System Architecture v2

Deliverable Number	**D2.3**
Deliverable Details: this document describes the final version of Resilmesh Architecture	
Deliverable Leading:	**University of Murcia (UMU)**
Dissemination Level:	**REL**
Due Date:	**31/07/2025**
Submitted Date:	**31/07/2025**
Author(s)	Jorge Bernal, Brian Lee, Jorgeley Inacio de Oliveira, Branka Stojanovic, Martin Husák, Ekam Puri Nieto, Pablo Fernández, Antonio Skarmeta, Vinh La
Reviewer(s):	Martin Husák, Endika Gil-Uriarte

Version History

Version	By	Date	Changes
B	UMU	24/09/2025	Changed SEN to PU
A	UMU, TUS	25/07/2025	Final version
A6	UMU	25/07/2025	Updated version applying comments from internal review
A5	ALIAS, MUNI	24/07/2025	Internal review
A4	ALL	22/07/2025	Document ready for internal review
A3	ALL	21/07/2025	Final contributions from partners
A2	ALL	12/07/2025	Contributions from partners in all sections
A1	UMU	10/06/2025	Table of Contents

Disclaimer

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

Table of Contents

Funded by
the European Union

Table of Figures

Table of Tables

Funded by
the European Union

Abbreviation List

Table 1 - Abbreviations and Acronyms

Abbreviation	Title
AI	Artificial Intelligence
AID	AI-based Detector
APT	Advanced Persistent Threat (APT)
CSA	Critical Service Awareness
CTI	Cyber Threat Intelligence
AIC	AI-Correlation
CASM	Cyber Attack Surface Management
DN	Data Normalization
ESP	Event Stream Processing
EE	Event Enrichment
EA	Event Aggregator
FL	Federated learning
ISIM	Infrastructure and Services Model
IDS	Intrusion detection systems (IDS)
IPS	Intrusion prevention systems (IPS)
MM	Mitigation Manager
ML	Machine Learning
MB	Message Broker
NSE	Network Status Evaluation
PT	PlayBook Tool
PPLF	Privacy-preserving Federated Learning
RM	Resource Orchestrator
RCTI	Robust CTI
SOAPA	Security Orchestration and Analytics Platform Architecture
SOAR	Security Automation, Orchestration and Response
SM	Service Mesh
SIEM	Security Incident and Event Manager
THF	Threat Hunting and Forensics
UEBA	User Entity and Behavioural Analytics
WO	Workflow Orchestrator
XDR	eXtended Detection and Response

Glossary

Table 2 - Glossary

Term	Description
Functional Component	A stand alone element of Resilmesh that implements single logical functionality- A functional component describes the main services that it will feature to implement the function in the framework. Each service defines the capabilities it offers, the type of service, who are the consumers services, pre and post conditions as well as the main envisioned interfaces.
Plane	A collection of functional components of the Resilmesh system that fulfils a specific logical purpose e.g. Threat Awareness Plane. Planes are arranged in a logical hierarchy such that a plane may offer services to the plane above.
Layer	An alternate word for plane - **deprecated**.
Application	A functional component that offers end-user services e.g. Threat Hunting.
Platform	This is the set of Resilmesh functional components that enables the application functional components to fulfil their purpose through the provision of connectivity and aggregation services. It comprises the Collaboration Mesh and Aggregation planes.
Framework	An alternate word for platform - **deprecated**.
System	The combination of the Application Planes and the Platform that together realise the Resilmesh "product".
Technology	The combination of physical computing components (hardware) and digital instructions/programs (software) that work together to process information and solve problems to realise the Resilmesh functional components, platform and applications.

Executive Summary

This document is the final outcome of Task 2.3 "System Architecture Design" and outlines the final version of the High-Level Functional Architecture (HLFA) defined for Resilmesh.

The architecture has been built considering: (1) use-case scenarios and requirements from Task T2.1, (2) non-functional requirements from T2.2, and (3) functional requirements derived from the project's technical goals, mainly addressed in WP3, WP4, and WP5. The experience in first prototype and work done so far mainly in WP3, WP4, and WP5 has allowed to improve the architecture over the first version.

The HLFA supports Resilmesh's goal of enabling real-time defense of critical business functions through a Cyber Situational Awareness (CSA)-based security orchestration and analytics platform. It aims to help organizations:

- Reduce attack surface complexity by improving visibility and interoperability,
- Enable flexible placement of security controls across dispersed infrastructures,
- Counter advanced threats using AI for early detection, prediction, and risk awareness.

The architecture defines each component's structure, functionality, services, interfaces, and their interrelations. It serves as a baseline for technical development, guiding platform design in WP3, algorithm development in WP4, and incident response and mitigation in WP5.

The architecture consists of several planes:

- **Infrastructure Plane**: Underlying managed cyber systems,
- **Aggregation Plane**: Normalizes and aggregates data from multiple sources,
- **Collaboration Mesh**: Connectivity layer supporting Resilmesh operation,
- **Threat Awareness Plane**: Functions for anomaly detection, alerting, and attack prediction,
- **Situation Assessment Plane**: Provides overall cyber situational assessment,
- **Security Operations Plane**: Decides and enforces mitigation actions through CoA playbooks.

The document also outlines key security workflows and identifies extension points and open challenges for future development, particularly through WP8.

1 Introduction

1.1 Motivation & Activities

Resilmesh is devising, designing and implementing a cyber situational awareness architecture inspired by the Security Orchestration and Analytics Platform Architecture (SOAPA) toolset, aimed to improve digital infrastructure resilience.

The architecture deals with the digital infrastructure complexity and heterogeneity by providing tools to give them better awareness of environment dependencies, threats and risk while preserving privacy

This final version of Resilmesh architecture defined herein provides insights to business modelling, including information about which are the core components of the architecture (that cannot be replaced), infrastructure support, and/or plug-in modules (e.g. tools that can be used depending on a use case or configuration, or tools that can be replaced if needed). A new section 3.2 has been added for this purpose.

Resilmesh architecture is extensible and modular to cope with new technologies that might facilitate resilience. In this sense, the document dedicates a section to describe points of extensions, expected to be covered by projects funded by Resilmesh OpenCalls. In particular, the Resilmesh collaboration mesh has been appointed as a key topic to be addressed by Open Call parties to add to the Resilmesh functional scope.

The architecture has been designed considering the fact that the associated implementation platform needs to be validated in diverse infrastructure categories. In this sense, explicit description of components applicable to different use cases and domains scenarios will be reported as part of WP7.

The main workflows and interactions between functional components have been revisited with some changes, enumerated for the reader's convenience at the beginning of Section 3.

1.2 Resilmesh Scope and Limitations

The scope of this document is to define the reference high level functional architecture

1.3 Ethics Considerations

We follow the code of conduct specified in D1.6 (Data Management Plan). An extension of D1.6 was later created to address specific considerations for the pilot activities (Pilot Data Management Plan) - specific to the architecture in use during the pilots. In essence: ethics considerations related to this deliverable have been tackled in the following ways. Specifically:

- **Personal data and anonymity**: No means of identifying participants beyond the Resilmesh Consortium is collected or maintained during or after the completion of this document. No personal data is collected for this document.
- **Confidentiality**: No means of identifying participants nor relevant data are collected or maintained for this document. Data will be protected and kept confidential and will be shared exclusively within the Resilmesh consortium.
- **Usage of the findings**: Findings will be used for Resilmesh innovations and research applications. For example, they will be used to form a requirements analysis for the Resilmesh solution, and might be used in future studies and publications.

Participants in this document are Resilmesh Consortium members, and they are informed of who to contact for comments, questions or raise complaints. Consent is not required as there is no management of personal data.

The aforementioned Pilot Design Data Management Plan should be understood as building upon and providing a more granular level of detail to the overarching Data Management Plan [D1.6]. The pilot-specific plan complements the DMP by focusing on the practical application of these principles and regulations within the specific context of the pilots. For broader data management principles, ethical considerations, and data protection regulations applicable to the entire Resilmesh project, readers should refer to the main DMP.

1.4 Report Structure

The document is structured as follows. Section 1 serves as an introduction to the scope, purpose, and context of the project. Section 2 provides the architecture Foundations and State of the Art analysis. Section 3 defines the architecture and main workflows, including main changes to the architecture of version 1. Section 4 details the functional components and associated services. Section 5 includes the possible point of extension of the architecture. Finally, Section 6 concludes this report.

