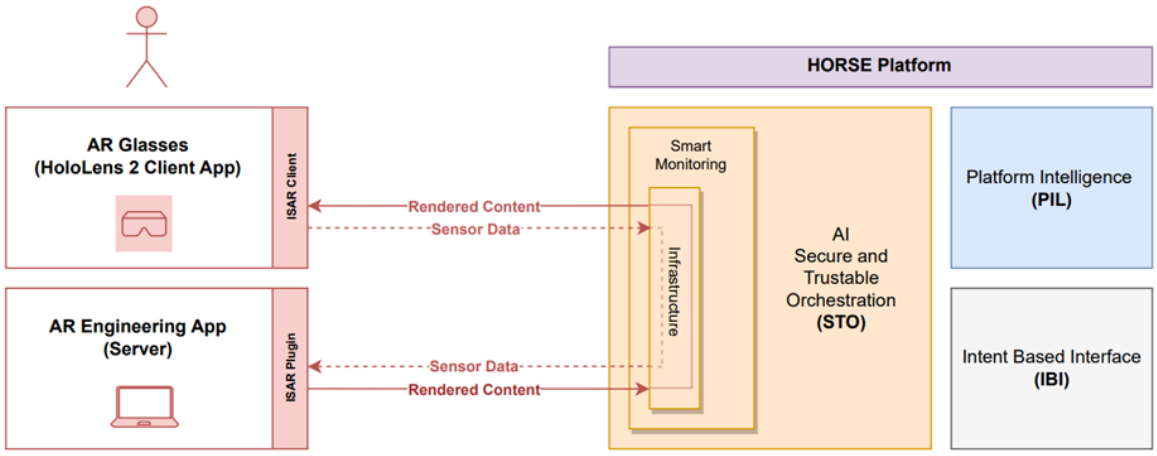


Figure 50: Architecture Mapping and Interaction for a single user in the network

Figure 51 below shows the data flow and the relevant XR technology for the 3D visualization and user interaction.

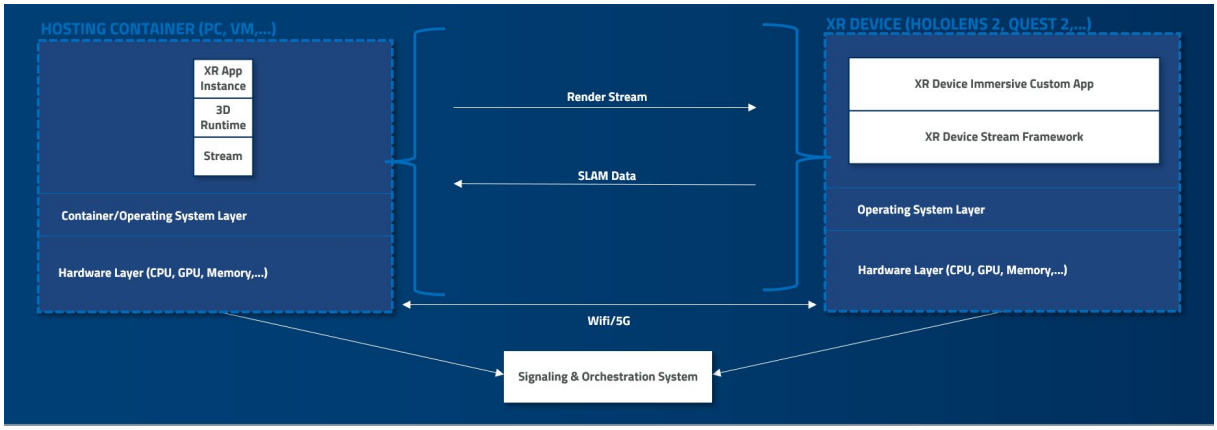


Figure 51: Data flow in streaming technology

2. Fast-Prototyping in Multi-Player Mode: The HORSE platform supports multi-player collaborative sessions, enabling designers and engineers from different locations to work simultaneously on 3D CAD models within Hololight SPACE. By addressing the stringent network demands for such interactive and resource-intensive sessions, the HORSE infrastructure facilitates seamless and efficient collaboration.

