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

D4.1 HORSE AI-assisted human-centric Secure and Trustable Orchestration developed (IT-1)

Revision: v.1.0

Work package	WP 4
Task	Task 4.1, Task 4.2, Task 4.3, Task 4.4
Due date	31/01/2024
Submission date	31/01/2024
Deliverable lead	STS
Version	1.0
Authors	Fabrizio Granelli (CNIT), Rodrigo Diaz (ATOS), Sofia Giannakidou (STS), Georgios Spanoudakis (STS), Konstantina Koloutsou (STS), Manuel Angel Jimenez (ATOS), Daniel Ruiz (ATOS), Alexandros Dimos (8BELLS), Charalampos Skianis (8BELLS), Ramin Rabbani (CNIT), Alessandro Carrega (CNIT)
Reviewers	Paulo Paixão (EFACEC ES), Orazio Toscano (Ericsson)

Abstract	The deliverable titled "HORSE AI-assisted Human-centric Secure and Trustable Orchestration (IT-1)" is a fundamental component of the broader project, serving as both a technical blueprint and a progress report for Work Package 4 (WP4). It plays a critical role in offering a comprehensive understanding of the project's objectives and the specific tasks undertaken in WP4. This document meticulously outlines the development of essential components at various maturity levels. The deliverable is instrumental in detailing the architecture and functionality of these components, acknowledging their varying maturity levels. By documenting these technical advancements and their potential for integration, the deliverable ensures transparency, accountability, and traceability of efforts. These efforts are directed towards creating a secure, trustable, and resilient orchestration platform that benefits both human operators and AI systems within the HORSE project.
Keywords	Orchestration, HORSE modules, Cross-Domain Trust Mechanisms

DOCUMENT REVISION HISTORY

Version	Date	Description of change	List of contributor(s)
V0.1	14/09/2023	1st version of the template for comments	Sofia Giannakidou (STS)
V0.2	28/09/2023	1st version of the Table of Contents	Sofia Giannakidou (STS), Georgios Spanoudakis (STS), Konstantina Koloutsou (STS)
V0.3	03/10/2023	Adjustments in the Table of Contents	Sofia Giannakidou (STS), Georgios Spanoudakis (STS), Konstantina Koloutsou (STS)
V0.4	13/10/2023	Contribution to the deliverable	Rodrigo Diaz (ATOS)
V0.5	03/01/2024	First round of contributions	Sofia Giannakidou (STS), Georgios Spanoudakis (STS), Konstantina Koloutsou (STS), Manuel Angel Jimenez (Atos), Daniel Ruiz (Atos), Alexandros Dimos (8BELLS), Charalampos Skianis (8BELLS), Ramin Rabbani (CNIT), Alessandro Carrega (CNIT)
V0.6	09/01/2024	Second round of contributions	Sofia Giannakidou (STS), Georgios Spanoudakis (STS), Konstantina Koloutsou (STS), Manuel Angel Jimenez (Atos), Daniel Ruiz (Atos), Alexandros Dimos (8BELLS), Charalampos Skianis (8BELLS), Ramin Rabbani (CNIT), Alessandro Carrega (CNIT)
V0.7	10/01/2024	Ready for project internal review	Sofia Giannakidou (STS)
V0.71	16/01/2024	Revision from Ericsson	Orazio Toscano (Ericsson)
V0.72	16/01/2024	Revision from EFACEC	Paulo Paixão (EFACEC ES)
V0.8	24/01/2024	Addressed comments received from	Sofia Giannakidou (STS), Georgios

		internal reviewers (Ericsson, EFACEC)	Spanoudakis (STS), Konstantina Koloutsou (STS), Manuel Angel Jimenez (Atos), Daniel Ruiz (Atos), Alexandros Dimos (8BELLS), Charalampos Skianis (8BELLS), Ramin Rabbani (CNIT), Alessandro Carrega (CNIT)
V0.81	29/01/2024	Version for QA	Sofia Giannakidou (STS)
V1.0	31/01/2024	Quality assessment and final version to be submitted	Fabrizio Granelli (CNIT)

Disclaimer

Copyright notice

Project co-funded by the European Commission in the Horizon Europe Programme		
Nature of the deliverable:	OTHER	
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

The document titled "HORSE AI-assisted Human-centric Secure and Trustable Orchestration (IT-1)" is a vital deliverable within the broader HORSE project, serving as both a technical blueprint and a progress report for Work Package 4 (WP4). This deliverable plays a pivotal role in comprehensively detailing the project's objectives, the specific tasks undertaken in WP4, and the development of critical components at varying levels of maturity, such as the Smart Monitoring (SM) Component, Pre-processing Component, Reliability Trust, and Resilience (RTR) Provisioning Framework, End-to-End Proactive Secure Connectivity Manager (ePEM), Compliance Assessment Procedures, and Domain Orchestrator Connectors (DOC).

The SM component, essential for lifecycle management in the 6G services environment, initially focused on leveraging EVEREST for cybersecurity and data protection. However, a strategic pivot towards data collection was made to align with the HORSE framework's analytical and processing orientation. This shift included adopting Elastic Beats[1] for efficient data forwarding to Elasticsearch, highlighting scalability and real-time data processing capabilities.

The Pre-Processing module, another integral part of the HORSE architecture, acts as an intermediary between raw data collection and subsequent analytical processes. It ensures a harmonised and standardised data landscape, crucial for the system's overall functionality.

The RTR Provisioning Framework enhances the security and reliability of the HORSE infrastructure. It offers a holistic understanding of the framework's design, functionalities, and operational considerations, contributing to a secure, reliable, and resilient orchestration platform.

The ePEM plays a central role in HORSE's security infrastructure, providing observability and orchestration over various components. It includes a sophisticated data collection approach facilitated by the Topology Manager, which integrates Prometheus servers. This system's adaptability and responsiveness in data collection are tailored to the unique characteristics and evolving needs of the network.

The DOC are designed to provide a unified resource management and orchestration layer across various domains within the HORSE architecture. This integration is critical for achieving seamless operation and effective resource allocation across the platform.

This document, while outlining the development of these critical components, acknowledges the differing maturity levels of each, such as the Compliance Assessment Procedures, which are slated for further development in the project's second iteration. By providing detailed insights into these advancements and their integration potential, this deliverable ensures transparency, accountability, and traceability of efforts aimed at creating a secure, trustable, and resilient orchestration platform. It acts as a foundational guide for ongoing and future development within the HORSE project, setting the stage for further updates and refinements in the subsequent iterations.

[1] https://www.elastic.co/beats

Table of contents

List of figures

Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
CAS	Compliance Assessment
CACAO	Collaborative Automated Course of Action Operations
CRUD	Create, Read, Update and Delete
DFF	Data Fusion Framework
DNS	Domain Name System
DOC	Domain Orchestrator Connectors
ePEM	End-to-End Proactive Secure Connectivity Manager
EVEREST	Event Reasoning Toolkit
GDPR	General Data Protection Regulation
GUI	Graphical User Interface
HTTP/HTTPS	Hypertext Transfer Protocol/Hypertext Transfer Protocol Secure
IBI	Intent-Based Interface
IDPS	Intrusion Detection and Prevention System
IP	Internet Protocol
JSON	JavaScript Object Notation
KDU	Kubernetes Deployment Unit
KPI	Key Performance Indicator
LCM	Lifecycle Management
MQTT	Message Queuing Telemetry Transport
NF	Network Function
NFV	Network Function Virtualisation
NFVO	Network Function Virtualisation Orchestrator

NFVCL	Network Function Virtualisation Convergence Layer
NSI	Network Service Instances
OCM	Open Cluster Management
ORAN	Open Radio Access Network
OSM	Open Source MANO
PNF	Parallel NFs
RAN	Radio Access Network
RBAC	Role-Based Access Control
REDIS	Remote Dictionary Server
REST	Representational State Transfer
RTR	Reliability Trust and Resilience
SDN	Software Defined Networking
SM	Smart Monitoring
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSL/TLS	Secure Sockets Layer/Transport Layer Security
VIM	Virtualised Infrastructure Manager
VDU	Virtual Deployment Unit
VNFM	Virtual Network Function Manager
VNF-FG	Virtual Network Function Forwarding Graph
WP	Work Package

1 About this Document

1.1 Role of the Deliverable

The deliverable, titled "HORSE AI-assisted Human-centric Secure and Trustable Orchestration (IT-1)," plays a pivotal role within the broader project by serving as a technical blueprint and progress report for WP4. Its primary role is to provide a comprehensive understanding of the project's objectives and the specific tasks undertaken in WP4. This document outlines the development of critical components, whose interfaces can be seen in Figure 1, each at varying levels of maturity, such as:

- Smart Monitoring Component
- Pre-processing Component
- Reliability, Trust, and Resilience Provisioning Framework
- End-to-End Proactive Secure Connectivity Manager
- Compliance Assessment Procedures
- Domain Orchestrator Connectors

The deliverable plays a crucial role in detailing the architecture and functionality of these components. Importantly, it acknowledges the differing maturity levels of each component, such as the Compliance Assessment Procedures, which are slated for development in the project's second iteration. By documenting these technical advancements and their integration potential, the deliverable ensures transparency, accountability, and traceability of efforts aimed at creating a secure, trustable, and resilient orchestration platform that benefits both humans and AI systems within the HORSE project.

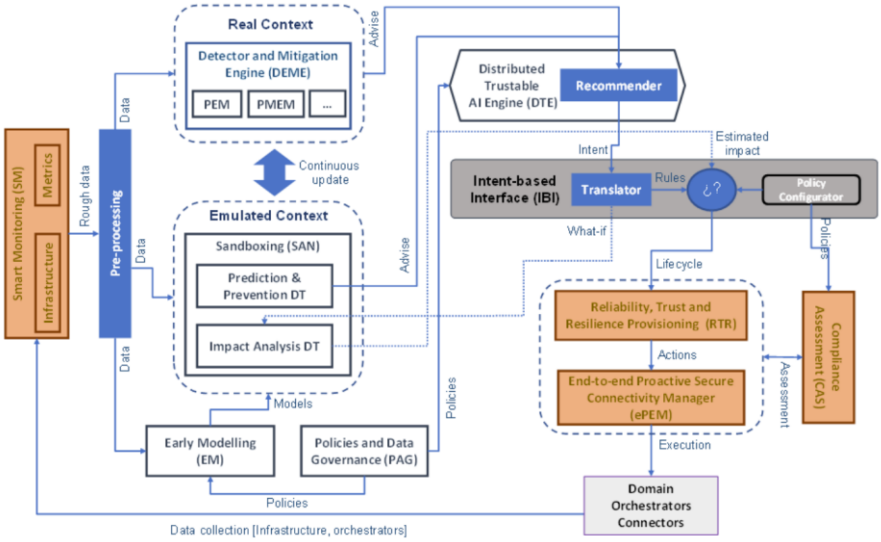

Figure 1: HORSE Components Interfaces

1.2 Relationship to other HORSE deliverables

D4.1 is focused on developing specific modules for the HORSE platform, particularly those related to AI-assisted orchestration that are human-centric, secure, and trustable and is related to the Deliverables stated bellow:

- Deliverable D2.2 – HORSE Architectural Design (IT-1)

- Deliverable D3.1 – HORSE Platform Intelligence developed (IT-1)

D2.2 provides the baseline architectural design that guides subsequent development tasks in the project. It establishes the framework within which other deliverables, D4.1 builds specific, secure, and trustable modules based on this architecture, and D3.1 expands the platform's intelligence by adding components that complement and interact with those developed in D4.1. Each deliverable is interconnected, contributing to a cohesive development process aimed at creating a robust, AI-assisted human-centric platform.

1.3 Structure of the Document

Chapter 2 addresses the Smart Monitoring component. This section offers a thorough examination of the component, beginning with its conceptual underpinnings and extending to its technical and operational specifics. It also discusses the development stages, security protocols, methods of integration, and provides instructions for efficient operation and maintenance.

Chapter 3 presents the Pre-processing component, beginning with a comprehensive overview that highlights its significance and role. This is followed by an in-depth exploration of the framework's development process and the methodologies employed in its creation.

Chapter 4 introduces the Reliability, Trust, and Resilience Provisioning Framework. It starts with an overview of the framework, outlining its importance and function. The chapter then delves into the development process and methodologies used in creating the framework, followed by a discussion on security considerations and data collection methods. The integration of the framework within the project, along with its technical aspects such as APIs, access control, and permissions, are also examined. The chapter concludes with guidelines on deployment, operation, and maintenance, offering practical insights for implementation.