2 Architecture foundations

2.1 Resilience principles in Resilmesh

The Resilmesh resilience approach is grounded on several key resiliency best practices described in NIST publication SP 800 160 R2 *Developing Cyber-Resilient Systems*. Thus, the architecture includes features such as: Analytic Monitoring, Contextual Awareness, Analytic Monitoring, Coordinated Protection, Dynamic Positioning, Adaptive Response

2.1.1 Cyber situational Awareness

Resilmesh aims to enhance the cyber resilience of critical infrastructure security teams by leveraging Cyber Situational Awareness (CSA). CSA, adapted from fields like aviation and the military, consists of three interdependent levels: perception, comprehension, and projection. Perception involves identifying system assets, understanding their interdependencies, and being aware of potential internal and external threats—supported by tools such as intrusion detection systems and threat intelligence. Comprehension builds on perception to understand the impact of threats based on knowledge of critical missions and assets, relying heavily on data analysis and visualization. However, both perception and comprehension face challenges due to the volume and heterogeneity of data, which often leads to information overload. Projection is the final level of CSA and involves predicting future cyber events, such as the next steps an adversary may take, where and when attacks may occur, or how the overall cyber threat landscape may evolve.

Techniques include alert correlation, predictive blocklisting, and time-series forecasting. However, accurate projection relies on strong perception and comprehension. CSA enables organizations to assess risks and take timely mitigation actions, forming the foundation for mission-based cyber resilience strategies. While many technical tools support CSA, including open-source and commercial options, effective implementation also requires coordination across technical, operational, and managerial levels. Within Resilmesh, CSA is key to understanding an enterprise's security posture and threat environment, translating insight into timely defensive actions.

For further information about the reader is referred to D2.2

2.1.2 Collaboration Mesh

The Collaboration Mesh in Resilmesh focuses on enhancing the interoperability and integration of various often-isolated security tools and controls. It relies on a connectivity underlay that includes the protocols, functions, and mechanisms necessary to operate across distributed environments. This underlay supports the creation of data processing pipelines for event and anomaly detection and leverages container orchestration to manage and deploy security microservices across edge and cloud infrastructure. It also incorporates a data streaming backplane, a pub/sub messaging broker, and a service mesh for enhanced application resilience, observability, and security in critical deployments.

Additionally, the Collaboration Mesh emphasizes composability and interoperability, aligning with industry standards such as Gartner’s Cybersecurity Mesh and the Open Cybersecurity Alliance's Open XDR initiative. These frameworks aim to promote interaction between diverse security tools using open standards, proprietary APIs, and integrations. Resilmesh will include an operations orchestration framework that integrates multiple security operations center (SOC) tools into a unified platform, automating key processes like cyber threat intelligence (CTI) processing and event management. This framework will be built with open-source components and standard protocols such as STIX for CTI sharing and OpenC2 for command execution,

enabling the coordination and sharing of response playbooks using the CACAO standard.

For further information about the reader is referred to D2.2

2.1.3 Distributed AI-based Anomaly detection

Anomaly detection plays a key role in identifying potential cyberattacks by detecting deviations from normal network behaviour. Traditional signature-based methods have proven ineffective against zero-day threats, leading to the growing use of machine and deep learning for analysing complex threat patterns. However, deep learning typically requires centralized, resource-intensive infrastructures and large datasets, which can pose challenges due to data privacy and governance concerns. Federated Learning offers a promising alternative by enabling decentralized entities to collaboratively train a global model without sharing raw data, preserving privacy and distributing computational demands.

Federated Learning not only protects data confidentiality but also enhances the performance, robustness, and scalability of anomaly detection models by training across varied environments. This improves the system's ability to detect novel and sophisticated threats in real time. Moreover, because models are trained locally, deployment is faster and more efficient. Despite these benefits, risks such as model poisoning and membership inference attacks remain and must be addressed using techniques like Differential Privacy. The scalability of Federated Learning to thousands of nodes further supports real-time threat monitoring and mitigation across complex, distributed networks, making it a valuable approach for modern cybersecurity frameworks.

For further information about the reader is referred to D2.2.

2.1.4 Security Operations and Analytics Platform Architecture

Resilmesh can be visualised or mapped as a Security Operations and Analytics Platform Architecture (SOAPA)[SOAPA].

SOAPA represents a consolidation of the traditional SIEM with evolving enterprise security operations and analytics.

A SOAPA security operations stack consists of:

- **Common data services** in security operations manage a growing volume of diverse data types, amounting to terabytes per day. This data is ingested, processed, and prepared for analysis within SOAPA which centralises these functions and allows analytics engines to focus solely on analysis tasks.
- The **software services layer** within SOAPA, akin to traditional middleware, handles the delivery of data elements to analytics engines in appropriate formats and contexts.

- The **analytics layer** within SOAPA is where data is transformed into actionable insights using tools like threat intelligence platforms, behavioural analytics, and SIEM (Security Information and Event Management) systems.
- Finally, the **security operations layer** within SOAPA executes actions based on the analysed data, such as system quarantining, security control modifications, or software patch installations, among other security tasks necessary for effective response and mitigation.

These are organised architecturally in the SOAPA architecture as shown in Figure 1.

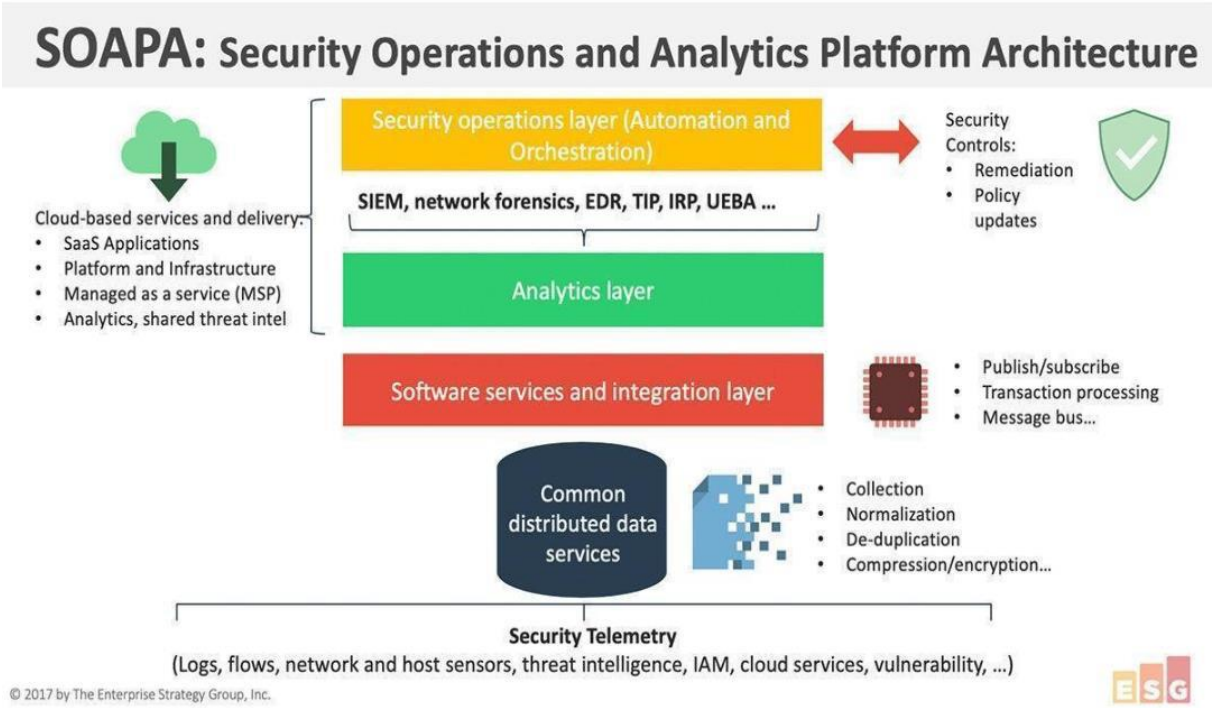

Figure 1 – SOAPA architecture

As [Oltsik] notes: "*Within SOAPA, SIEM -like functionality still plays a starring role, often aggregating analytics data into a common repository. But unlike the past, SIEM is one of several security tools within SOAPA, …. SOAPA is a dynamic architecture, meaning that new data sources and control planes will be added incrementally overtime*"

Since Resilmesh does exactly this it may be very useful **for dissemination purposes** to show how Resilmesh can be presented as a SOAPA [SOAPA] and we now consider how this can be done.

It should be noted that SOAPA is NOT an alternative architectural representation for Resilmesh.

Broadly we can consider the mapping of Resilmesh to SOAPA concept and layers as in next table.

Table 3 - SOAPA mapping to Resilmesh Planes

SOAPA Layer	Resilmesh Planes
Security Operations Layer	contains parts of the collaboration mesh plus SOAR functions.
Analytics Layer	contains Threat awareness and Situation assessment planes
Software Services Layers	contains parts of the collaboration mesh function
Distributed Data Services	contains the Aggregation Plane aggregation functions

This is depicted in the figure below which shows how the Resilmesh functional components map to the SOAPA layers.