3. Industrial Metaverse and XR Devices: In industrial applications, XR technologies enable immersive collaboration for tasks such as rapid prototyping, factory planning, and workforce training. The HORSE platform underpins these scenarios by providing essential network characteristics, including high bandwidth, low latency, and robust end-to-end security, ensuring smooth operation for complex and highly interactive XR environments.

Figure 52: Multiple users with multiple devices collaborating in the same session

4. Multi-User Experiences: HORSE's network architecture enables multiple users to operate independent instances of CAD rendering applications within the same network. This capability allows at least three end-users to simultaneously engage in their sessions within Hololight SPACE, fostering a shared yet individualized XR environment supported by high network reliability and throughput.

Figure 53 below is an example of the information flow in a scenario where multiple users are non-collaboratively working on their own independent CAD files using the same network infrastructure. In this scenario, each user will have their own AR application running (server and client). Correspondingly, each user will have the application stream sent from server to the AR device's client application, and in return receive sensor data. As in other scenarios, the streaming will leverage the network infrastructure of the HORSE platform. Each of these independent instances of the AR application will therefore be profiled by the Smart Monitoring module for the same data metrics as described above for single user sessions. Such monitoring is critical for environments such as this, as simultaneous network resource

demands can have a significant impact on latency and bandwidth, which proportionally negatively influences end-user experiences and XR performance. Monitoring and subsequent orchestration of the network infrastructure enables a method to alleviate such conditions when they occur.

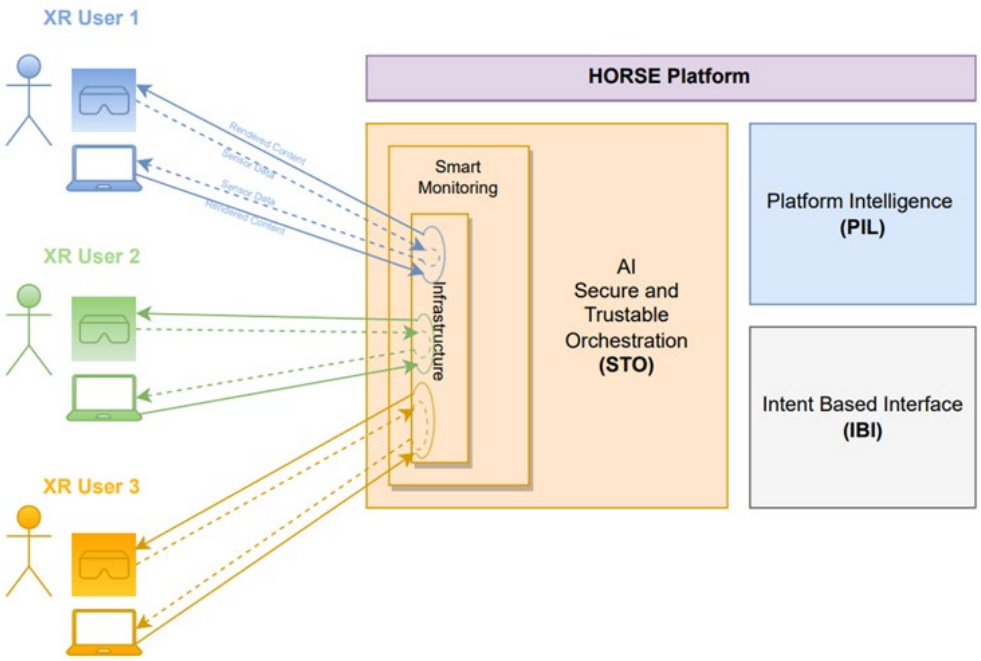


Figure 53: Architecture Mapping and Interaction for multiple users with independent instances in the shared network

The network challenges associated with each use case scenario may differ in scope but nonetheless all benefit from the properties of the HORSE platform. In all scenarios, the relevant modules which supply appropriate orchestration needs and threat detection/mitigation abilities will be utilized, along with the infrastructure itself. In providing such measures, the HORSE platform enables a safe environment and response patterns during potential situations of attacks.

During the data exchange between the HoloLens 2 streaming client application and the server-based AR application, network traffic data is monitored by the HORSE platform. Specifically, the Smart Monitoring component of the STO collects this data from the network infrastructure. Subsequently, the data undergoes pre-processing before being routed to the DEME, SAN, and EM modules. These modules facilitate the contextual detection of threats, emulate realistic scenarios, and provide missing information to populate the sandbox, respectively.