Chapter 5 focuses on the End-to-End Proactive Secure Connectivity Manager component, mirroring the structure of the previous chapter. It provides a comprehensive look at the component, from its conceptual foundation to the technical and operational details. The chapter covers the development process, security measures, integration methods, and guidelines for effective operation and maintenance.

Chapter 6 is dedicated to the Domain Orchestrator Connectors (DOC). This chapter traverses the conceptual underpinnings, development details, and security aspects of the connectors. It also discusses their integration into the wider project and provides guidelines for their deployment and maintenance

Chapter 7 deals with the Continuous Compliance Assessment component, covering its conceptual overview.

The final chapter synthesises and discusses the broader implications of the project. It highlights the key achievements and breakthroughs, examining how these integrate with and support the broader objectives of the project. This chapter is essential in tying together the various components discussed in the previous chapters, offering a cohesive understanding of the project's overall impact and significance.

2 Smart Monitoring Procedures

The evolution of 6G services demands a robust and efficient lifecycle management infrastructure. Integral to this infrastructure are the services offered by the SM component, which has undergone significant strategic development. This chapter provides an overview of the updated role and integration of the SM component within the 6G services lifecycle management framework.

The initial approach detailed in Deliverable 2.2 [1] leveraged the capabilities of EVEREST, a state-of-the-art monitoring tool renowned for its cybersecurity measures and data protection features. The primary role of EVEREST was to safeguard our systems against cyber threats and ensure compliance with stringent data protection regulations. Its design focused on continuous monitoring and real-time alerts to pre-empt potential security breaches. However, a comprehensive evaluation and the planned integration with the HORSE framework prompted a strategic pivot. The HORSE ecosystem, known for its robust analytical and data processing tools, necessitated a re-evaluation of the SM component's role. Consequently, it was decided to recalibrate the focus from a monitoring-centric approach to one that emphasised data collection.

Under this new directive, the SM component's primary function is streamlined to data collection only. This shift aligns seamlessly with the HORSE framework's operational model, which is centered around analysis and processing rather than direct monitoring. To accommodate this refocused role, Elastic Beats [2] will be adopted for the development of the SM. Elastic Beats, characterised by its lightweight yet powerful data shipping capabilities, enables the SM component to efficiently forward data to Elasticsearch. This platform not only ensures scalability and flexibility but also meets our stringent requirements for real-time data processing and analysis. This strategic adjustment in the SM component's role is designed to streamline operations within the 6G services lifecycle management. By aligning more closely with the HORSE framework, the aim is to foster a more cohesive and efficient ecosystem for managing the lifecycle of 6G services.

While this new strategy means foregoing the advanced monitoring capabilities of EVEREST, the analytical prowess of the HORSE framework, in conjunction with the targeted data collection facilitated by Elastic Beats, will create a secure and robust environment for the 6G operations.

2.1 Overview and Development Details

To facilitate efficient data collection in the SM component, Elastic Beats will be utilised as a key tool in the lifecycle management of the HORSE Project. Elastic Beats is an essential component in the data collection process for 6G services lifecycle management. Beats will play a pivotal role in gathering and transferring data to a centralised analysis platform, such as Elasticsearch. Below is a detailed description of how Elastic Beats functions, with specific examples like network traffic data collection from the underlying infrastructure.

Elastic Beats is a suite of lightweight, single-purpose data shippers. Each 'Beat' is designed to collect specific types of data from different sources. These Beats can efficiently forward this data to Elasticsearch or Logstash for further processing and analysis.

Depending on the type of data required, different Beats are deployed. For example, 'Packetbeat' is used for monitoring network traffic, 'Metricbeat' for system metrics, 'Filebeat' for log files, etc. Beats are deployed on the servers or devices from where the data needs to be collected. They are configured to specify what data to collect and how frequently. Beats capture real-time data. For instance, Packetbeat intercepts and analyses network traffic directly from

the infrastructure, capturing details about the traffic flow, request-response time, etc. An expanded view on the utility of Elastic Beats in the context of SM entails:

1. Real-Time Monitoring and Quick Response, which is achieved through immediate data availability. With Beats like Packetbeat, data from network traffic is collected in real-time. This immediacy allows network operators to quickly identify and respond to issues such as traffic congestion, network failures, or cyber threats. Furthermore, system metrics collected by Metricbeat can predict potential system failures or performance bottlenecks. This enables proactive maintenance, reducing downtime and improving service reliability, which is particularly important for the development of the SM component of HORSE.

2. Enhanced Security through the detection of anomalies and threats. Packetbeat's ability to monitor network traffic in real-time plays a vital role in identifying unusual patterns that could indicate cyber-attacks. Early detection of such threats is crucial in preventing data breaches or service disruptions. Additionally, Filebeat can collect logs that are essential for auditing purposes. This helps in maintaining compliance with various regulatory requirements, especially those concerning data security and privacy in the telecom sector.
3. Scalability and Flexibility. Elastic Beats can be easily deployed across various parts of the 6G infrastructure, regardless of the underlying hardware or software environment. This adaptability makes it a versatile tool for data collection. It is noteworthy that more Beats can be deployed without significant changes to the existing setup, ensuring that the growing data demands are met efficiently.

4. Data-Driven Decision Making. The comprehensive data collected and analysed provides valuable insights into network performance and user behaviour. This data-driven approach aids in making informed decisions for network upgrades, capacity planning, and service optimisations. Moreover, understanding traffic patterns and system performance helps in improving the overall user experience. This is particularly important in the envisioned deployment and networks, where customer expectations are high regarding speed and reliability.

In summary, Elastic Beats stands out as a comprehensive and efficient solution for data collection in the 6G services lifecycle, offering substantial benefits in terms of real-time monitoring, enhanced security, scalability, and cost-efficiency, and thus for the SM component. By deploying specific Beats such as Packetbeat, Metricbeat, and Filebeat, it facilitates the collection of a wide range of data in real-time from various infrastructure components. This data, once stored in Elasticsearch, becomes a powerful resource for analysis, ensuring optimal performance and security of 6G services. Its capabilities in monitoring, security, scalability, and aiding data-driven decisions render Elastic Beats an invaluable tool for network operators. Leveraging this technology will enable SM to support robust performance, enhanced security, and superior service quality in the complex and demanding environment of modern telecommunications infrastructure.

2.2 Security and Data Collection

In the context of the SM, usage of Elastic Beats can enhance security in several ways throughout the data collection process:

- Encrypted Data Transmission: Elastic Beats can be configured to encrypt data during transmission. This means that as data is collected and sent to Elasticsearch or Logstash, it is protected from interception or tampering by unauthorised entities. This is particularly crucial for sensitive data traversing potentially vulnerable network paths.
- Secure Authentication and Authorisation: Elastic Beats supports integration with secure authentication mechanisms. It can work with platforms that enforce access controls,

ensuring that only authorised Beats can send data to Elasticsearch. This prevents unauthorised data injection, which could lead to data corruption or security breaches.

- Limited Data Access: Beats are designed to collect only specific types of data, which can be finely tuned according to need. This minimises the risk of sensitive data being inadvertently collected and stored, thus reducing the potential impact of a data breach.
- Audit Trails: Filebeat, one of the Beats, can be used to collect and forward logs to a centralised location. These logs can include access logs, application logs, and system events, which are crucial for auditing and detecting suspicious activities. The availability of comprehensive logs aids in forensic analysis in the event of a security incident.
- Regular Updates and Security Patches: Elastic, the company behind Elastic Beats, regularly updates its software to address security vulnerabilities and enhance functionality. Keeping Beats updated ensures that any known security flaws are promptly addressed, thereby maintaining a strong security posture.
- Minimal Footprint on Host Systems: As lightweight data shippers, Beats have a minimal footprint on the systems where they are deployed. This reduces the risk of them being exploited as a vector for cyber attacks compared to more resource-intensive monitoring solutions.

The SM will heavily leverage the above-mentioned attributes to ensure the security of the data collection processes in targeted deployments. Through encrypted data transmission, secure authentication, limited data access, comprehensive audit trails, anomaly detection capabilities, regular updates, and a minimal system footprint, it ensures that the data collection is not only efficient but also secure against various cyber threats.

2.3 Integration and Interfaces

The integration of Elastic Beats interfaces into servers and devices for efficient data collection through SM represents a significant advancement in telecommunications infrastructure management. In terms of deployment of Elastic Beats on servers and devices, the SM will benefit from:

- Customisation and Configuration. Elastic Beats are designed to be easily deployable on a variety of servers and devices. Each Beat is tailored to specific data collection needs, allowing for customisation based on the type of data required. For example, Metricbeat for system metrics, Filebeat for log files, and Packetbeat for network traffic.
- Simplified Installation Process. The installation of Beats is streamlined, involving minimal steps. Once installed, they require only minimal resources.

2.3.1 APIs and Format Exposed Through Interfaces

Provision of usable and readily parse-able data is of utmost importance to STS's SM. Integration of Elastic Beats interfaces, coupled with the appropriately exposed APIs shall consider:

- User-Friendly Interfaces: To that end, Elastic Beats are equipped with user-friendly interfaces that simplify configuration and management. These interfaces allow for easy setup of data collection parameters and customisation of data sources. This will benefit SM, as it will be allowed to integrate convoluted data structures via higher abstraction layers.
- API Integration: Beats offer robust API support, facilitating integration with a wide range of systems and applications. These APIs are essential for automating data collection tasks

and ensuring seamless data flow between Beats and other components of the 6G network infrastructure.

- Configuration Flexibility: Through APIs and configuration files, Beats can be finely tuned to collect specific data types, manage data flow, and control data shipping intervals. This flexibility is vital in adapting to the diverse data requirements and scenarios that the SM will be tasked to accommodate.

When it comes to specific data formats (and handling thereof), the SM may consider:

- Support for Multiple Data Formats: Regarding this, Elastic Beats can handle various data formats, making them versatile in different deployment scenarios. Whether it's structured data like (JavaScript Object Notation) JSON, unstructured logs, or network packets, Beats can efficiently process, parse and forward this data.
- Data Normalisation and Enrichment: Before forwarding data to the appropriate data processing pipelines employed by SM (e.g., Elasticsearch, Logstash), Beats can normalise and enrich the data. This pre-processing enhances data quality and consistency, making it ready for analysis and also subsequent visualisations.

The integration of the above-mentioned technologies and interfaces into network components (servers and devices) is a pivotal development in data collection and network management. The flexibility, ease of deployment, robust API support, and secure data handling capabilities of Elastic Beats make them an ideal solution for implementing the services over which the HORSE monitoring solution will be overlayed.

2.3.2 Access Control and Permissions

Deploying the Elastic Beats substrate within the HORSE 6G network infrastructure involves navigating various access controls and permissions to ensure secure and efficient data collection. The success of this deployment hinges on appropriately managing these controls to safeguard network integrity and comply with regulatory standards.

Firstly, in terms of user and administrative permissions, the below-listed points are considered:

- Administrative Access: To install Elastic Beats on servers or devices, administrative or root-level permissions are typically required. This level of access is necessary to ensure that Beats can be properly installed, configured, and given access to the necessary system resources and data streams. Thus, the entirety of the SM solution will require root-level permissions in order to deploy all underlying components and deliver its services.
- User Access Management: Proper user access management is, of course, crucial. This involves creating specific user roles for managing individual Beats and assigning permissions based on the principle of least privilege. Such an approach minimises potential security risks by restricting access only to those who need it for their specific job functions.

Secondly, regarding network permissions and firewalls SM shall consider:

- Network Access: Elastic Beats need network access to send collected data to Elasticsearch or Logstash. This requires configuring network permissions and possibly updating firewall rules to allow outbound connections from Beats to the data aggregation tools. Access management of individual (sub)processes and Beats, as well as privilege provisioning can be orchestrated by SM to ensure proper alignment with the task and requirements at hand.
- Secure Communication Channels: It’s essential to establish secure communication channels, especially in environments where Beats are transmitting sensitive data. This involves setting up Secure Sockets Layer/Transport Layer Security (SSL/TLS) encryption for data in transit to prevent interception or tampering.