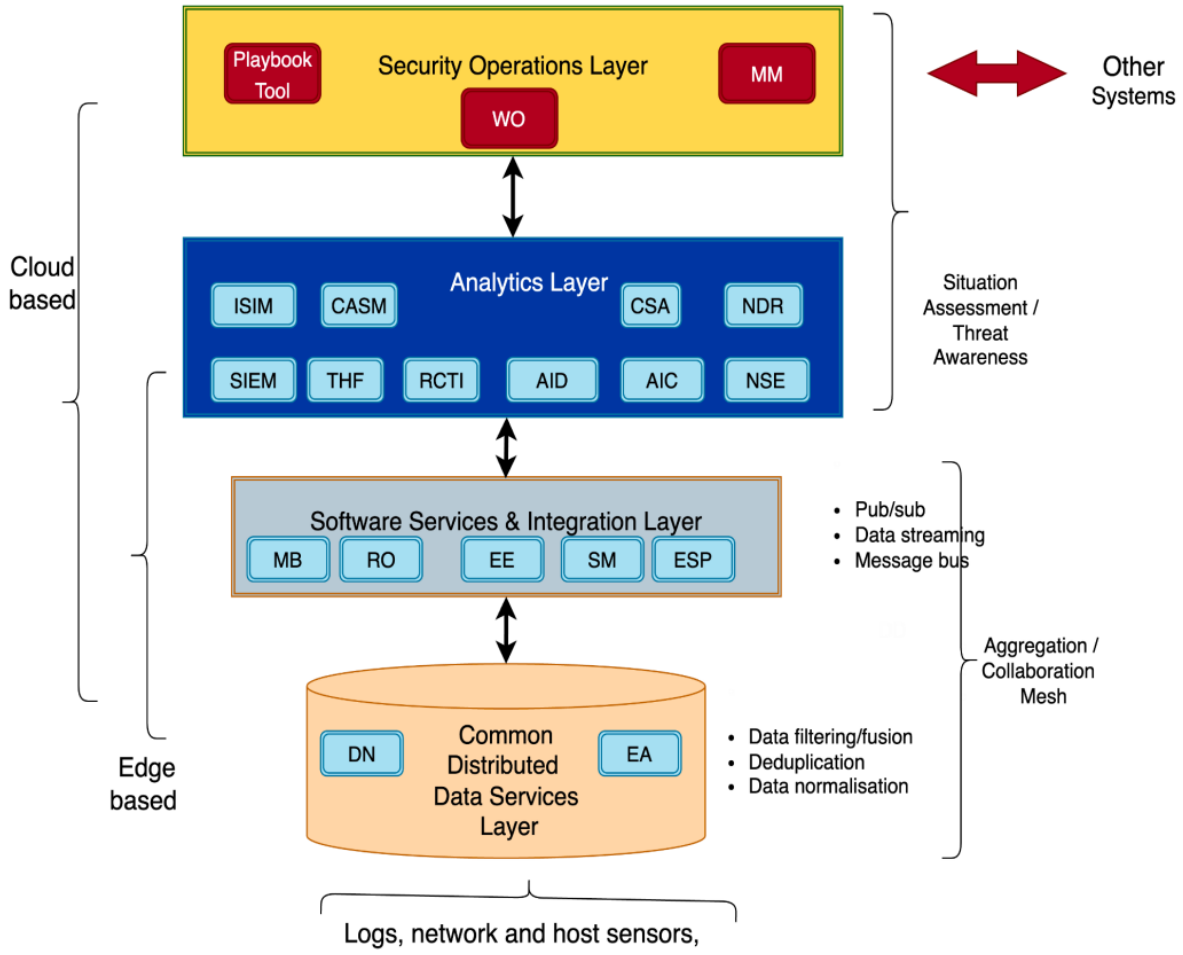

Figure 2 - Resilmesh SOAPA

3 High Level Architecture design

This section describes the Resilmesh High Level Functional Architecture design. The section dedicates different subsections to describe the planes and the associated functional components that fall into each plane. In addition, this section describes the main workflows envisioned in the project to satisfy the project goals.

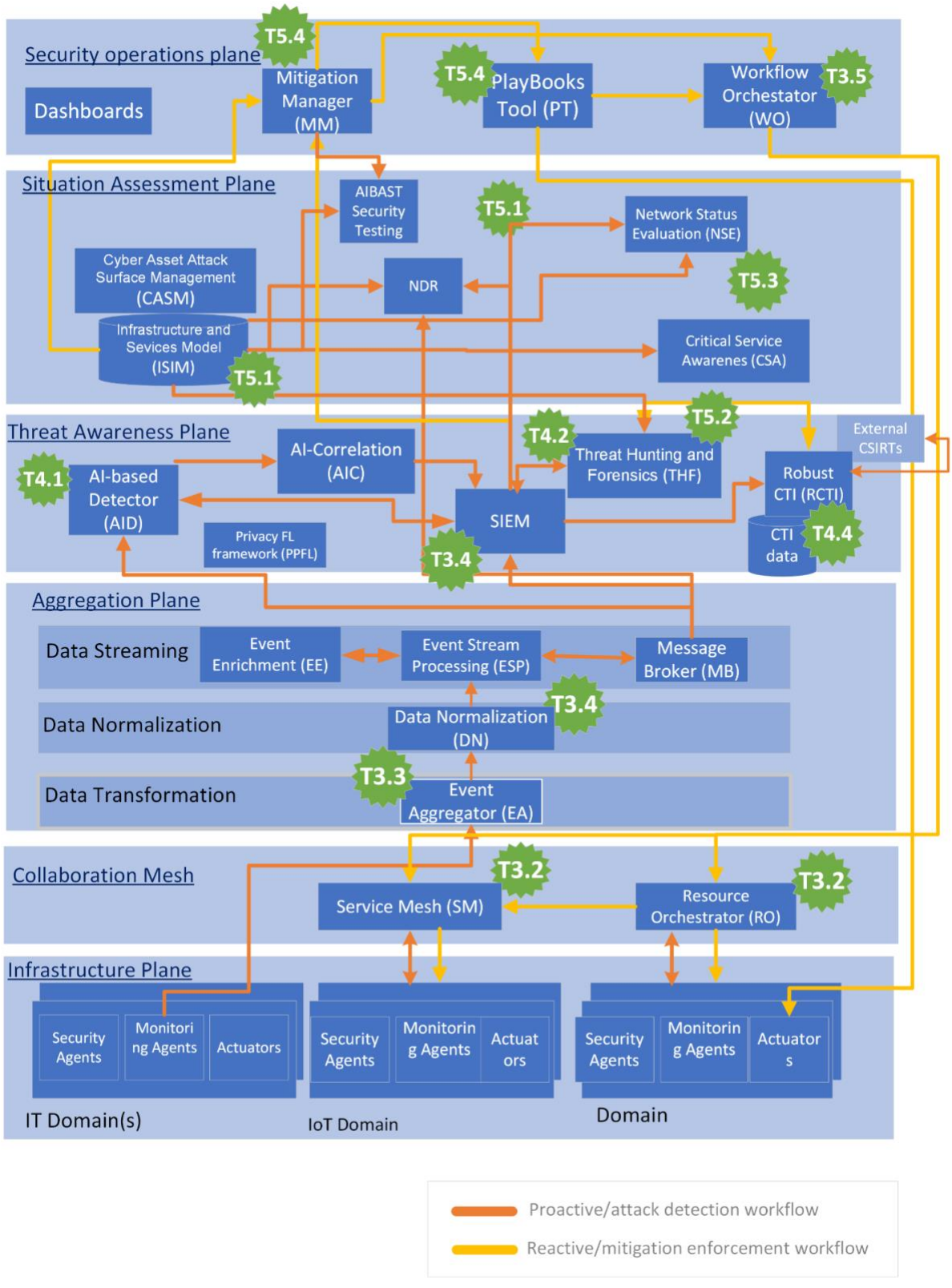

Figure 3 - High Level Architecture

3.1 Main architectural changes between version 1 and version 2

The following list enumerates and summarizes the main changes that have been accomplished in the architecture since version 1 in deliverable D2.2.