The SAN and DEME modules transmit recommendations to the DTE, which generates AI-driven actions subsequently relayed to the IBI. The IBI produces workflows that are applied to the HORSE infrastructure and further communicated to the RTR and ePEM for action definition and coordination. Finally, domain orchestrator connectors ensure that all relevant infrastructure elements are effectively orchestrated in alignment with the defined actions and workflows. This integrated operation of the modules, see Figure 54, enables XR technologies in this use case to optimally utilize advanced network infrastructures, particularly within the context of 6G.

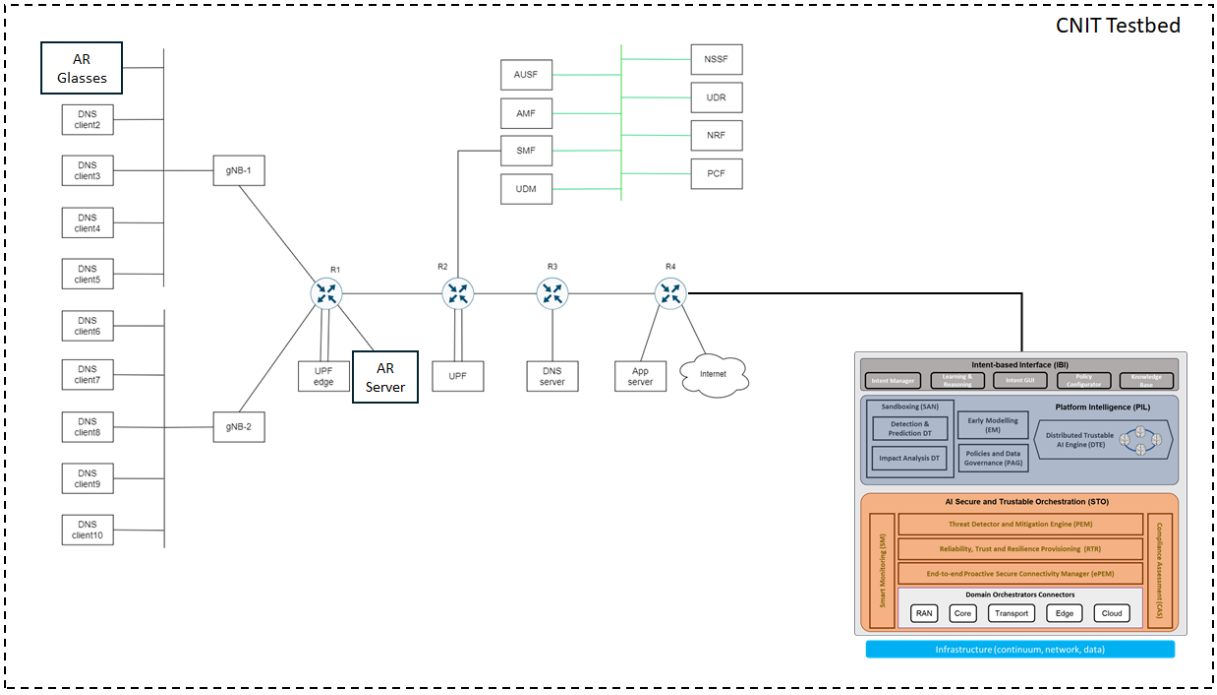


Figure 54: Use case 2 integration in the HORSE framework - CNIT testbed

5.2.1 Data collection

In the scenarios described in previous section, first traffic captures were conducted during the 16 and 17th December 2024 at HOLO. The captures were done using wireshark running on the local server. The traffic contains all packages exchanged between HoloLight Space application running on local server and HoloLens 2. Figure 55 shows relevant traffic related to HoloLight Space, this traffic has been filtered using the command ip.addr == 192.168.101.69.