Thirdly, considering data source permissions, SM will address access control in the following ways:

- Access to Data Sources: SM (and thus its underlying Beats) must have permission to access the data they are meant to collect. For instance, Filebeat needs read access to log files, Metricbeat requires access to system metrics, and Packetbeat needs permissions to capture network traffic.
- Selective Data Access: It's important to configure Beats to collect only the necessary data, respecting privacy and compliance requirements. This involves setting permissions and filters to ensure that Beats do not access or transmit sensitive data unintentionally. Similarly to network access handling, SM will implement selective data access management on a need-to-know basis, providing access in a discriminatory manner, ensuring data security.

Fourthly, in regard to compliance and regulatory considerations, SM will ensure alignment with the proper authorities and regulations and shall consider:

- Compliance with Data Protection Laws: When deploying Beats in regions with strict data protection laws, like General Data Protection Regulation (GDPR) in the European Union, it's crucial to ensure that data collection and handling practices comply with these regulations. This might involve the SM configuring individual Beats to anonymise data, or even to avoid collecting certain types of data altogether.
- Audit Trails and Logging: Maintaining audit trails of who accessed the Beats, when, and for what purpose is important for compliance. Elastic Beats should be configured to log such access, which is essential for security audits and compliance reporting.

Fifthly and lastly, throughout the SM's development process, the consortium shall consider integration with existing security infrastructure, which will be done through:

- Alignment with Security Policies: Deployment of Elastic Beats should align with the organisation's existing security policies. This includes integrating with existing identity management systems, adhering to network security protocols, and following organisational best practices for software deployment.
- Regular Updates and Patch Management: Ensuring that Elastic Beats are regularly updated and patched is crucial for security. This requires permissions to update software, which should be part of the ongoing maintenance plan.

Deploying Elastic Beats within the HORSE Project needs to take into consideration the access controls and permissions. From ensuring administrative access for installation to managing network and data source permissions, and complying with regulatory standards, each aspect plays a vital role in the secure and effective deployment of Beats. By meticulously managing these permissions, organisations can leverage the full potential of Elastic Beats for efficient data collection while maintaining a robust security posture.

3 Pre-Processing Module

3.1 Overview and Development Details

The implementation of the Pre-processing module within the HORSE architecture involves a meticulous consideration of technologies and methodologies to seamlessly integrate with the data input from the SM component. This module serves as a critical intermediary between raw data collection and subsequent analytical processes, ensuring a harmonised and standardised data landscape.

Data Input Integration:

- The primary data source for the Pre-processing module is the SM component, which collects diverse data from infrastructure components, domain orchestrators, and various metrics related to resource usage. Leveraging Elastic Beats, the SM component efficiently forwards obtained data to Elasticsearch, ensuring a continuous and scalable flow of information. The integration with Elastic Beats establishes a robust connection, allowing the Pre-processing module to consume real-time data streams seamlessly.

Data Fusion Framework (DFF) Platform Integration:

In order to implement and support the operation inherent to the Pre-Processing Module, DFF will be integrated to the module. DFF is a product developed by 8 Bells that fosters functionalities on data source management, secure data flow, system data exchange, efficient data access and retrieval as well as system interfacing. In summary, the main purpose of the DFF is centred on efficient and reliable data sharing by making a diverse set of data sources easily accessible to the interested recipients. Two distinct methods of integration are employed:

- GUI Interaction:
  - A web application, accessible at https://dff.8bellsresearch.com/, provides a graphical user interface (GUI) for users to sign in, upload data schemes in JSON format, and subscribe to data feeds.
  - Utilising the GUI, users can efficiently upload data with accompanying short descriptions, fostering a user-friendly experience.
- API Endpoint Connectivity:
  - The Pre-processing module leverages DFF's API endpoints for direct integration.
  - Specific HTTP requests facilitate functionalities such as user login, POSTing data to DFF, and GETting data from DFF.
  - This method ensures a programmatic and efficient interaction with the DFF platform, enhancing automation and system integration capabilities.

System Flexibility and Scalability:

- The Pre-processing module orchestrates the aggregation of data from disparate origins and subsequently distributes them to parties interested in acquiring those data. Through its integration with DFF and Elastic Beats, it accommodates structurally varied datasets without compromising scalability. The utilisation of DFF's API endpoints aligns with the modular design of the HORSE architecture, providing flexibility for future enhancements and adapting to evolving data processing requirements.

In summary, the development of the Pre-processing module meticulously incorporates technologies such as Elastic Beats and the DFF platform, optimising the integration with Smart

Monitoring data and ensuring a seamless, standardised, and scalable data preprocessing pipeline within the HORSE architecture.

3.2 Security and Data Collection

Elastic Beats enhances security in several ways throughout the data collection process in a 6G network environment: Ensuring the security and integrity of data is paramount within the Pre-processing module of the HORSE architecture, especially given its pivotal role in harmonising and standardising information sourced from the SM component. Rigorous measures are implemented to safeguard data during collection, transmission, and storage, adhering to industry standards and best practices.

Secure Data Transmission:

The integration with the SM component employs secure communication protocols, leveraging HTTPS to encrypt data in transit. Elastic Beats, serving as the intermediary between SM and the Pre-processing module, adheres to industry-standard security protocols, ensuring the confidentiality and integrity of the transmitted data streams. This encryption protocol mitigates the risk of unauthorised access and interception during the data transmission process.

DFF Platform Security:

Interaction with the DFF platform via both GUI and API endpoints is fortified with robust security measures:

- GUI Security:
  - User authentication through Keyrock [3] enhances security, allowing only authorised users to sign up and access the system.
  - SSL/TLS encryption secures data transmitted through the GUI, preventing eavesdropping and unauthorised access.
- API Endpoint Security:
  - Authentication mechanisms, such as token-based systems, are implemented for API endpoint interactions.
  - Access controls restrict unauthorised access to sensitive functionalities like data upload, ensuring that only authenticated and authorised entities interact with the Pre-processing module via the DFF API.

Data Upload Validation:

To guarantee the integrity of the data uploaded to the Pre-processing module, comprehensive validation mechanisms are employed:

- Schema Verification:
  - Uploaded data schemes in JSON format undergo thorough schema validation, ensuring adherence to predefined data structure standards.
  - Any inconsistencies or deviations from the expected schema trigger alerts, preventing the ingestion of malformed or malicious data.
- Historical Data Auditing:
  - The "History Upload" feature in the main console provides a chronological record of uploaded files, detailing timestamps and user-provided descriptions.
  - This auditing capability enhances traceability and accountability, allowing administrators to review historical data uploads for potential anomalies or security breaches.

Compliance with Data Protection Regulations:

- The Pre-processing module operates in accordance with stringent data protection regulations. Adherence to standards such as GDPR ensures that user data is handled responsibly, promoting transparency and user trust. Data anonymisation and minimisation principles are applied, mitigating the risk of privacy breaches and ensuring compliance with evolving regulatory frameworks.

In conclusion, the Security and Data Collection aspects of the Pre-processing module underscore a commitment to robust cybersecurity practices, safeguarding data integrity from collection through harmonisation within the HORSE architecture.

3.3 Integration and Interfaces

The seamless integration of the Pre-processing module within the HORSE architecture relies on well-defined interfaces and robust access control mechanisms. This section outlines two crucial aspects: APIs and Formats Exposed Through Interfaces, and Access Control and Permissions.

3.3.1 APIs and Format Exposed Through Interfaces

The Pre-processing module interfaces with the DFF through APIs, facilitating efficient data harmonisation and standardisation. Two distinct API methods are employed:

- Data Ingestion API:
  - Utilising specific HTTP requests, data is seamlessly transmitted to DFF for processing.
  - The API supports JSON format for data ingestion, ensuring compatibility and flexibility in handling diverse datasets.
- Data Retrieval API:
  - The Pre-processing module can retrieve harmonised data from DFF through secure HTTP requests.
  - The API response adheres to a structured format, allowing the module to easily incorporate processed data into subsequent analytical processes.

3.3.2 Access Control and Permissions

DFF API Security:

Access to DFF APIs is meticulously controlled to prevent unauthorised interactions and ensure data integrity:

- Authentication Mechanisms:
  - User authentication mechanisms, such as token-based systems, validate the identity of entities interacting with the APIs.
  - Each API request requires a valid authentication token, minimising the risk of unauthorised access.
- Role-Based Access Control (RBAC):
  - RBAC is implemented to assign specific roles and permissions to users interacting with the Pre-processing module.
  - Different roles dictate the level of access, preventing unauthorised modification or extraction of sensitive data.

GUI Access Controls:

For users interacting with the Pre-processing module through the GUI, access controls are implemented to govern functionality:

- User Registration and Authentication:
  - Users register and authenticate through Keyrock, ensuring that only valid users access the system.
  - SSL/TLS encryption secures user credentials during the authentication process.
- Role-Based Permissions:
  - Different roles, such as administrators and regular users, are established with varying levels of permissions.
  - This ensures that users have access only to functionalities and data relevant to their roles.

Data Format Standardisation:

Data standardisation is a pivotal aspect of the Pre-processing module's interface:

- Input Data Format (Smart Monitoring):
  - Elastic Beats forwards data from the SM component in a standardised format.
  - This standardised format ensures consistency and facilitates seamless integration into the Pre-processing module.
- Output Data Format (DFF Integration):
  - Harmonised data retrieved from DFF adheres to a consistent structure, simplifying its utilisation in subsequent analysis.
  - The JSON format serves as a versatile and interoperable standard for data exchange.

In summary, the Integration and Interfaces section emphasises the strategic use of APIs, data formats, and stringent access controls to ensure a cohesive and secure interaction between the Pre-processing module and both the Smart Monitoring component and the Data Fusion Framework.

4 Reliability, Trust, and Resilience Provisioning Framework

This chapter delves into the core aspects of the Reliability, Trust, and Resilience Provisioning Framework developed under the purview of WP4. It provides a comprehensive view of the efforts undertaken by the 8-BELLS team to enhance the security and reliability of the HORSE infrastructure.

4.1 Overview

The chapter concludes with an overview of the complete Reliability, Trust, and Resilience Provisioning Framework. It outlines the framework's key components, principles, and how it contributes to the overall goal of ensuring a secure, reliable, and resilient orchestration platform within the HORSE project. The comprehensive exploration of development details, security and data collection, integration and interfaces, deployment, operation, and maintenance guidelines provide a holistic understanding of the framework's design, functionalities, and operational considerations. This section serves as a bridge between the detailed technical aspects discussed in the preceding sections and the overarching goals and impact of the Reliability, Trust, and Resilience Provisioning Framework within the broader context of the HORSE project.

4.2 Development Details

The RTR Provisioning module within the HORSE architecture embodies a sophisticated integration of cutting-edge technologies and meticulous development methodologies to fortify the platform's secure performance. This section delves into the foundational aspects of the RTR module's development, highlighting its core components, methodologies, and technologies leveraged.

- Technology Stack:

The RTR module is architected using a robust technology stack comprising industry-leading tools and frameworks. Key technologies include Python for its versatile scripting capabilities, ensuring seamless integration with various components. Additionally, the module exploits the power of Django, a high-level web framework, to facilitate rapid development and maintainability.

- Input Processing from the Intent-Based Interface (IBI):

The primary input to the RTR module originates from the IBI, encapsulating high-level security intents in a standardised JSON format. Leveraging RESTful API endpoints, the RTR module dynamically ingests these intents, interpreting parameters such as threat type, mitigation actions, and associated time frames.

The JSON below illustrates IBI's output and describes a mitigation action for "Network Denial of Service: Reflection Amplification". This example is provided by TUBS:

```
{
   "subnet_address": "192.168.0.1/32",
   "mitigation_action": "filter_network_traffic",
   "threat": {
     "type": "denial_of_service",
     "service": "dns",
     "port": ["53"]
```

```
  },
  time_frame: {
    "start": "2024-01-30T18:25:43.000Z",
    "end": "2024-01-30T18:30:00.000Z"
  }
}
```

More specifically, regarding the JSON's fields:

- The subnet address indicates the host that was targeted by an attack. The subnet address could indicate an entire subnet of just a specific host, and what defines if it is a subnet of a host is the IP Mask.
- The "mitigation action" is the high-level intent describing the mitigation action for a certain type of threat
- The "threat" provides insight on the type of threat, service and port can be omitted.
- "time frame" refers to the duration of the mitigation action, if omitted the action should be enforced continuously.

- Mitigation Action Generation:

Upon receiving intents from IBI, the RTR module orchestrates the generation of mitigation actions tailored to the specified threats. Advanced algorithms and decision-making processes analyse the incoming intents, mapping them to predefined mitigation strategies. This ensures a swift and precise response to potential security incidents.