- AIBAST component intended to perform AI security testing, has been added into the architecture that was not included in version 1.
- ISIM has received a new interface that can push events to the Broker with information about discovered assets and vulnerabilities, so that other components such as MM can subscribe to the topic and receive said events.
- NSE will no longer have a Risk Score interface for the MM and will instead push all Risk Score data to ISIM, allowing MM to retrieve all contextual information from its existing ISIM interface.
- NDR's interface with SIEM is now bidirectional, in order to allow NDR to publish anomalous behavior reports and generate SIEM alerts. Additionally, NDR now has a new interface with MB and uses real time event streams as input. NDR will no longer interface directly with MM and will instead have its alerts forwarded by the SIEM, streamlining MM's alert input to come from a single source.
- AID will now include flow processing functionality, provided by a new Flow Processor subcomponent.
- CSA will no longer interact with MM, as this information is already stored in ISIM.
- MM's threat mitigation process has been updated to make use of AI constraint solvers in order to optimize its decision-making process.
- RCTI has been updated with an additional subcomponent, PP-CTI, which enables secure sharing of CTI events by transforming data in compliance with user-defined privacy policies.

3.2 Deploying Resilmesh to different Domains

As noted in the proposal document Resilmesh aims to

- **reduce CyS attack surface impact** by developing tools to combat **complexity** (better visibility of assets and services and their dependencies), **heterogeneity** (interoperability and extensibility) and **dispersed infrastructure** (flexible placement of security controls across the infrastructure)
- **combat APT sophistication** by developing advanced AI algorithms and tools for early and ongoing attack detection and prediction and improved situation and risk awareness
- **adapt to evolving security architectures and best practices** by highlighting Resilmesh enabled security best practices to prepare for disruption by APTs as well as Resilmesh 'zero trust ready' approaches

and also:

"*Cyber system domains include a wide range of civil and critical infrastructures that have very varying technologies, topologies, and application requirements. Topologies can be widely dispersed (water and energy infrastructures), concentrated in a few locations (manufacturing, health) or widespread (communications infrastructure). Cyber system resources are a mixed of constrained (IoT/edge) and powerful (cloud) computing devices and maybe a single technology (IT or OT) or a mix of both. Resilmesh use case pilots have thus been carefully chosen to demonstrate the applicability of the Resilmesh approach across these different cyber system domains and are also designed to validate the full complement of platform features over the three use cases …. Moreover, the platform provides baked-in extensibility 'hot-spots'/hooks to facilitate the easy addition of new platform functions as well use of the platform in new domains*".

Resilmesh was therefore designed to be deployable in a variety of business types and sizes and thus avoids a 'one size fits all' philosophy to allow the customisation and tailoring of the system to meet individual end-user deployment needs. **In this section we elaborate on this platform and application deployment flexibility and customisation capability to describe how Resilmesh meets the above business goals.**

3.2.1 Resilmesh Core System

The Resilmesh core *system* provides:

- security event collection, processing, aggregation, event alerting, correlation, response and threat intelligence sharing (MISP)
- a set of key integration reference points (IRP) - hot spots- that enable extending the platform for customised deployments

Specifically, the IRPs consist of

- *Sensor IRP (SIRP*) -the point at which security events and logs are ingested in the system from security sensors – shown as monitoring agents in the Infrastructure Plane. The SIRP provides extensibility through ingestion of different types of data and conversion to a standard format. SIRP acts as an event aggregator (sometimes also known as 'unified logging layer' in the industry) and is implemented by Vector in Resilmesh.
- *Connectivity IRP (CIRP)* – provides connectivity and message delivery from any functional component to any other functional component across any type or range of network. The CIRP is implemented by the Message Broker – by default NATS.
- *Wazuh IRP (WIRP)*- Wazuh plays a critical role as an integration function in Resilmesh. It can forward both unprocessed events (logs) or actual alerts to other applications and it also provides a query based API to allow applications to access its data. Moreover it complements the SIRP to enable heterogeneous event formatting syntaxes to be adapted to a common , Wazuh, based format. Wazuh/OpenSearch also allows the incorporation of extra functionality such as dashboards and advanced AI models.

- *Asset IRP (AIRP)*- This IRP gives access to the asset management database to security applications for risk calculation, mission criticality analysis, attack mitigation manager, etc. This IRP is provided by the ISIM and CASM functional components.
- *Consolidated Dashboard IRP* – This is both a single launch point for all applications as well as a consolidated set of UI for the Situation Assessment plane.

The Resilmesh core system then consists of the following functional components that together provide the basic Resilmesh event handling and extensibility capabilities – see Table 2.

Table 4 - Resilmesh core components

Functional Component	Purpose
Vector	Event Aggregation
Docker Compose	Container resource orchestration
NATS	Messaging broker
MISP	CTI sharing
Wazuh	SIEM – event processing and alerting and XDR
ISIM	Asset data base
CASM	Collects asset data and stores in ISIM
SACD	Consolidated dashboards

In addition to IRPs - extensibility hooks – Resilmesh may also provides customisation through **substitutability** i.e the replacement, or augmentation, of some core **platform** components by others including;

- Vector – could be substituted, or complemented, by similar event aggregation functions such as Fluentd, Filebeat, Logstash, etc.
- NATS – could be substituted by similar messaging platforms such as Kafka. Note however this remove the use of the NATS service mesh capability – however the NATS mesh capability be substituted by a mesh such as Istio [Istio]
- Docker Compose- this could be substituted by a Kubernetes base approach for container deployment.
- MISP – could be substituted or complemented by other threat intelligence platforms.

3.2.2 Customisation for specific sectors/domains

The extensibility and substitution capabilities of the Resilmesh core enable customisation along the following dimensions

· **Complexity** – ISIM provides the asset management capability to view all the enterprise assets enabling the identification and mapping of different infrastructure layers – especially relevant for critical infrastructure domains. This can be complemented by including the Critical Service Awareness (CSA) functional component – leveraging the AIRP – to include mission to asset mapping.

· **Heterogeneity** – can be realised in a number of ways.

  - Event data from different sensor types (including domain specific event types and protocols e.g. Modbus) can be adapted to the Resilmesh event format via the SIRP and WIRP

  - Data or protocol specific anomaly detectors (models algorithms) can be added by leveraging CIRP and WIRP to add new detection functional components and to forward the collected event to the detector – see also the

  - Domain specific asset types can be modelled by extending the ISIM schema and adding CASM asset type collection agents. An example of this can be seen for the use of robots assets in the Smart Manufacturing pilot/use case.

· **Dispersed environments** – the message broker i.e. NATS provides the capability to connect both platform and application functional components over very wide areas and with very light resource footprint where required. This will allow adapting Resilmesh to any type of geographical topology. The inbuilt service mesh capability can also enable more secure and resilient operation of the connectivity mesh via the use of the NATS service mesh. The Resilmesh substitutability capabilities also enhance this though the possibility to scale up the resource orchestration (Kubernetes) or implement an alternate service mesh (Istio)

3.2.3 Other features allowed in Resilmesh

· **Addition of extra/enhanced security functions**: Extra security may be required to expand the scope of the core system e.g. a security operations layer or to provide some specific detection or analysis functionality (e.g. improved correlation, threat hunting). These can be enabled through selective use of the IRP to access, process and display data. These addition can be custom design or include already existing Resilmesh components such as

  - Threat hunting

  - AI based correlation

  - Security Operations layer

  - Event enrichment

- **Adaptation to legacy systems**: via extension or substitution of either platform or application components. For example if an organisation is already using event or message broker components these MAY be substitutable for the corresponding platform component. At the applications level (Threat, Situation Assessment) existing controls MAY be incorporated through the use of open protocols such as the OCA OXA family or the development of custom adapter to the relevant IRP, i.e., WIRP, AIRP and SACD.

Taken together these integration capabilities give Resilmesh the capacity to be deployed and adapted to provide SOC capability to almost any enterprise type we are likely to encounter.

3.2.4 Zero Trust Domains

The capabilities already listed above provide extensive support for Zero Trust deployments. However Resilmesh provides two specific capabilities to enhance ZT.

- Zone based Federated Learning-based Anomaly Detection (FLAD) – federated learning detection can ensure privacy preserving anomaly detection for ZT zones and hence further enhance the ZT enclave isolation architectural principal

- Zone base risk assessment – Resilmesh MAY (depending on zone configuration) be able to provide zone specific risk assessment enabling security operators to quickly identify zones which are most at risk.

3.3 The Infrastructure Plane

The network communication between components is handled in this plane. It includes both physical and virtual network elements responsible for tasks such as forwarding the traffic according to commands and rules received by the Network Controllers through the southbound API. In virtualized scenarios, this infrastructure includes the cloud computing technologies (i.e., computing, storage and networking) to deliver virtual Infrastructure-as-a-Service (IaaS). Special controllers are used to control devices in more heterogeneous scenarios that include IoT.