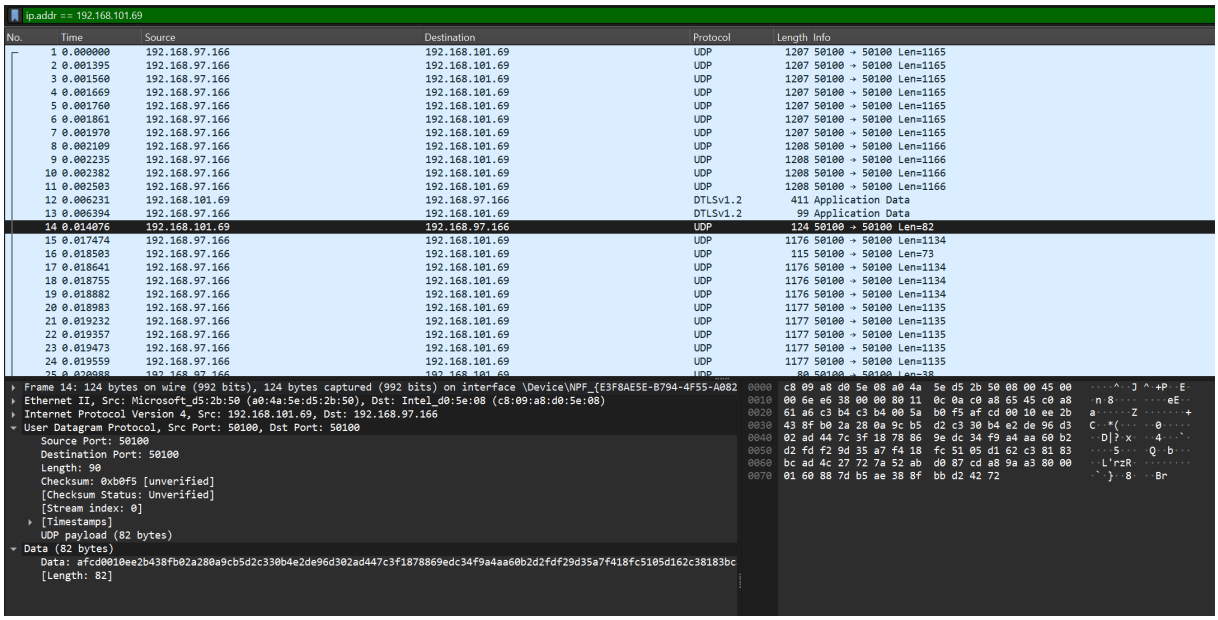


Figure 55: Traffic captures related to Hololight Space

6 Conclusions

This document presents the final specification of the HORSE architectural design, conducted in IT-2, to meet the requirements for an autonomous, self-evolving and extendable 6G-ready architecture. The focus of this architecture is on network automation, while addressing key aspects such as trust, artificial intelligence (AI), and security.

Section 2 revisits the evolution of the HORSE architecture, tracing its development from the reference initial version defined in the project proposal, through the first iteration (IT-1) to the final version in IT-2. This evolution process considers the influence of emerging 6G technologies and applications, as well as the ongoing work of leading standardization bodies in the field.

In Section 3, a comprehensive description of the HORSE architectural design, as developed in IT-2, is provided. This includes valuable insights gained from the practical implementation and integration of HORSE components during IT-1. The section offers an in-depth overview of the architecture's components, detailing their final functionalities. These functionalities will be implemented in the technical work packages (WPs), specifically WP3 and WP4, and integrated in WP5.

Section 4 introduces two canonical workflows, which demonstrate how the architecture's components are involved in threat detection and prediction processes. These workflows outline the entire process, illustrating the interaction between the different modules of the architecture.

Additionally, this deliverable updates the proposed scenarios for the two key use cases of the project: Secure Smart LRT Systems (SS-LRT) and Remote Rendering to Power Extended Reality in Industrial (R^2XRI). The latter focuses on the role of extended reality (XR) in industrial applications. The document also outlines how the HORSE platform will be integrated into both use cases, including the process for data collection.

This deliverable represents the final outcome of the HORSE project, aligning with the latest advancements in 6G, cybersecurity, and AI. The final version of the HORSE architecture will guide the development tasks for the HORSE components, driving the implementation of an autonomous, dynamic and extendable 6G-ready platform.

7 References

[1] HORSE Project (2023). Deliverable D2.2 “HORSE Architectural Design (IT-1).

[2] HORSE Project (2024) D2.3 HORSE Landscape: Technologies, state of the art, AI policies and requirements (IT-2).