- Unique Mitigation Action Identification:

Each generated mitigation action is assigned a unique identifier, allowing for unambiguous reference and streamlined management. This identification mechanism facilitates subsequent interactions, such as modification or deletion of specific mitigation actions. It adheres to RESTful principles, accommodating HTTP verbs like POST and DELETE for seamless integration with the broader HORSE architecture.

- Output to ePEM:

A key facet of the RTR Provisioning module is the generation of Ansible Security Playbooks to convey mitigation actions to the ePEM. This type of playbook focuses on providing a simple and powerful automation framework, which relies on YAML configuration files. Ansible allows for the automation of various tasks, making it easier to manage and configure systems in a scalable and consistent manner. This way RTR offers a standardised and structured format for machine-executable security playbooks.

- Example of Ansible Playbook Structure:

Ansible Playbooks [4] do share some standard components among each other, but each playbook is tailor-made to fulfil a series of actions specific to the requested task. In this section, we provide an open-source playbook by RedHat. The playbook annexed in this deliverable is tasked with configuring the local syslog service to collect logs generated by the open-source version of Snort.

Ansible Playbooks comprise of various modules, some of the most common are:

- Name: describes the name of the playbook based on the operations it performs.
- Hosts: Specifies the name of the host the playbook will be executed.

- Tasks: A unit of work or action that should be performed on the target hosts.
- Handlers: Tasks that are only executed by other tasks.
- Modules: The building blocks of tasks, representing the actual units of work performed by Ansible.
- Service: This module enables one to control the state of a service on a remote host.
- Roles: A way to organise and reuse tasks, variables, and other Ansible components in a structured manner.
- Variables: Can be defined at various levels (play, host, group, or role) and are used to parametrise the playbook.
- Conditionals: Ansible allows the use of conditionals to control the flow of execution based on certain criteria.

Ansible relies on SSH for remote communication, and users typically employ key-based authentication for secure communication between the control node and the managed nodes.

In adherence to the Ansible standards, this structured output facilitates both human-understandability and machine-executability, ensuring seamless integration and execution within the ePEM module of the HORSE architecture.

- Scalability and Performance:

The development of the RTR module prioritises scalability and performance. Employing containerisation technologies such as Docker ensures efficient resource utilisation, allowing the module to dynamically adapt to varying workloads while maintaining optimal performance levels.

- Logging and Auditing:

Robust logging and auditing mechanisms are integral to the RTR module, enabling comprehensive tracking of all processed intents and executed mitigation actions. This not only facilitates real-time monitoring but also contributes to post-incident analysis and continuous improvement of the platform's security posture.

In summary, the development of the RTR module embodies a commitment to excellence, combining state-of-the-art technologies, meticulous input processing from IBI, unique identification of mitigation actions, seamless integration with ePEM, and a focus on scalability and performance for a resilient and trustworthy HORSE platform.

4.3 Security and Data Collection

Security and data collection constitute foundational pillars in the design and operation of the RTR Provisioning module within the HORSE architecture. This section elucidates the meticulous approach undertaken to ensure the robustness of data collection mechanisms, adherence to security protocols, and the responsible handling of sensitive information.

- Secure Data Transmission:

The RTR module prioritises the security of data transmission between interconnected components. Employing industry-standard encryption protocols, such as TLS, ensures the confidentiality and integrity of data exchanged between the RTR module, the IBI, and the ePEM. This safeguarding of communication channels mitigates the risk of data interception and tampering.

- Authentication and Authorisation:

Rigorous authentication and authorisation mechanisms are integral to the RTR module's security architecture. Each communication endpoint, including interactions with IBI and ePEM, undergoes stringent verification processes. The implementation of access control lists and secure token-based authentication ensures that only authorised entities can initiate and respond to data requests. This layered security approach bolsters the overall trustworthiness of the RTR module.

- Sensitive Data Handling:

In adherence to privacy and data protection principles, the RTR module employs secure and ethical practices in handling sensitive information. Personally identifiable information (PII) is minimised, and anonymisation techniques are applied wherever feasible. Data retention policies are established to ensure the responsible management of information, aligning with regulatory frameworks and best practices.

- Dynamic Threat Intelligence Integration:

The RTR module continuously enhances its threat detection capabilities through the integration of dynamic threat intelligence feeds. These feeds, sourced from reputable security databases and organisations, enrich the module's knowledge base. The utilisation of machine learning algorithms allows the RTR module to adapt and identify emerging threats in real-time, contributing to a proactive and adaptive security stance.

- Logging and Auditing:

To facilitate forensic analysis and ensure transparency, the RTR module maintains detailed logs of all interactions and processed data. These logs include information on received intents, executed mitigation actions, and communication events with external components. Regular audits of these logs not only aid in identifying potential security incidents but also contribute to continuous improvement initiatives.

- Data Collection for Performance Optimisation:

The RTR module intelligently collects performance metrics related to its operations. This data encompasses response times, resource utilisation, and throughput. Leveraging this information, the module can adapt its strategies, ensuring optimal performance in varying operational contexts. This approach aligns with the platform's commitment to scalability and efficiency.

- Compliance with Data Protection Standards:

The RTR module aligns with industry-standard data protection regulations and standards. Compliance with frameworks such as GDPR (General Data Protection Regulation) ensures that data handling practices are ethical, transparent, and provide users with control over their information. Regular assessments and updates are conducted to align with evolving regulatory landscapes.

4.4 Integration and Interfaces

The seamless integration of the RTR Provisioning module into the broader HORSE architecture relies on well-defined interfaces and robust integration mechanisms. This section details two critical aspects: APIs and the format exposed through interfaces, and access control and permissions.

4.4.1 APIs and Format Exposed Through Interfaces

- RESTful API Design:

The RTR module embraces a RESTful architecture to facilitate smooth communication with external components, specifically the IBI for input and the ePEM for output. The RESTful APIs adhere to industry best practices, ensuring simplicity, scalability, and compatibility with diverse systems.

- JSON Data Format:

The data format exposed through the interfaces is standardised using JSON. JSON provides a lightweight and human-readable structure, facilitating efficient data exchange between the RTR module, IBI, and ePEM. The format encapsulates key details such as mitigation actions, threat information, and time frames, as previously defined in the IBI's JSON output specifications. The RTR module interfaces with the ePEM module by providing mitigation actions in the form of Collaborative Automated Course of Action Operations (CACAO) Security Playbooks [5]. The structured JSON format adheres to the CACAO standard, ensuring a clear definition of playbook metadata, workflow, data markings, and other essential components.

- Mitigation Action Identification:

The RTR module generates unique identifiers for each mitigation action, ensuring traceability and enabling subsequent modifications or deletions. These identifiers are included in the JSON data format, promoting a standardised approach to reference and manage mitigation actions.

4.4.2 Access Control and Permissions

- Authentication Mechanisms:

Access to the RTR module is guarded by robust authentication mechanisms. The module employs secure token-based authentication, validating the identity of interacting components. This ensures that only authorised entities, including the IBI for input and the ePEM for output, can initiate and respond to data requests, mitigating the risk of unauthorised access.

- Authorisation Policies:

Authorisation policies are meticulously implemented to control the level of access granted to different entities within the HORSE ecosystem. Access control lists define the specific operations and resources that each entity, including IBI and ePEM, is permitted to execute. Granular permissions contribute to a fine-grained security model, aligning with the principle of least privilege.

- Role-Based Access Control (RBAC):

The RTR module employs a role-based access control model to streamline access permissions. Different roles are assigned to entities based on their responsibilities and functionalities. For example, the IBI may have permissions to submit intents, while the ePEM may have permissions to receive mitigation actions. RBAC enhances manageability and ensures a tailored approach to access control.

- Logging and Auditing of Access:

Comprehensive logging and auditing mechanisms are integral to the access control framework. Every access attempt, whether successful or unsuccessful, is logged for forensic analysis. Regular audits of access logs not only contribute to identifying potential security incidents but also aid in evaluating and refining access control policies.

4.5 Deployment, Operation and Maintenance Guidelines

The successful deployment, operation, and maintenance of the RTR Provisioning module are crucial to ensuring the continuous security and functionality of the HORSE platform. This

section outlines comprehensive guidelines to facilitate the seamless integration and sustained effectiveness of the RTR module throughout its lifecycle.

1. Deployment Guidelines:

- Environment Preparations:

Before deployment, ensure that the target environment aligns with the system requirements specified for the RTR module. Verify the availability of essential dependencies, such as the Python runtime environment and required external libraries.

- Configuration Management:

Implement configuration management practices to streamline deployment processes. Utilise configuration files to specify environment-specific settings, allowing for easy adaptation to diverse deployment scenarios.

- Scalability Considerations:

Anticipate future scalability needs during deployment. Design the deployment architecture to accommodate varying workloads, leveraging containerisation technologies like Docker for efficient resource utilisation and horizontal scaling.

2. Operation Guidelines:

- Real-time Monitoring:

Establish continuous monitoring mechanisms to track the operational performance of the RTR module. Monitor key performance indicators, including response times, resource utilisation, and error rates, to promptly identify and address potential issues.

- Incident Response Procedures:

Develop and document incident response procedures to guide the team in the event of security incidents or operational disruptions. Clearly define roles and responsibilities to facilitate a coordinated and effective response.

- Logging and Auditing:

Leverage detailed logging and auditing features to maintain a comprehensive record of module activities. Regularly review logs for anomalies, unauthorised access attempts, or any irregularities that may indicate potential security threats.

3. Maintenance Guidelines:

- Patch Management:

Implement a robust patch management strategy to keep the RTR module and its dependencies up to date. Regularly apply security patches and updates to mitigate vulnerabilities and ensure the module's resilience against emerging threats.

- Backup and Recovery Procedures:

Establish backup and recovery procedures to safeguard critical data and configurations. Regularly perform backups and test the restoration process to guarantee the availability of essential information in the event of data loss or system failure.

- Version Control:

Employ version control systems to track changes in the RTR module's codebase. Clearly document each version release, including feature enhancements, bug fixes, and security updates. This practice facilitates traceability and rollback capabilities if needed.

4. Continuous Improvement:

- Feedback Mechanisms:

Establish channels for continuous feedback from users and stakeholders. Regularly solicit input on the RTR module's performance, features, and user experience to inform future updates and enhancements.

- Adaptation to Emerging Threats:

Stay vigilant to emerging security threats and adjust the RTR module's configurations and strategies accordingly. Regularly review threat intelligence feeds and update the module to address evolving threat landscapes.

- Training and Knowledge Transfer:

Provide ongoing training to the operations and maintenance team. Foster a culture of knowledge sharing and documentation to ensure a smooth transition of responsibilities and expertise within the team.

5 End-to-End Proactive Secure Connectivity Manager

5.1 Overview

The ePEM plays a pivotal role in the HORSE security infrastructure. HORSE represents a cutting-edge security infrastructure designed to safeguard complex, distributed, and heterogeneous systems. In this intricate environment, the ePEM serves as a central architectural element, orchestrating actions and providing observability over the various components that constitute the end-to-end services secured within the HORSE security perimeter.

5.2 Development Details

ePEM is at the heart of managing end-to-end secure connectivity within the HORSE security perimeter. It acts as the central coordinator, ensuring that all elements and artefacts operate securely and harmoniously. In its role as an orchestrator, ePEM orchestrates actions and maintains observability across the diverse and heterogeneous components that constitute the HORSE ecosystem. It's responsible for harmonising the complex interplay of resources and services to ensure the infrastructure's resilience and security.

ePEM collaborates with Domain Orchestrators and controllers to enhance the efficiency and intelligence of operations. These external entities bring different degrees of automation and intelligence to the management of resources and artifacts throughout their lifecycle. This includes services related to Network Function Virtualisation (NFV) and resources associated with Software-Defined Networking (SDN).

Topology Information Management:

ePEM maintains a comprehensive database of the logical topology of the distributed infrastructure. This includes information at both the wide-area connectivity and Virtualised Infrastructure Manager (VIM) levels. Additionally, it records details about the orchestrators and controllers responsible for governing these components. Each topological entity is annotated with resource constraints and access levels, which are essential for efficient resource management and access control.

Management of NFV/Applicative Services:

ePEM actively participates in the management of NFV and applicative services. It maintains awareness keeps track of the localisation and degrees of freedom granted by VIMs to Virtual Network Functions (VNFs) and application components within the HORSE security perimeter. It continuously updates information related to NFV/applicative services based on the exposure levels provided by domain orchestrators and controllers.