Computing and communications Cyber Systems (CyS) domains include a wide range of civil and critical infrastructures that have very varying technologies, topology, and application requirements. Topologies can be widely dispersed (water and energy infrastructures), concentrated in a few locations (manufacturing, health) or widespread (communications infrastructure). CyS resources are a mix of constrained (IoT/edge) and powerful (cloud) computing devices and maybe a single technology (IT or OT) or a mix of both. Resilmesh use case pilots have thus been chosen to demonstrate the applicability of the Resilmesh approach across these dimensions e.g. across the different topologies types, concentrated (flexible manufacturing), widespread (regional infrastructure) and distributed (digital trust). In Resilmesh this

Infrastructure plane represents the infrastructure needed for the three use cases: digital trust service, and flexible manufacturing and civic regional infrastructure.

- The **manufacturing environment** considers several industrial robots, powered by the robotics "*de facto* standard" of Robot Operating System (ROS) - middleware. In such a manufacturing environment, IT and OT operations are interconnected according to the well-known layered Purdue Enterprise Reference Architecture and Purdue reference model for ICS, SCADA and OT systems. The manufacturing environment is structured in at least 5 levels, including, corporate Level 4 IT networks, Level 3 Operations, Level 2 - Central control Stations, Level 1 Control Network and controllers and Level 0 comprising physical actuators. Hardware components will include widely used and popular industrial grade robotics platforms, industrial grade networking equipment and engineering infrastructure. Software components will include firmware versions by robot manufacturers, drivers (official manufacturer ROS drivers) and available ROS versions.
- The **digital trust service** infrastructure is a multi-cloud environment comprising two public cloud providers and one on-premises provider. The on-premises provider offers two data centres, which are used to support disaster recovery (DR) strategies. The overall environment includes physical systems such as firewalls and other security appliances. Services are distributed across the cloud and on-premises infrastructures and are primarily containerized, enabling flexibility, scalability, and high availability.
- The **civil region infrastructure** is a physical and virtualized IT system and network composed of several data centres and information systems that are deployed in a distributed and dispersed subnetworks managed by the civil government.

Security probes or **sensors** are deployed in the infrastructure to support monitoring the managed system. For instance, Packetbeat [1]agents is an analyser that gathers and sends data from hosts and containers, FileBeat [2] agents for collecting logs from security devices, cloud, containers, hosts. In addition, special purpose probes for OT can be shipped to gather logs, and data from the OT network protocols and system logs.

In addition, this infrastructure layer will host the set of **actuators** that will help to configure dynamically and on-demand the security command and controls, and enforce the remediation(s) and mitigations as requested by the Resilmesh platform. These actuators and security agents can be managed through diverse protocols such as OpenC2.

3.4 Aggregation Plane

The aggregation plane is the initial preprocessing step needed to ensure the data coming from the Infrastructure Plane flows properly throughout the pipeline. It

[1] https://www.elastic.co/es/beats/packetbeat
[2] https://www.elastic.co/es/beats/filebeat

collects, aggregates, and normalises data and events from multiple heterogeneous sources including logs, IDS, network sources, AI models etc. It contains a rich set of data/event (including network traffic) filtering, fusion, logging, storage and forwarding functions.

It is comprised of three functional sub-layers:

- **Data Transformation** - this receives the data from on-device agents and other sensors and filters and aggregates the incoming raw data. It contains the *Event Aggregation* (EA) functional component.
- **Data Normalisation** - this layer adapts the aggregated data to a common data format scheme for processing in Elasticsearch - the Elasticsearch Common Schema (ECS). It contains the *Data Normalisation* (DN) functional component.
- **Data Streaming** - this layer distributes the normalised data to the upper layers and also performs further processing on the data streams, if required. It contains the *Message Broker (MB), Event Stream Processing (ESP) and Event Enrichment (EE*) functions.

Event Aggregation (EA) : This functional component ingests event logs from multiple sources and transforms the events via various operations (routing / logging / fusion / filtering / augmentation / reduction / monitoring) and then outputs the data to one or more destinations via the streaming layer. Outputted events are transferred to the DN function

Data Normalisation (DN): transforms data from multiple disparate formats coming from different sources, to a single common format that can then be used for analytics, visualisation, reporting, etc

Message Broker (MB) : This is an intermediary function that applications and services use to communicate with each other to exchange information. Message brokers can be used to route and deliver messages to the required destinations.

Event Stream Processing: This is an optional component that may be deployed for specific purposes. It provides a capability for real-time processing of continuous data streams through.

- **Stream Processing (SP)**: involves the real-time handling of data, where computation occurs directly as data is generated or received. Most data is produced incrementally over time as a sequence of events.
- **Complex Event Processing (CEP)**: is a generalisation of traditional stream processing for aggregating, processing, and analysing data streams in order to make high-level inferences about complex events within the business domain using models of causality and conceptual hierarchies. CEP is often used for tasks such as event correlation.

Event Enrichment

This function enriches the events with contextual information from the enrichment API provided by partner Silent Push. Silent Push scans, clusters, scores and enriches the global IPv4 range in a first-party database that outputs Indicators Of Future Attack

(IOFA) – domain, IP and URL data that explains the relationship between billions of observable data points across the internet. It is a threat intelligence mechanism that allows security teams to pinpoint the origin, function and risk level of a domain or IP address, by applying multiple categories and subcategories that provide up to 10x more context than standard DNS lookups and queries are able to provide.

3.5 The Collaboration Mesh Plane

This is the Collaboration Mesh connectivity underlay as described previously in the Architecture Foundations section. Note that the SM is an optional component and is deployed only for specific scenarios. Also while a microservice based architecture (with container RO) is the preferred application software architecture, other approaches such as bare-metal may be used when required.

3.6 The Threat awareness plane

The Threat Awareness plane of the Resilmesh platform contains a set of information processing and analysis functions to manage anomaly detection, event correlation and alerting, and attack detection and prediction. The layer functions are implemented as a combination of new development and reuse of existing components.

The Threat Awareness plane includes an **AI-based Anomaly Detection (AID)** module to detect anomalies for any type of IT or OT application and/or network protocol. Models can be placed at the endpoint/edge or in the cloud as required. While Resilmesh supports the use of any ML or AI techniques to develop models, the focus is on the use of deep learning techniques. It will evaluate different multi-view deep learning approaches, such as multi-view fusion-based methods and multi-view alignment-based methods, to deal with the intrinsic heterogeneity of mixed technology domains. Resilmesh will evaluate both the use of centralised and the use of distributed edge/endpoint based anomaly detectors and will consider the use of stacked or hierarchical deep learning techniques for both feature level fusion and/or decision (model) level fusion. Centralised anomaly detector will use input data obtained through the Resilmesh platform. Edge/endpoint anomaly detector, in contrast to centralised anomaly detection, will include an additional component - network flow processing module, which independently processes network traffic and extracts relevant statistical features needed for anomaly detection.
This plane also includes a federated learning platform - the **Privacy FL Framework (PPFL)** - to support federated training of the different types of deep learning algorithms mentioned above. The platform will support both horizontal and vertical federated learning, as well as a combination of both. The final output of these detectors will be events sent to the AIC and/or SIEM correlation function, or as features to be passed on to other models.

The **AI Correlator (AIC)** functional component is intended to provide AI methods to correlate security events includes with application to i) classify security events for *event detection, event grouping,* and *event pattern extraction,* ii) *Intrusion detection* which deals with multi-stage and targeted attacks or *anomaly detection* to notify the security administrator about misuses and deviations from normal behaviour,

respectively and iii) *Intrusion/attack projection* based on incoming events, which allows early detection of intruder targets. AI correlation may be embedded within other components e.g. such as NSE or NDR or may exist as a stand alone component.

The **Security Incident and Event Manager (SIEM)** is a central component in Resilmesh and provides a number of capabilities including event logging (through Elasticsearch), event correlation as well as a range of XDR capabilities.

The **THF (TTP-based Hunting and Forensics)** TTP-based threat hunting focuses on identifying adversaries by their tactics, techniques, and procedures—the "how" of their operations rather than the "what" or "when." This approach leverages the ATT&CK framework developed by MITRE. THF supports the use of TTP-based hunting techniques for cyber attack investigation. THF will have an UI where the user can carry out the mentioned hunting and analysis features.

3.7 The Situation Assessment plane

The **Situation Assessment** plane contains a set of functional components that collectively provide cyber situational assessment capability to Resilmesh - and which, together with the Threat Awareness plane implements cyber situational awareness in Resilmesh. These components are:

The **Infrastructure and Service Information Model (ISIM)** captures and represents all the entities of interest in the environment including devices, networks applications (services), users, and data. The information model interconnects the pieces of information on the assets in the environment. When deploying the overall system in a new environment there is a need to fill the database with data from external sources. For each category of assets, their enumeration will be collected from existing databases, repositories, or service, or collected via a set of custom tools. Moreover, ISIM is able to interconnect information on assets with information on vulnerabilities, e.g., to indicate a present or suspected vulnerability of an asset. The ISIM provides REST API and GraphQL interfaces to allow applications to enlist assets and vulnerabilities and query their status.