[3] "ETSI 2024 work programme on Cybersecurity," 2024. [Online]. Available: https://www.etsi.org/technologies/cyber-security. [Accessed 6 12 2024].

[4] “ETSI Cybersecurity Technical Committee,” [Online]. Available: https://www.etsi.org/committee/1393-cyber.

[5] S. AlDaajeh and S. Alrabaee, “Strategic cybersecurity,” *Computers & Security,* vol. 141, p. 103845, 2024.

[6] P. Kumar, “ Large language models (LLMs): survey, technical frameworks, and future challenges,” *Artificial Intelligence Review,* vol. 57, no. 10, p. 260, 2024.

[7] Y. Cao, H. Zhao, Y. Cheng, T. Shu, Y. Chen and G. Liu, “Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods,” *IEEE Transactions on Neural Networks and Learning Systems,* 2024.

[8] M. A. Ferrag; et al., “Generative AI and Large Language Models for Cyber Security: All Insights You Need,” *arXiv preprint arXiv:2405.12750,* 2024.

[9] R. Fang; et al., “Llm agents can autonomously exploit one-day vulnerabilities,” *arXiv preprint arXiv:2404.08144,* 2024.

[10] . N. Nahar; et al., “A Survey on Zero Trust Architecture: Applications and Challenges of 6G Networks,” *IEEE Access,* 2024.

[11] V. Ziegler; et al. , “Security and Trust in the 6G Era,” *IEEE Access,* vol. 9, pp. 142314-142327, 2021.

[12] S. I. Loutfi; et al. , “An overview of mobility awareness with mobile edge computing over 6G network: Challenges and future research directions,” *Results in Engineering,* p. 102601, 2024.

[13] “Elasticsearch,” [Online]. Available: https://www.elastic.co/elasticsearch.

[14] “Beat,” [Online]. Available: https://www.elastic.co/beats/.

[15] C. Zhou; et al., “Digital Twin Network: Concepts and Reference Architecture,” *Internet Engineering Task Force.*

[16] "Comnetsemu," [Online]. Available: https://github.com/stevelorenz/comnetsemu.

[17] Z. Xiang, S. Pandi, J. Cabrera, F. Granelli, P. Seeling and F. Fitzek, "An open source testbed for virtualized communication networks," *IEEE Communications Magazine,* vol. 59, no. 2, pp. 77-83, 2021.

[18] S. Tariq, E. Rodriguez, X. Masip-Bruin, P. Trakadas, A. Jukan and D. López, "Strategy for Modeling Threats in 5G and B5G Networks," in *5th Workshop on Secure IoT, Edge and Cloud systems (SIoTEC) 2024, CCGRID'2024, The 24th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing*, Philadelphia, 2024.

[19] T. M. Moerland; et al., "Model-based reinforcement learning: A survey," *Foundations and Trends Machine Learning,* vol. 16, no. 1, pp. 1-118, 2024.

[20] D. P. Möller, et al., "Cyberattacker Profiles, Cyberattack Models and Scenarios, and Cybersecurity Ontology," *Guide to Cybersecurity in Digital Transformation: Trends, Methods, Technologies, Applications and Best Practices,* pp. 181-229, 2023.

[21] "NIST SP 800-55 Rev. 2. Performance Measurement Guide for Information Security," [Online]. Available: https://doi.org/10.6028/NIST.SP.800-55r2.iwd.

[22] 3GPP, "TR 23.700-80, Study on 5G System Support for AI/ML-based Services", V18.0.0 (2022-12)," [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=4009.

[23] ENISA, "5G Cybersecurity Standards," [Online]. Available: https://www.enisa.europa.eu/publications/5g-cybersecurity-standards.

[24] "ETSI, "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point"," [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/005/03.05.01_60/gs_nfv-sol005v030501p.pdf.

[25] T. Project, D3.2 Final evaluation of Life-cycle automation and high performance SDN components, 2022.

[26] "Kubernetes, Custom Resources CRD," [Online]. Available: https://kubernetes.io/docs/concepts/extend-kubernetes/api-extension/custom-resources/.

[27] C. N. C. Foundation, "Open Cluster Management (OCM)," [Online]. Available: https://open-cluster-management.io/.