Data Homogenisation and Simplification:

An aspect of ePEM's role is to homogenise and consolidate data from diverse sources after the pre-processing module. This simplification provides a unified, coherent view of the services managed by the HORSE platform. This streamlined view significantly aids in decision-making and security management.

Meta-Actions for Security:

ePEM autonomously acquires and exposes a repertoire of action types that can be applied to each artefact or group of artifacts within the end-to-end services. These meta-actions assist in

the formulation of contingency plans for security threats and vulnerabilities. They are derived from a collection of predesigned Blueprint profiles that encapsulate the functional behaviour of diverse network elements.

Blueprint Profiles:

Blueprint profiles within the context of ePEM encompass a diverse range of complex network elements, including 5G/6G radio mobile networks, distributed firewalls, monitoring overlay systems, and other intricate elements. These profiles serve as comprehensive templates that outline specific actions and primitives essential for orchestrating activities throughout different phases of network management. By providing a standardised framework, Blueprint profiles empower ePEM to coordinate and execute operations seamlessly, ensuring efficiency and coherence in the management of various network elements.

Whether orchestrating the configuration of a radio mobile network', deploying a distributed firewall, or the coordination of a monitoring overlay system, Blueprint profiles play a pivotal role in streamlining processes and enhancing the overall operational effectiveness of the HORSE infrastructure. This standardised approach facilitates efficient and consistent management of network elements, fostering a cohesive and streamlined network management environment.

In essence, Blueprint profiles serve as the cornerstone of ePEM's ability to effectively manage a diverse landscape of network elements. Their standardised nature and comprehensive design ensure that network management operations are executed seamlessly, contributing to the overall success of the HORSE infrastructure.

Enhancements to Network Function Virtualisation Orchestrators (NFVO) and VIMs:

Modular Architecture

ePEM has been built over a modular and flexible architecture that can be easily extended to support various Network Functions (xNF) and ecosystems. At the foundations of this architecture, illustrated in Figure 2, lies the metamodel, specifically designed to augment extensibility and flexibility and drive clear interaction patterns among the different internal modules during Lifecycle Management (LCM) operations.

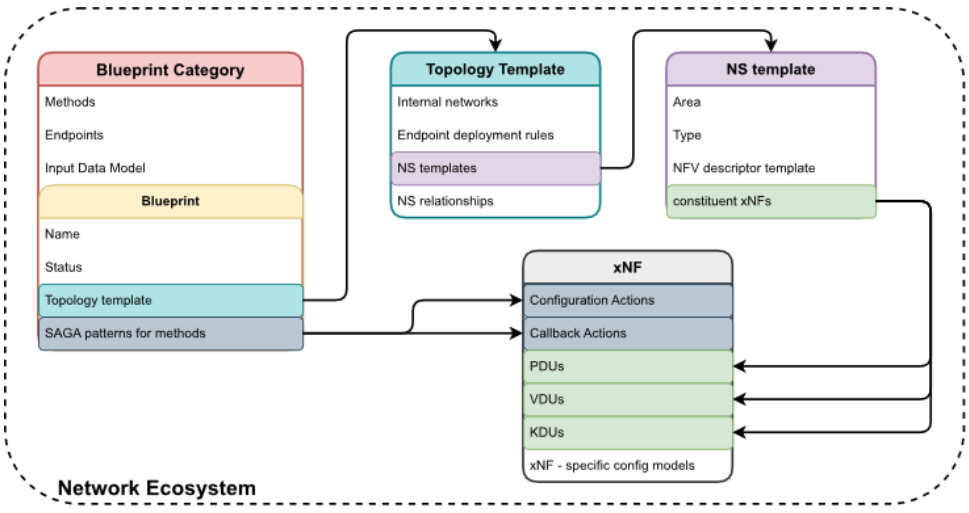

Figure 2: ePEM Modular Architecture

In the ePEM metamodel for Enhanced Extensibility in LCM Operations, every ecosystem instance is built through a Blueprint, which on its turn falls into a Category. The Blueprint Category corresponds to the high-level network ecosystem function type, like a 5G system, a network security tool-chain, etc. The Blueprint is meant to support *ad-hoc* operations for specific implementations falling into that Category. For instance, the ePEM currently provides

4 different 5G system implementations, based on different open-source projects, namely Free5GC, Open5GS, OpenAir Interface and SD-Core.

The Blueprint Category allows to have a homogeneous north-bound interface against the different implementations available for an ecosystem, since it defines a single input meta-data model (including the possible ecosystem end-points) and the associated ecosystem-level LCM methods. For example, the 5G System Blueprint Category exposes operations to add/remove/reconfigure Radio Access Network (RAN) over specific geographical areas, to create/modify/destroy network slices, etc., and it fixes the end-points to be physical devices like base stations or Open RAN (O-RAN) radio units, and networks to be used as 5G DNN.

A Blueprint provides the implementation-specific means to support the Category methods and translate the metadata model into sets of Network Service Instances (NSIs) and xNFs, interconnected and operating with coherent (yet implementation-specific) configurations. To this end, Blueprints defines the template of the ecosystem's internal topology, as well as the specification of the internal procedures to be executed for every supported Category method. As detailed in section named "SAGA pattern", these internal procedures are realised as SAGA pattern interactions among specific ePEM modules.

The topology template defines the graph pattern including the templates of internal network and of NSIs that can be applied, and their possible relationship bindings. On its turn, an NSI template specifies the list of possible constituent xNFs, and the logic to build NFV SOL-006 descriptors to be onboarded and used by the NFV Orchestrator.

Finally, the metadata model of xNFs plays a key role in ePEM's architecture. It defines not only the specific physical/virtual/Kubernetes deployment units to be used to materialise NS templates, but also defines the implementation-specific methods and callbacks that can be executed on an xNF, and the models of its configuration. In other words, xNF templates represent a sort of glue between NFV-driven LCM operations to instantiate or remove artefacts from the ecosystem (e.g., creating a RAN NS in a new area), and management operations affecting the configuration of running xNFs (e.g., add a new 5G subscriber, add a new policy, etc.).

Each of these operations might include a variable number of different actions, including:

- day 0 and 1 actions for adding NSI instances.
- day 2 actionsinvolving moditying the configuration settings of xNFs and retrieving information from the deployed xNFs.
- removal of deployed NSIs.

The Internal Modular Architecture

The ePEM internal architecture, illustrated in Figure 3, encompasses the meta-models introduced in the previous Section. A first module, named NFV Convergence Layer (NFVCL) North Bound Interface aims at exposing CRUD REST APIs for ecosystem LCM trough the methods defined in the Blueprint Category meta-models that are available and onboarded to the ePEM. Among these methods, the ecosystem creation and deletion are mandatory (and correspond to an HTTP POST and DELETE messages).

The NFVCL North Bound Interface module acts as a central hub for orchestrating ecosystem management operations. It leverages the metamodel definitions to translate REST API calls into corresponding ecosystem-specific actions. This facilitates seamless interaction with diverse ecosystem implementations, enabling unified management of network resources.

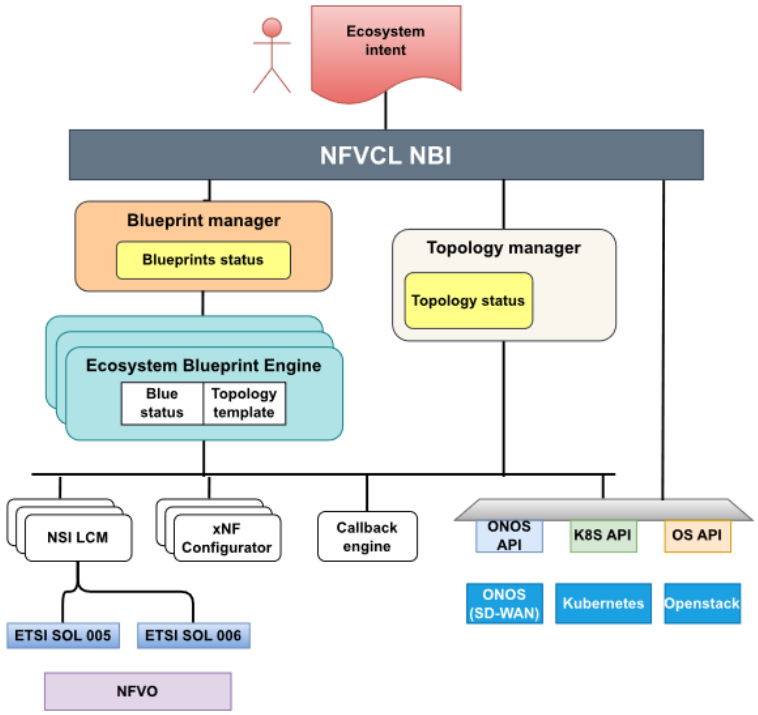

Figure 3: ePEM Internal Architecture and Interfaces

The Topology manager provides essential resources required by the ecosystems. Topology status encompasses details such as the VIM list with their respective network statuses, available Kubernetes clusters suitable for deploying (Kubernetes Deployment Units) KDUs, a comprehensive list of accessible Parallel NFs (PNFs), and metric servers for storing Key Performance Indicators (KPIs). An instrumental capability of the Topology Manager is its ability to terraform resources on the VIM. In simpler terms, when a Blueprint requires something that does not currently exist, the Topology Manager can dynamically create it on demand. Terraform, in this context, refers to the dynamic provisioning and orchestration of resources, allowing for the seamless adaptation and creation of components as needed by the system.

The Blueprint Manager takes care of all the requests towards Blueprints, from creation to dayN operations. An Ecosystem Blueprint Engine instance (Operator) is instantiated for every active ecosystem at its initialisation, in this way, it is possible to work on multiple requests, for different Blueprint instances, at the same time.

The Operator can be thought as a sort of worker, dedicated to handle and to serialise incoming LCM initialisation/change requests on the ecosystem. This component maintains the information related to both the Blueprint meta-model (status) and the Topology template. In particular, it is in charge of binding any supported Blueprint category method into a coordinated set of multiple implementation-specific operation requests against resources in the topology, the LCM of NFV Network Service Instances, or configuration changes within one or multiple xNFs. While these operations are executed into further dedicated components, namely, the Topology Manager, the NSI LCM engines, and the xNF configurators, the Ecosystem Blueprint Operator acts as a central coordinator for the distributed transaction through a publish-subscribe communication system (in the current version, Remote Dictionary Server (REDIS) is applied in this regard). This architecture is known in software engineering as orchestrator-based saga pattern.

ePEM can interact directly with external entities, like Kubernetes or Openstack, to enable features not supported by the VIM. For example, when working with Open Source MANO (OSM), there is no way to create/update images for Virtual Deployment Units (VDUs), they must exist on the VIM. Kubernetes APIs are strongly used for managing and configuring

clusters deployed through ePEM, some useful operations performed in this way are plugins installation and user creation.

Persistency Layer

ePEM utilises a MongoDB database to store critical information essential for its seamless operations. This database serves as a repository for two primary collections that underpin the system's functionality: one dedicated to the topology status and the other capturing the dynamic states and topology templates for each instantiated Blueprint. Upon initialisation, ePEM prioritises loading fundamental topology information, establishing a solid groundwork for subsequent operations. Thereafter, when an LCM operation is initiated for a specific Blueprint, the system efficiently retrieves the corresponding status for that specific Blueprint instance from the database.

Topology status, a crucial facet of the stored information, encapsulates comprehensive details about the overall network structure, providing insights into the current state of interconnected elements and their operational statuses. This holistic overview serves as the bedrock for effective decision-making and orchestration processes within ePEM.

Recognising the diverse needs of each Blueprint ecosystem, the system acknowledges the necessity for varying input data. This acknowledgment underscores the nuanced nature of Blueprint statuses, which are inherently influenced by the distinct requirements of different Blueprint types. Even within the same category, each Blueprint type can exhibit a unique data structure for its status, ensuring that the stored information aligns precisely with the intricacies of its operational context. This tailored approach allows ePEM to accommodate the diverse requirements of different Blueprints, fostering adaptability and flexibility in its operations.

Furthermore, the stored data of a Blueprint extends beyond real-time states, encompassing historical perspectives through the inclusion of past actions, such as previously executed LCM primitives. This historical context provides a valuable repository of insights, allowing the system to trace the evolution of each Blueprint's lifecycle and facilitating comprehensive auditing and analysis.