The **Cyber Asset Attack Surface Management (CASM)** tool collects the data to be stored in the ISIM. The principal role of the CASM is to monitor the organisation's internal and external attack surface and security posture. It provides an interactive query capability to allow operators to determine cyber security posture based on the relationship between assets - based on ISIM. It also carries out domain (and sub-domain) enumeration to discover so-called ‘shadow-IT’ and to identify and manage threats discovered in Internet-facing assets using independent scans of the organisation attack surface. CASM can detect changes of asset status and trigger required actions or alerts to the user or the Mitigation Manager (MM) as required.

The **Network Detection and Response (NDR)** component serves as an analytics engine that receives input from multiple sources, including the Event Aggregator (EA), Event Stream Processor (ESP), and the SIEM. These components provide normalized,

enriched, and correlated security events from across the infrastructure. NDR processes this data using a hybrid detection approach combining both signature-based and anomaly-based techniques to identify known threats and detect deviations from normal behavior. Once a potential threat or anomaly is identified, NDR sends alerts back to the SIEM, which handles further correlation and forwarding to other systems such as the Mitigation Manager. In parallel, NDR also pushes detection results and contextual information to the Situation Awareness Command Dashboard (SACD), where they are visualized for human operators. This allows analysts to quickly interpret alerts, investigate incidents, and take informed mitigation actions, either manually or in coordination with automated response mechanisms.

The **Critical Service Awareness (CSA)** component will provide hierarchical risk assessment to aggregate infrastructure risk into a risk for the critical service or business mission (CS/M). It will implement tools to assess such risks. To achieve this, the component will use data stored in ISIM (information model).

The **Network Situation Evaluation (NSE)** functional component provides a risk assessment of the overall network based on input from other functions and can also project the attack intensity for the network. It provides visualisation of both current and future network risk status. It uses inputs from a number of tools to do this including the ISIM and SIEM.

AIBAST is an AI-powered tool that automates penetration testing using large language models (LLMs), enabling continuous, intelligent assessment of system vulnerabilities. Integrated into the Situation Assessment plane, AIBAST strengthens ResilMesh's cyber situational awareness by delivering real-time insights into exploitable weaknesses. Its findings seamlessly support the Mitigation Manager, helping prioritize responses and drive proactive defense actions.

3.8 The Security operations plane

The Security Operations Plane contains a set of functional components aiming to decide the most suitable mitigation actions that should be taken to respond to a detected incident, orchestrate these actions as CoA playbooks and analyse collected information to identify adversaries by tactics and techniques carried out during the incident.

Mitigation Manager (MM) Functional Component is responsible for deciding which mitigation actions, if any, are taken to deal with an incident. MM receives security alerts forwarded from the SIEM as mitigation requests. It then interfaces with the ISIM component to analyse mission, risk and network status projection as factors in the mitigation decision process, as well as to gather the information model captures and represents all the entities of interest in the environment (such as devices, networks, applications, data, or users).

MM will make use of the Optaplanner rule-based inference engine to infer the best mitigations for each particular situation, leveraging situational information on available playbooks together with the Cyber Situational Awareness data and Risk Scores provided by ISIM. Additionally, MM will employ a localized graph tracking algorithm based on the MITRE Attack Flow format in order to record ongoing attacks and predict potential next steps, information which will be fed to the logic solver to enrich the available context for calculating effective mitigations.

Once mitigation actions have been decided, MM interacts with **Playbooks Tool (PT)** through a REST API to trigger the orchestration of the selected mitigation playbook(s) to counter the incident. Then, playbooks Tool (such as Shuffle) launches CoA playbooks that will execute workflows which contain a set of several actions to mitigate incidents, like OpenC2 commands. Some of the mitigation actions could be network filtering, quarantining hosts or updating outdated software. Workflow actions also can enrich SIEM adding new rules (like yara or sigma rules) or execute other tools like Security Orchestrator that can enforce security or privacy policies.

Playbooks Tool should also be user-friendly as it should enable the user to create automated and exportable workflows and design associated PT apps to integrate tools like Security Orchestrator as a workflow action.

If a detected anomaly is confirmed as malicious activity, this is investigated, gathering information to understand the threat and identify potential adversaries. Once the gathered information is sufficient, mitigation actions can be deployed by PT to counter the attack.

To support PT, the **Workflow Orchestrator (WO)** will orchestrate and automate complex actions taken as a CoA playbook step. WO will perform part of the playbook actions (those which are difficult to implement in Shuffle PT) when requested by Playbooks Tool within a CoA playbook execution. WO then uses a subset of tools from Target Actuators to carry out the requested actions. PT can request an action to the WO via a REST/gRPC API, and then Workflow Orchestrator can provide information about the progress and outcome of the performed actions to the Mitigation Engine, establishing a constant feedback loop.

3.9 Main workflows

The workflows are grouped into three main blocks, detection workflows, situation assessment workflows, and mitigation workflows.

3.9.1 Detection workflows

3.9.1.1 Attack/incident detection flow

MB forwards each raw event to AID, which quickly checks whether the behaviour is anomalous; if AID flags it, the component emits an **Alert** to both SIEM and to AIC so that the correlator can consider it alongside other alerts.

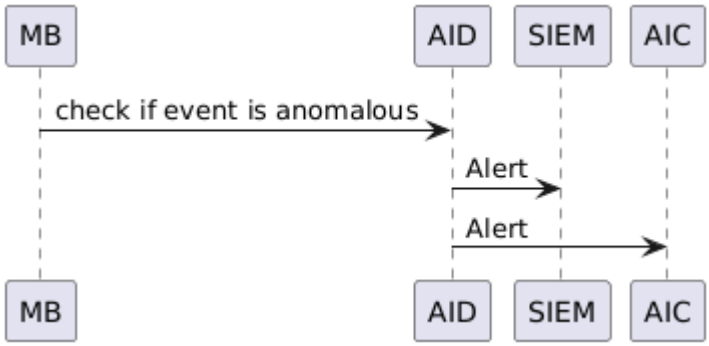

Figure 4 - Incident detection flow

3.9.1.1.1 AIC flow

Alongside AID, SIEM sends each individual **Alert** it receives over to AIC; the correlator (AIC) stores the incoming alerts, analyses them together to "Correlate event sequence", and when the combination of alerts looks suspicious it issues a higher-level **Alert of suspicious event sequence** back to SIEM for further action.

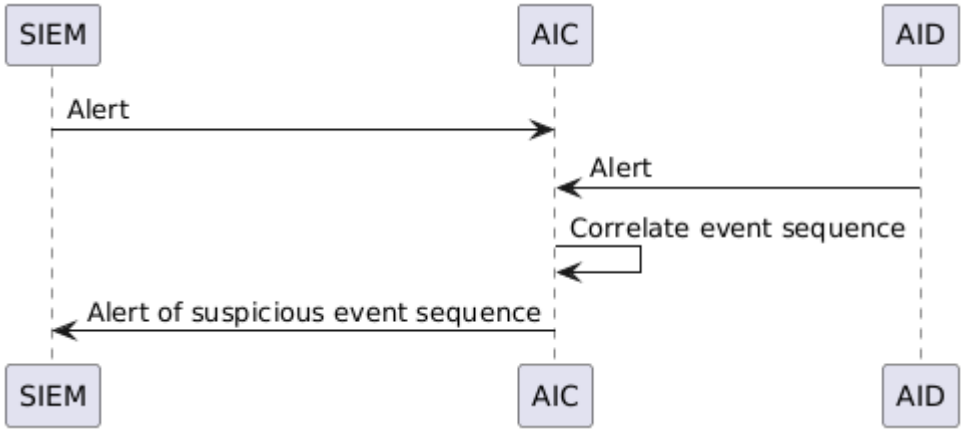

Figure 5 - Incident detection and Correlation

3.9.1.1.2 Cyber Threat Intelligence Sharing

This flow describes how threat intelligence is shared between Resilmesh and other systems.