In summary, the MongoDB database plays a pivotal role in ePEM's operational framework, ensuring that vital information is not only securely stored but also dynamically retrieved to support real-time decision-making and comprehensive historical analysis. This meticulous approach to data management underscores ePEM's commitment to robust, adaptive, and insightful network orchestration within the HORSE architecture.

Key takeaways:

- The MongoDB database stores critical information for ePEM's operation.
- Two primary collections are dedicated to topology status and Blueprint statuses.
- Topology status provides insights into the overall network structure and element states.
- Blueprint statuses vary based on the unique requirements of different Blueprint types.
- Stored data includes historical perspectives for comprehensive auditing and analysis.

Benefits of the MongoDB database:

- Secure and reliable data storage for critical information.
- Dynamic retrieval of information for real-time decision-making.
- Comprehensive historical analysis for tracing Blueprint evolution.
- Support for diverse Blueprint requirements and data structures.

Overall, the MongoDB database plays a crucial role in enabling ePEM to effectively manage and orchestrate diverse network ecosystems within the HORSE architecture.

Relationship with the Open-Source Management and Orchestration (MANO) orchestrator

ePEM is actively engaged in the ongoing evolution of the network operating system within the OSM framework. This system is instrumental in supporting the lifecycle management of network functions and Virtual Network Function Forwarding Graphs (VNF-FGs) embedded in the application's deployment graph. As the network landscape continues to evolve, the efficient management of network functions and their associated forwarding graphs becomes essential. ePEM, therefore, seeks to contribute to and integrate advancements in the network operating system, ensuring seamless lifecycle management and optimal performance of network functions within the deployed applications.

In essence, ePEM's commitment to closely monitoring and actively participating in the advancements within the OSM initiative underscores its dedication to staying at the forefront of NFV orchestration and management. By aligning with the progress in multi-site resource management and network operating systems, ePEM is committed to providing. deliver a state-of-the-art solution that meets the evolving demands of the HORSE infrastructure, providing a reliable, scalable, and adaptive orchestration environment.

5.3 Security and Data Collection

In ensuring the robust security of the HORSE infrastructure, ePEM leverages a comprehensive cybersecurity toolkit, including VyOS [6] and Suricata [7], alongside additional security measures tailored to meet the specific demands of the network.

VyOS assumes a pivotal role within ePEM's security architecture, functioning as a cornerstone that fortifies the overall security infrastructure. As an open-source network operating system renowned for its adaptability, VyOS seamlessly integrates into both standard hardware and virtualised environments, offering ePEM a versatile and scalable solution. The utilisation of VyOS empowers ePEM with a comprehensive set of tools and functionalities, enhancing its capability to establish, manage, and safeguard secure communication paths across the network.

Leveraging the rich feature set of VyOS, ePEM strategically deploys secure communication channels, meticulously enforces network segmentation, and enacts sophisticated firewall policies. This strategic implementation is instrumental in securing the network against a spectrum of potential threats, providing a robust defense against unauthorised access against unauthorised access attempts. VyOS's adaptability and scalability are particularly advantageous for ePEM, allowing it to dynamically respond to the evolving landscape of security challenges.

VyOS not only serves as a protective shield against unauthorised access but also operates as a proactive enforcer of security policies. By leveraging the advanced routing and firewall capabilities of VyOS, ePEM establishes a resilient and structured defence framework. This framework not only safeguards critical assets within the network but also ensures the integrity and confidentiality of data traversing through it.

In essence, VyOS within the ePEM ecosystem is not just a security tool; it is an essential element that contributes to the architecture's robustness and adaptability. Its integration underscores the commitment to maintaining a secure, agile, and scalable network infrastructure, aligning seamlessly with ePEM's overarching goal of achieving end-to-end proactive secure connectivity within the HORSE infrastructure.

Working together with VyOS, Suricata significantly enhances the security posture of ePEM, assuming the pivotal role of an Intrusion Detection and Prevention System (IDPS). Suricata functions as an ever-watchful sentinel within the network, actively and continuously scrutinising

the entirety of network traffic for any potential security incidents or anomalies. This proactive surveillance is fundamental in identifying potential threats before they escalate, contributing to a robust and responsive security infrastructure within the ePEM framework.

Upon detecting a security threat, Suricata can optionally initiate preventive actions, adding an extra layer of defence to the network. This rapid response mechanism can fortify the overall security landscape, minimising the potential impact of security incidents and ensuring the continuity of secure network operations. The option for preventive measures aligns seamlessly with ePEM's mission of maintaining end-to-end secure connectivity within the HORSE infrastructure.

Suricata's strength lies in its sophisticated utilisation of robust signature-based detection methods, allowing it to recognise and thwart known threats effectively. Furthermore, its adaptive nature enables the assimilation of emerging threat intelligence seamlessly. This capability is crucial in the ever-evolving cybersecurity landscape, where new threats continuously emerge. By staying abreast of the latest threat intelligence, Suricata empowers ePEM to not only identify known threats but also proactively detect and respond to novel and evolving malicious activities.

The integration of Suricata equips ePEM with a dynamic and responsive security framework, ensuring that the network infrastructure remains resilient against a wide spectrum of cyber threats and vulnerabilities. Its capabilities go beyond merely reacting to threats; Suricata actively contributes to the prevention and mitigation of potential risks, reinforcing ePEM's commitment to maintaining the sustained integrity and availability of the HORSE infrastructure. In this way, Suricata serves as an indispensable pillar of the network's security arsenal employed by ePEM, aligning with its overarching goal of delivering a secure and reliable end-to-end network environment.

Additional Security Functions: Customised Security Measures

In addition to the security features inherent in ePEM, it incorporates additional security functions tailored to the specific needs of the HORSE infrastructure. These functions may include:

Custom Network Function Deployments:

ePEM showcases a dynamic capability to deploy supplementary network functions with a unwavering emphasis on security, on an as-needed basis. This flexibility is demonstrated by scenarios like deploying a specialised VNF tailored for advanced threat analysis. In this context, ePEM can promptly allocate resources to establish a dedicated enclave for analysing and understanding intricate security threats, contributing to a more resilient and adaptive defence strategy.

Furthermore, ePEM extends its on-demand security capabilities to include the deployment of a secure gateway, a crucial component designed to filter and scrutinise incoming and outgoing traffic. This feature proves invaluable in safeguarding the network against external threats. By proactively deploying these security-centric network functions, ePEM underscores its dedication to proactive threat prevention, maintaining a resilient and adaptable security posture within the HORSE infrastructure.

Internal Security Enhancements:

ePEM demonstrates a remarkable ability to enhance the security of established network functions in response to emerging threats. This adaptive approach allows ePEM to respond swiftly to evolving security requirements. By seamlessly integrating internal security mechanisms, ePEM can fortify the existing security measures of already deployed network functions, maintaining a robust and adaptable security posture. This includes the incorporation of encryption for communication channels, the enforcement of secure application programming

interfaces for inter-component communication, or the integration of runtime threat detection and prevention mechanisms, ePEM's adaptive approach guarantees a customised and responsive security posture. This strategic flexibility effectively addresses current security challenges and prepares the network to effectively counter emerging threats, demonstrating ePEM's dedication to preserving a resilient and secure network infrastructure.

Adaptive Security Policies:

ePEM can be equipped with the capability to implement adaptive security policies that are inherently flexible and can dynamically adjust in response to "up-to-date threat intelligence and dynamic network conditions. This inherent adaptability enables ePEM to proactively counter emerging security threats and vulnerabilities, oystering a highly resilient and secure network environment. Continuous assessment of threat intelligence and real-time network monitoring enable ePEM to stay ahead of potential risks. Consequently, ePEM not only effectively addresses current security challenges but also preemptively detects and mitigates emerging threats swiftly, thereby reinforcing its dedication to upholding an agile and robust security posture within the network infrastructure.

Data Collection

In the realm of data collection from the various Network Functions orchestrated by the ePEM, the system boasts a sophisticated approach facilitated by its Topology Manager. This integral component serves as a central hub, offering a meticulously deployed network of Prometheus servers strategically positioned for efficient data collection purposes. These Prometheus instances stand ready, forming a robust infrastructure for gathering and processing critical metrics emanating from diverse xNFs spread across the network.

The seamless integration of Prometheus servers into the intricate network of interconnected components within ePEM is orchestrated through the Topology Manager, acting as a communication bridge between Blueprints and data collection resources. Blueprints, functioning as comprehensive templates for orchestrating and managing various network elements, utilising the Topology Manager's functionality to forge seamless connections with Prometheus instances. This strategic integration empowers Blueprints to dynamically deploy metric exporters on xNFs, tailoring the data collection process to align precisely with specific operational requirements, environmental conditions, or evolving network requirements.

A remarkable feature of this data collection framework is its adaptability and responsiveness. The nature and volume of collected data are not predetermined but are closely linked to the exporter's configuration. This configuration, injected on-demand, provides granular control over the type and level of detail of data collected from each VNF. This modular and adaptable approach ensures that ePEM can flexibly adjust its data collection strategies, tailoring them to the unique characteristics and evolving needs of different xNFs within the network.

In essence, the ePEM's data collection methodology, orchestrated through the Topology Manager and seamlessly integrated into Blueprints, represents an embodiment of efficiency and adaptability. By establishing a comprehensive framework for data management, ePEM enables the network to enhance its analytical capabilities, fostering an informed, secure, and agile environment within the HORSE architecture. This strategic emphasis on data collection aligns with ePEM's overarching mission to enhance network observability and intelligence, facilitating informed decision-making and proactive responses to dynamic network conditions.

5.4 Integration and Interfaces

5.4.1 APIs and Format Exposed Through Interfaces

ePEM's RESTful APIs serve as a foundational element, providing a user-friendly and standardised interface for interaction within the HORSE infrastructure. The simplicity, scalability, and seamless integration offered by these APIs empower external systems and services to engage with ePEM effortlessly. Tasks such as dynamic resource provisioning, proactive security policy enforcement, and data retrieval become streamlined processes due to the well-defined nature of these RESTful APIs.

In the realm of southbound APIs, ePEM establishes essential connections with a diverse array of network elements and devices. These interfaces play a pivotal role in communicating with both physical and virtual components, such as switches, routers, and virtualised network functions. The result is a cohesive orchestration of resources across the network, ensuring efficient management and utilisation.

Looking upwards, ePEM's northbound interfaces open the gateway to collaboration with higher-level services and orchestrators. This inclusivity extends to coordination with the Open-Source MANO orchestrator, cloud services, and domain orchestrators. Through these northbound interfaces, ePEM is adept at translating high-level service requests into actionable network directives, facilitating seamless integration with a broader ecosystem.

To guarantee interoperability, ePEM supports an array of data formats, including JSON, XML, and YAML. These flexible formats allow ePEM to exchange data and configuration information with external entities in a structured and interoperable manner. This adaptability is particularly valuable in heterogeneous environments where diverse systems and devices coexist.

Ensuring compatibility with various systems and devices is a key consideration in ePEM's design. This is achieved through support for standard communication protocols like HTTP/HTTPS, (Message Queuing Telemetry Transport) MQTT, and Simple Network Management Protocol SNMP. The use of these widely accepted protocols enhances ePEM's ability to communicate seamlessly across different technology stacks, contributing to a cohesive and interconnected network environment.

Security remains at the forefront of ePEM's API architecture. Authentication, authorisation, and encryption mechanisms work in unison to safeguard the APIs. Access to sensitive functionalities and data is strictly controlled, preventing unauthorised access. Moreover, data transmitted through APIs is encrypted, providing robust protection against potential threats such as eavesdropping and tampering. This commitment to security ensures a trustworthy and resilient communication framework within the ePEM ecosystem.

5.4.2 Access Control and Permissions

Role-Based Access Control (RBAC). ePEM implements a robust Role-Based Access Control system to manage permissions and access. Different user roles, such as administrators, operators, and tenants, have granular access privileges. This granular control ensures that only authorised users can perform specific actions, reducing the risk of unintended configuration modifications or unauthorised data retrieval.

Policy-Based Permissions. ePEM's access control is policy-driven. Access rights are defined in accordance with user roles, and these policies dictate what actions users can perform and what data they can access. Policies are customizable and adaptable to align with specific security and operational needs.

Multi-Tenancy Support. In a multi-tenant environment, ePEM provides the capability to manage access control and permissions for different tenant organisations. Tenants have their own isolated spaces within the infrastructure, and access is effectively safeguarding the privacy and security of their resources and data.

Audit Trails and Logging. ePEM logs and audits access and permission changes. This audit trail provides a record of accountability and traceability. In the event of security incidents or policy infringements, audit logs provide a record of who initiated system access, the specific actions performed, and when they occurred.

Dynamic Permissions. Access permissions can be adjusted dynamically in response to evolving circumstances. For example, during security incidents, ePEM can dynamically restrict or grant access to specific users or roles to respond to threats or vulnerabilities effectively.

Identity Management Integration. ePEM interfaces with identity management systems for user authentication and authorisation. This integration ensures seamless alignment of user access with the organisation's established identity and access management policies.

5.5 Deployment, Operation and Maintenance Guidelines

A core feature of the ePEM lies in its process automation, a capability designed to streamline the deployment and management of Blueprints throughout their lifecycle. During instantiation (Day 0-1), the ePEM eliminates the need for user interaction. Instead, a comprehensive Blueprint description and initial configuration settings is all that's required for the seamless deployment of a Blueprint. Notably, this process automation extends beyond initial setup to encompass ongoing configuration adjustments (Day 2) and even Blueprint termination (Day N), providing a seamless and automated lifecycle management experience.

A distinguishing feature of the ePEM is its ability to eliminate the requirement for executing command lists on different xNFs in distinct ecosystems. This is achieved by leveraging the integration of SAGAs within each Blueprint category. SAGAs are instrumental in orchestrating and harmonising intricate processes, guaranteeing that command execution is synchronised, optimising the effectiveness and consistency of Blueprint operations.

To enhance visibility and facilitate real-time responsiveness, LCM operations on the Topology, Blueprints, and more trigger system-wide events, which are promptly communicated to a central REDIS instance. This event-driven architecture ensures that interested parties or system components can subscribe to the event stream and observe real-time operations. This proactive approach enables swift responses to events, fostering agility and responsiveness within the ePEM framework.

The ePEM users can access information through three primary methods. APIs provide direct access to output, making it less suitable for asynchronous tasks. Additionally, users can choose to monitor the direct console output for real-time insights into ongoing operations. For added convenience, the ePEM pioneers a novel approach by allowing users to monitor the console output using the REDIS instance [8]. This unique feature not only improves accessibility but also obviates the need for users to have direct console access to the underlying machines, contributing to a more user-friendly and efficient operational experience within the ePEM environment.

SAGA pattern

As illustrated below in Figure 4, each method in ePEM follows a saga pattern composed of multiple rounds. Each round encompasses an ordered sequence of actions for topology

configuration, initial deployment (Day0-1) and ongoing configuration (DayN). Additionally, optional Day2 actions may be included, The Topology Manager, the NSI LCM, and the xNF Configurator modules, respectively.

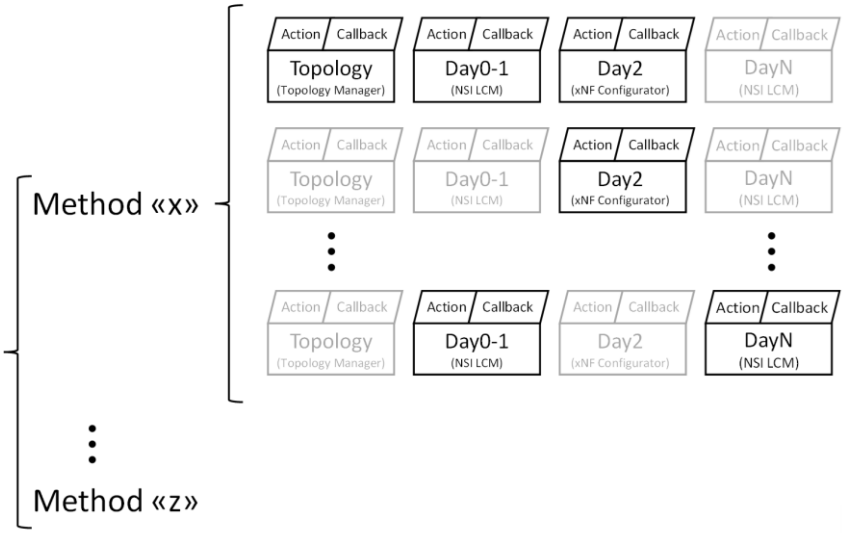

Figure 4: ePEM SAGA Patern

Such actions are represented by execution commands that aim to add/modify/remove entities or to trigger commands in the ecosystem xNFs. Optionally, actions may be followed by callbacks, which are designed to collect information and parameters generated by the action execution. For instance, an action on an xNF acting as a “master” server might produce a key to be used by other “slave” xNFs in the ecosystem. Day2 callbacks are meant to retrieve this key string and move it to the Blueprint Engine, so that it can be used on further actions involving “slave” xNFs.

Day0-1 callbacks are meant to retrieve dynamic deployment information of xNFs (e.g., IP addresses, etc.).

Operation

To better understand how the ePEM is working, below is an actual implementation using Open-Source MANO (OSM) as NFVO. The presence of at least one VIM (Openstack) is mandatory, while the K8s cluster could be deployed using a Blueprint resulting in one or more VDUs. As illustrated in Figure 5, outside the ePEM, the component that actually apply the changes on VNFs is the manager (VNFM) that is located inside a OSM dedicated Kubernetes cluster.

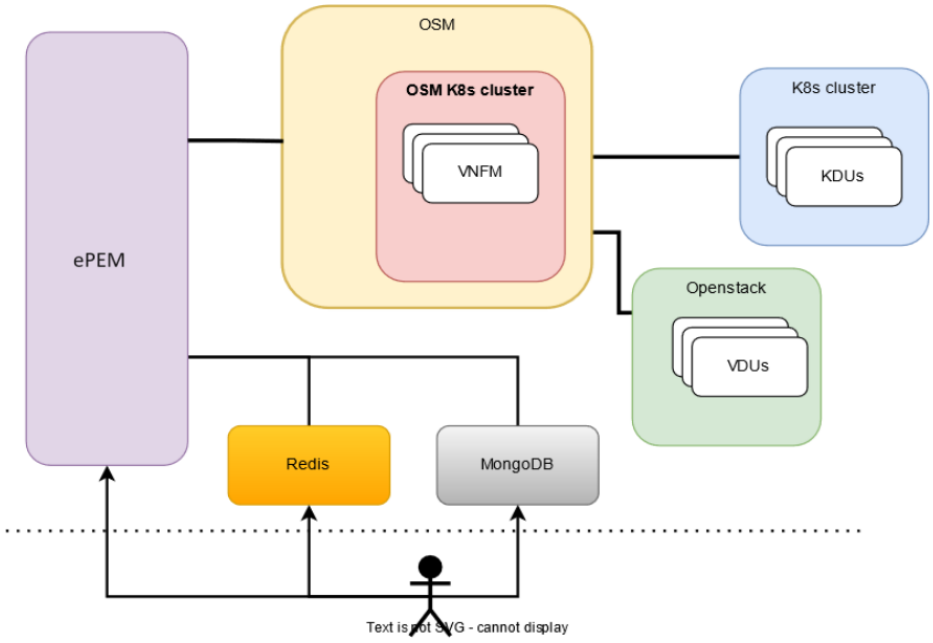

Figure 5: Visualisation of ePEM Implementation

User Interaction

The key feature of the ePEM is the process automation (ZeroOps): when a Blueprint is instantiated (Day0-1), it does not need interaction with the user - a description of the topology template and initial configuration parameters are sufficient for a Blueprint to be deployed. This feature includes also post-initialisation configuration (Day2) and Blueprint deletion (DayN).

In practice, it is possible to avoid the execution of a command list on different xNF in different ecosystems thanks to the SAGAs present inside every Blueprint category.

LCM operations on the Topology, Blueprints, triggers internal events, these are sent to REDIS. Anyone who registers on the event topic can observe what is happening and act accordingly.

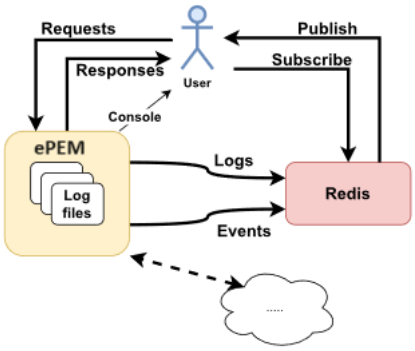

Figure 6: ePEM User Interaction

As shown in Figure 6, the user of the ePEM can retrieve information in different ways:

- Using the APIs output (not working for asynchronous tasks).
- Observe the direct console output (or the one saved in log files).
- Observe the console output after having subscribed to logs on the REDIS instance.
- After having subscribed to one or more Events, receive a notification when it occurs.

6 Domain Orchestrator Connectors

6.1 Overview

These connectors are essential to achieve unified resource management and orchestration across diverse network segments within the HORSE project. These connectors provide an abstraction between the ePEM and the heterogeneous infrastructure, facilitating unified resource management, cross-domain trust and secure licensing, ultimately contributing to project success and efficiency.

6.2 Main Functionalities

Domain Orchestrator Connectors offers a unified interface to the ePEM to execute different actions into the heterogenous and multi-domain infrastructure. It also collects information from the infrastructure nodes to send it to the SM with a particular data form and unified manner.

Unified Resource Stratum

Domain Orchestrator Connectors offers a unified resource stratum through its northbound interface with a single REST API which the behaviour of the different available heterogenous resources can be modified. In this way DOC offer a high level of abstraction between Horse Context and the infrastructure

Cross-Domain Trusting Mechanisms

The main objective of this element is to maintain a single orchestrator which can manage a multicluster environment, to allow a multi-site domain. To implement that Open Cluster Management (OCM) platform should be a great option, its aspects and features are detailed in below sections.

6.3 Integration and Interfaces

This section explains how DOC is integrated into the HORSE Infrastructure and their different interfaces to communicate between them. Figure 7 shows how DOC communicates with different elements of the system:

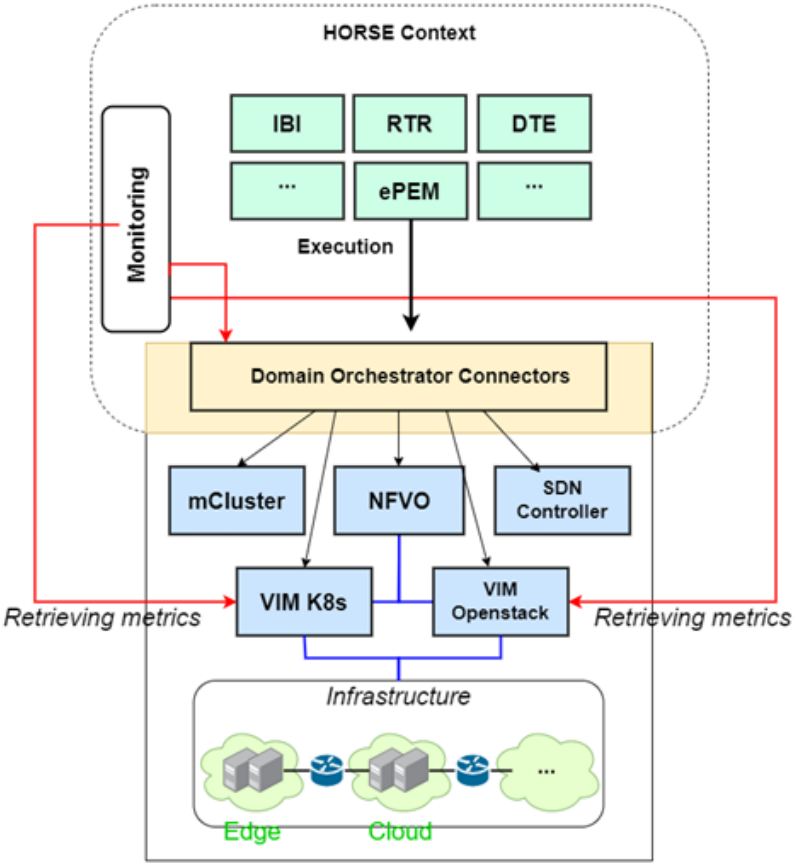

Figure 7: DOC integration and interfaces

APIs and Format Exposed Through Interfaces

- **Northbound APIs.** DOC receive actions to be executed on the infrastructure through Northbound REST API from ePEM. In addition, information about the infrastructure is also sent to the SM.
- **Southbound APIs.** DOC interfaces with various network elements and devices using southbound APIs as shown in figure X. These APIs are essential for communicating with physical and virtual network components.
- **Data Formats.** DOC supports various data formats to ensure interoperability with a wide range of systems and devices.
- **API Security.** Security is a paramount consideration in DOC's APIs, data transmitted through APIs is encrypted to protect against eavesdropping and tampering.