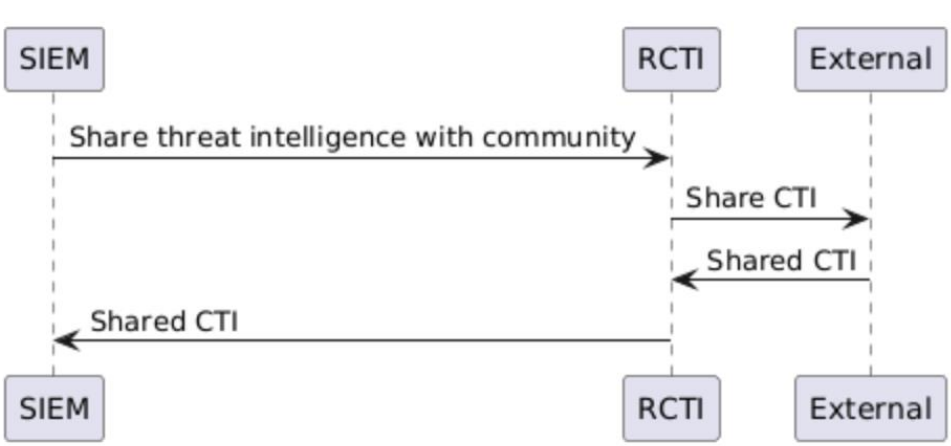

Figure 6 - Cyber Threat Intelligence Sharing workflows

1. The SIEM (or other functions) may share threat intelligence with other partners.
2. The system may also receive information from other partner

3.9.1.1.3 Threat hunting

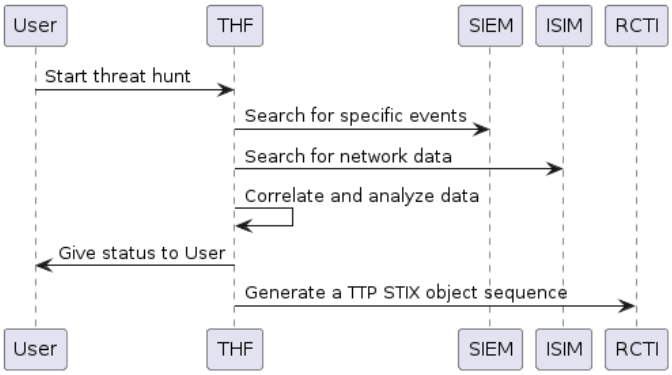

Figure 7 - Threat hunting workflow

1. The user (threat hunter) initiates a threat hint based on a specific TTP based hypothesis.
2. The THF fetches network and TTP data based on the provided hypothesis.
3. Depending on the specific hypothesis and user request different analyses may be carried out .
4. Hunt progress and status is displayed to the User.
5. Steps 1-4 are iterative and may be repeated a number of times.

6. In some cases the output may be shared as threat intelligence as STIX behaviour sequence object.

3.9.2 Situation Assessment workflows

The SA flow also has a number of alternate subflows detailed below. The following figure visualises flows that are related to situation assessment in Resilmesh. The ISIM component can be considered as the component that all other components use due to its role in managing data in the graph database.

The first flow represents how CASM interacts with the ISIM after it scans the network. Results obtained by CASM include IP addresses, domain names, network services, and software versions. Another functionality of CASM is to obtain Common Vulnerabilities and Exposures [CVE] from the National Vulnerability Database [NVD]. The information about assets and vulnerabilities is stored in ISIM.

The second flow by CSA is related to creating representations of enterprise missions in the ISIM. First, CSA is responsible for collecting the missions and representing them in a JSON file. Consequently, the descriptions of missions are stored in ISIM by posting them to an API endpoint.

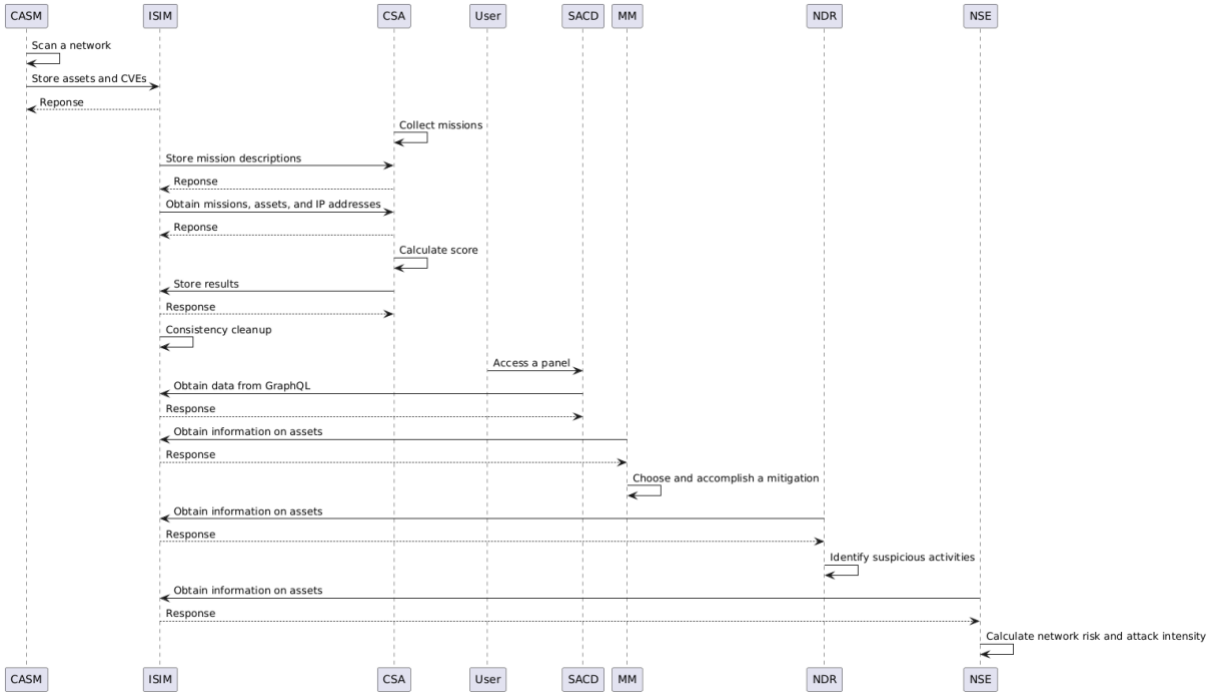

Figure 8 - Situation Assessment flows

The third flow expresses risk assessment by CSA. The CSA needs to obtain relevant information for the assessment in the first step, such as missions, assets, and IP addresses. It computes scores for cyber assets based on the obtained data. The results are consequently stored in ISIM.

The fourth flow represents cleanup of old and inconsistent data in ISIM. The fifth one represents that a user can interact with SACD to visualize information in a dashboard panel. It triggers communication with ISIM's GraphQL API to obtain necessary data that are visualized in the dashboard.

The remaining flows are related to MM, NDR, and NSE. They obtain information about assets to accomplish their tasks. In the case of MM, it is necessary to choose and employ a mitigation based on the information. In the case of NDR, the following task is to identify any suspicious activities based on the obtained data.

3.9.2.1 AIBAST Subflow

MM requests AIBAST to **initiate pen-testing**; AIBAST then tells RCGE to **perform** the necessary test actions and keeps polling RCGE for **status** updates; once the exercises finish, AIBAST assembles the results into a **pen-testing report** and sends it back to MM.

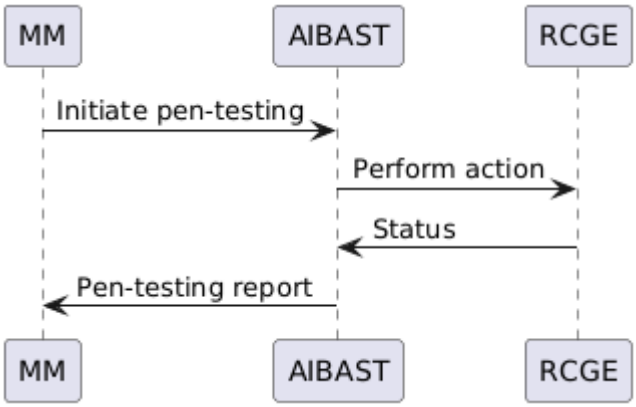

Figure 9 - AIBAST Subflow

3.9.2.2 NDR Subflow

The flow shows how **NDR** component interacts with other elements in the Resilmesh platform to detect and respond to suspicious network activity. NDR begins by retrieving event logs from the SIEM (Security Information and Event Management) system using a Get_Events request.