6.3.1 Northbound Interfaces

- DOC receives executions actions from ePEM using REST API design for this purpose, and Figure 8 shows an example of the workflow when action is received from ePEM.

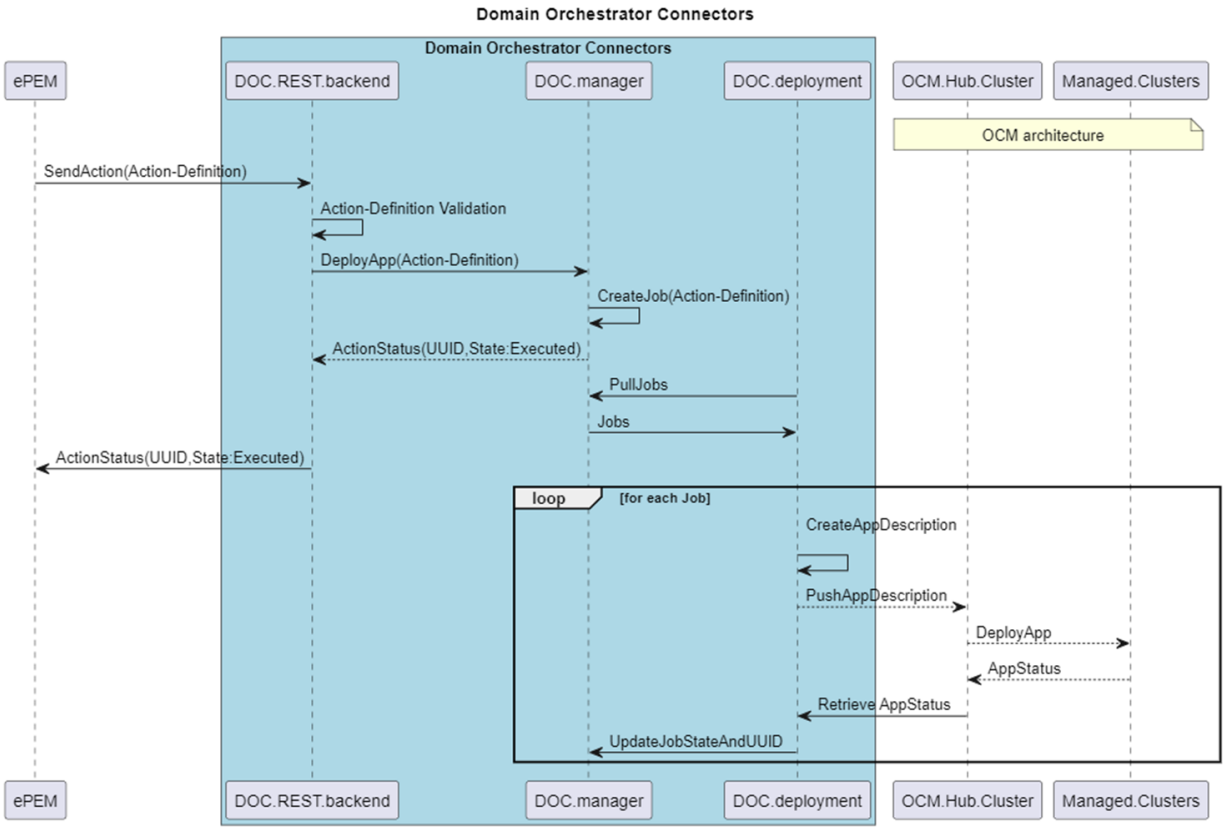

Figure 8: DOC sequence diagram

The actions that DOC received from ePEM are divided into two groups (actions on services and Manage infrastructure):

- Actions on service:
  - Deploy new services
  - Remove existing services
  - Modify existing services
- Manage infrastructure:
  - Add/registry/remove infrastructure
  - Create/remove cluster
  - Update versions
  - Modify nodes (controller, workers...)

After the executions of actions as illustrated in Figure 8, the status of it is sent to the ePEM with the UUID, which is a uniquely identifier assigned by RTR component.

6.3.2 NFV Orchestration

The Open-Source MANO (OSM) stands as a groundbreaking open-source initiative, driven by the ambitious goal of establishing a production-quality NFV orchestrator through the collaborative efforts of the open-source community. Its overarching objective is to create a robust NFV orchestrator using open-source code, intending for it to serve as a benchmark implementation of the MANO stack. As part of its commitment to staying at the forefront of technological advancements, DOC is dedicated to closely monitoring and actively participating in the developments within the OSM ecosystem.

A key focus for DOC lies in establishes communication between HORSE modules and different environments. By aligning with the advancements in the multi-site resource manager, DOC aims to enhance its own capabilities in orchestrating and managing resources within the HORSE infrastructure.

6.3.3 Resources

DOC seek to advance the functionalities of NFVO and cloud-native VIMs. This augmentation is achieved through the facilitation of on-demand injection and removal of VIM level operators and side-car containers within network function/application component pods.

This dynamic management capability serves as a transformative element, allowing for real-time adjustments to the infrastructure based on specific operational requirements.

- By enabling the seamless addition or removal of VIM level operators and side-car containers, DOC provides a versatile solution to enhance observability and security reinforcement.
- This adaptability ensures that the network can dynamically respond to evolving conditions, offering heightened visibility into its operations and the ability to reinforce security measures precisely when and where needed. Whether it involves introducing additional monitoring components for enhanced observability or reinforcing security postures through the addition of specific security measures, in this aspect, DOC works closely with the ePEM's dynamic management capabilities contribute to a more responsive, secure, and efficient network infrastructure.

6.3.4 Distributed Multicluster

A distributed multicluster is the way to achieve an infrastructure in different regions and cloud providers, although it used to cause some complexity when managing. To solve this, OCM will be used to configure the clusters through a single orchestrator.

There are some features from OCM that are essential for the goals of HORSE framework. OCM focuses on manage multi-Kubernetes clusters during full lifecycle, providing a framework to enable any capability within multiple Kubernetes distribution on multiple cloud providers, which is a main goal of HORSE framework to deploy heterogenous infrastructure.

OCM is a modular Software that allow to enhance individual API separate between them, that is an advantage since Infrastructure should be a heterogenous and could need modifications for APIs depending on the deployments, that flexible extensibility framework is a great a definitive point to choose this option.

7 Compliance Assessment Procedures

The Compliance Assessment (CAS) module in HORSE system plays a vital role in aligning security policies and solutions, created by the Trustable AI engine, with applicable regulations. The IBI in HORSE suggests high-level network policies based on specific needs and objectives. CAS then checks these policies for proper alignment with the relevant regulatory guidelines. If a policy is found non-compliant, CAS informs IBI, initiating a feedback loop for further refinement. This interplay between modules enables HORSE to effectively balance operational efficiency with regulatory compliance.

CAS's role extends beyond simple compliance checks. It acts as a gatekeeper, ensuring that the actions planned on the actual infrastructure are in accordance with policies defined by the IBI. Moreover, it serves as a communication bridge to the Policy Configurator sub-module. This is particularly evident in its ability to verify adherence to international standards like 3GPP's security specifications and ENISA standards, thereby mitigating risks such as network attacks and data breaches in 5G contexts [9]. The fundamental principle underlying CAS is poised to be instrumental in advancing regulatory compliance assessments in emerging 6G networks.

Through its collaboration with the Policy Configurator, CAS ensures that policies selected for deployment not only align with intended objectives and requirements but also meet the necessary regulatory standards. This process significantly enhances the overall security posture of the HORSE system.

However, it's important to note that the CAS module, while integral to the envisaged ecosystem, is not as mature as other components in the HORSE platform. Ongoing development and refinement are planned for future iterations. This commitment to continuous improvement aims to maintain and elevate level of quality and effectiveness for which the HORSE platform is aiming, especially as it adapts to evolving security challenges and regulatory landscapes.

8 Discussion

8.1 Achievements of WP4

In the first iteration of WP4, the project team achieved significant advancements in developing the AI-assisted, human-centric Secure and Trustable Orchestration platform. The key achievements were aligned with the four main objectives of WP4 [10]. Firstly, in response to O4.1, the team successfully designed and developed multiple mechanisms to protect the HORSE infrastructure, as part of the Reliability, Trust, and Resilience Provisioning Framework. This included the strategic development of the Smart Monitoring Procedures, which was initially focused on leveraging EVEREST for cybersecurity and data protection, but later shifted towards a data collection-centric approach with the integration of the Elastic Beats platform.

Additionally, addressing O4.2, the team designed and developed the ePEM. This vital component established a central role in the HORSE security infrastructure, ensuring the orchestration and security of distributed and heterogeneous systems within the project's framework. For O4.3, the team focused on monitoring mechanisms and compliance assessments for AI/ML models, integrating these with the broader HORSE architecture. Finally, in pursuit of O4.4, the team developed domain orchestrator connectors, facilitating a unified cross-domain resource management and orchestration stratum, which was essential for the integration task in WP5, driving the comprehensive integration of the HORSE platform. These achievements collectively laid a strong foundation for the HORSE project, establishing a robust and flexible infrastructure capable of addressing complex security and orchestration challenges.

8.2 Integration with WP5

The tools developed in WP4 contribute significantly to the objectives of WP5, particularly in terms of providing a secure, resilient, and flexible architecture. The data collection and processing capabilities of the SM component and the Pre-Processing module align with WP5's goal of developing an analytical platform and implementing a Dashboard. The Reliability Trust and Resilience Provisioning Framework’s focus on security and resilience directly supports WP5’s objective of developing a Continuous Security Assurance solution. ePEM's role in orchestrating secure connectivity and its sophisticated data collection mechanism are crucial for the development of ML pipelines and the envisioned knowledge graph in WP5. The seamless integration and adaptability of these components in the context of WP5 will ensure that the HORSE platform remains secure, efficient, and capable of handling complex, cross-domain resource management, aiding in the development of the Activehealth app and other software resources in WP5.

Conclusions

In conclusion, this deliverable serves as a comprehensive exposition of the developments made in the initial iteration of Work Package 4. It outlines the progression and current state of critical components within the HORSE AI-assisted, human-centric Secure and Trustable Orchestration platform, each of which is at a distinct level of maturity. The achievements detailed in this document underscore the strides made in enhancing the project's infrastructure, focusing on aspects such as security, data collection, and integration of sophisticated monitoring and processing tools. The strategic pivot in the operational focus of various components, like the Smart Monitoring Procedures, has been instrumental in aligning with the overarching goals of the HORSE project.

As the project evolves, further iterations will build upon these foundational achievements. The next iteration is anticipated to bring forward more detailed insights and advancements, furthering the integration and functionality of the platform. This document is instrumental in not only highlighting the current achievements but also setting the stage for the continued evolution of the project, ensuring that future developments are built on a robust and well-documented foundation.

References

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

[2] Elasticsearch, “Filebeat: Lightweight shipper for logs”, https://www.elastic.co/beats/filebeat

[3] Keyrock Open Source Identity Management Tool, https://keyrock-fiware.github.io/

[4] Ansible playbooks, https://docs.ansible.com/

[5] Collaborative Automated Course of Action Operations (CACAO) Security Playbooks, https://docs.oasis-open.org/cacao/security-playbooks/v1.0/security-playbooks-v1.0.html

[6] VyOS – Open source router and firewall platform, https://vyos.io/

[7] Suricata, https://suricata.io/

[8] Redis, https://redis.io/

[9] ENISA, “5G Cybersecurity Standards”, https://www.enisa.europa.eu/publications/5gcybersecurity-standard

[10] HORSE Project, “Holistic, Omnipresent, Resilient Services for Future 6G Wireless and Computing Ecosystems (HORSE) Proposal”, 2022.

Annex

Ansible Security Automation - Snort Integration Playbook

```
- name: enable snort logs
  hosts: localhost

  handlers:
    - name: restart syslog
      service:
        state: restarted
        name: rsyslog

  tasks:
    -name: enable snort log collector
     lineinfile:
       path: /etc/rsyslog.conf
       line: "if $programme == 'snort' then
       state: present
     notify:
       - restart rsyslog
```