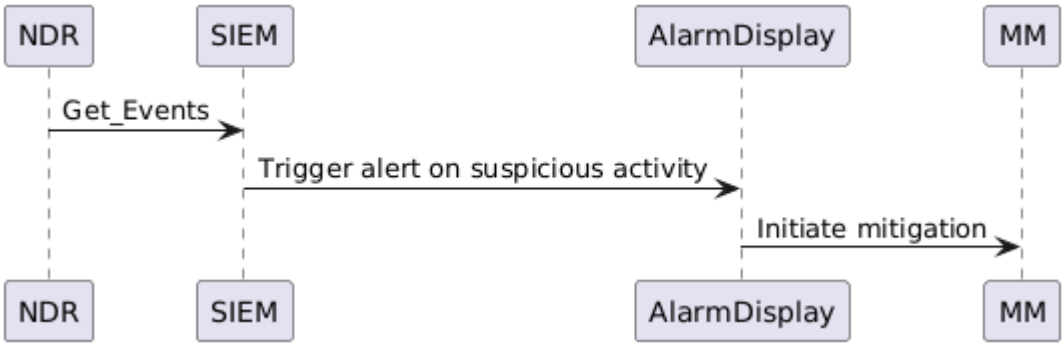

Figure 10 - NDR Subflow

Upon detecting suspicious activity, the NDR triggers an alert, which is forwarded to the AlarmDisplay component. The AlarmDisplay then notifies the operator and, based on predefined rules or manual intervention, initiates a mitigation request to the Mitigation Manager (MM). The MM then executes the appropriate response actions to contain or neutralize the threat. This workflow enables automated, real-time threat detection and response across the system.

3.9.2.3 NSE flow

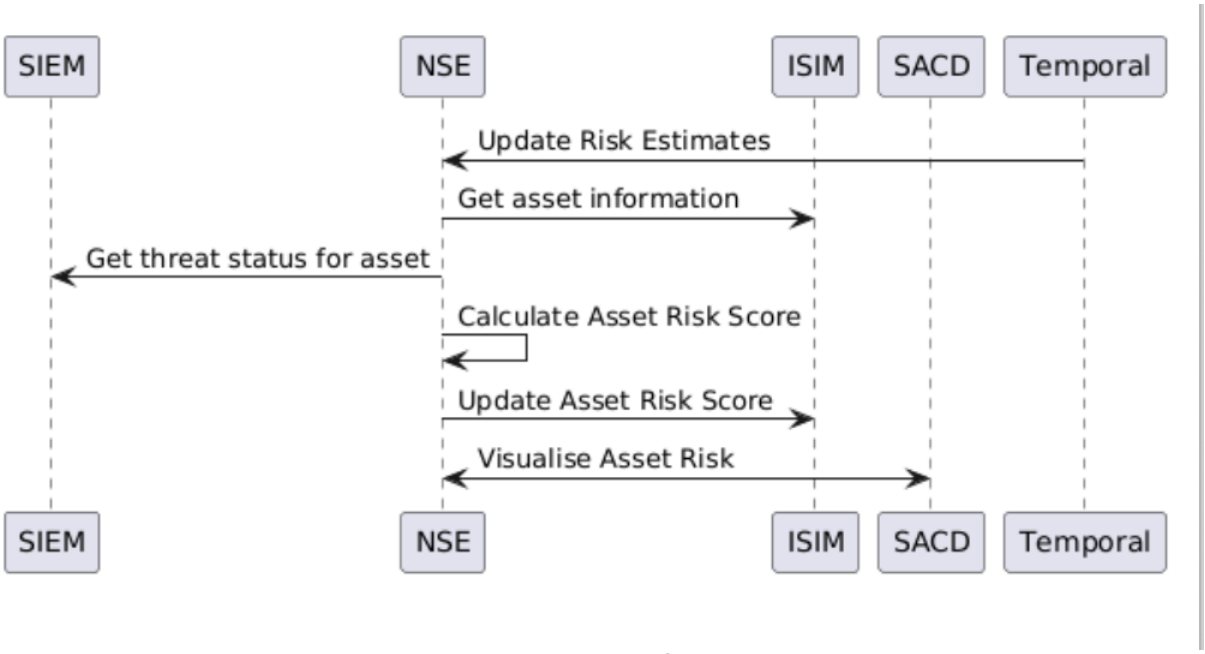

Figure 11 - NSE flow

This flow describes the operation of NSE for the estimation of asset risk.

1. The workflow is triggered periodically by a mechanism such as Temporal. The trigger will define the scope of the request.
2. NSE will fetch asset information from ISIM including criticality and vulnerability scores.
3. It will then fetch threat information related to the assets from Wazuh

4. It then calculates the risk score..
5. Updates the risk score in ISIM ..
6. The Risk score may be visualised on request from SACD.

3.9.3 Reactive/mitigation workflow

This flow describes the steps taken by the Resilmesh project in order to respond to a threat event using the mitigations inferred by the Mitigation Manager component.

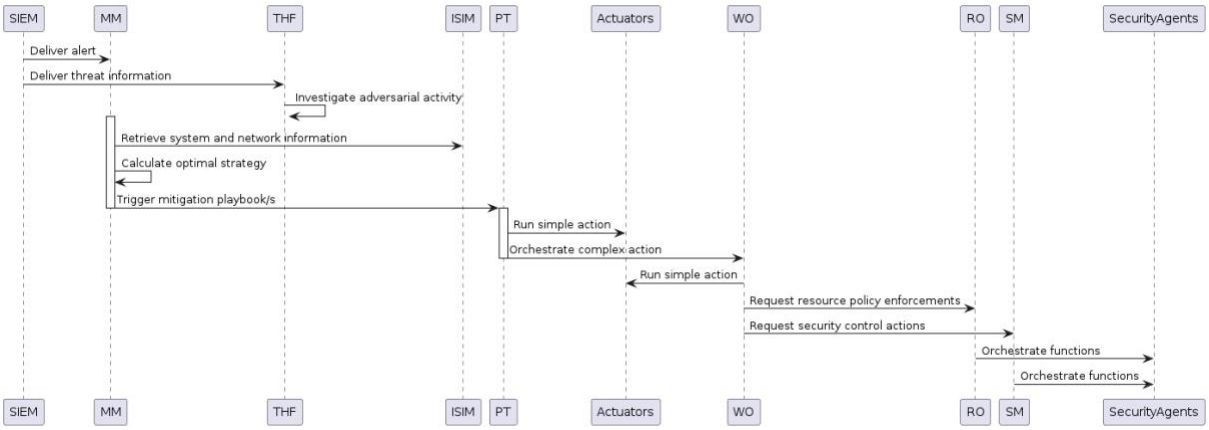

Figure 12 - Mitigation Flow

The first steps describe the initial detection, which causes the SIEM to forward a threat alert to the Threat Hunting and Forensics component. THF will be tasked with gathering contextual information to fully understand the threat, and to attempt to identify the adversaries responsible for the security event. After investigating the threat details, the THF will be able to generate reports and data visualizations. Using its correlation function, THF will link related activities across time and terrain to construct a coherent narrative of potential adversarial actions. All of these investigation and hunting processes will feed the hunting database with the fetched data and the outcomes of the correlation analysis and will also support further investigations, trend analysis, and threat intelligence enrichment.

Simultaneously, the alert will also be forwarded to the Mitigation Manager component. In order to enrich the contextual information to start the decision process, MM requires situational information and risk scores regarding the network and system status. MM will thus query ISIM to fetch this information, available through the CRUSOE data model and enriched with the NSE's risk score calculations.

Following its decision process, the Mitigation Manager will obtain a series of mitigation actions to counter the current threat. In order to enact them, their execution will be triggered by forwarding requests over to the Playbooks Tool component. PT will then start the execution of the requested mitigation playbook. Most playbooks will trigger actions in some target actuators. Among other options, this can be achieved by creating and sending OpenC2 commands to the actuators. Other playbooks might

require complex orchestration due to their complexity, and would thus be unsuitable for PT. To handle this situation, PT will contact WO for requesting the orchestration of the playbook complex actions. WO will then handle running the orchestrated actions through the proper target actuators.

Playbook mitigation actions can also provoke the orchestration of virtual network functions to ensure that the needed security actions are taken. WO will request container deployment orchestration and/or network policies enforcement to RO that will create a virtual network function (security agent) to accomplish the requested security actions. In a similar fashion, WO can request security functions or routing control actions to SM which also creates virtual network functions to orchestrate these actions. RO can manage container lifecycle, including deployment, scaling, updating and terminating. Moreover, it defines and enforces network policies to control the communications among containers within the cluster or externally. RO supports integrating with SM to facilitate advanced network management. By using SM, the WO can invoke security functions including end-to-end encryption (mutual TLS), authentication and authorization policies. Furthermore, WO is able to orchestrate routing control actions like configuring traffic shifting and mirroring.

4 Functional component descriptions

This section aims to detail the functional components that have been introduced previously in section 3. Each functional component is described using the same template that firstly defines a description of the Function associated to the functional component. Then, each functional component describes the main services that it will feature to implement the function in the platform. Each service defines the capabilities it offers, the type of service, who are the consumers services, pre and post conditions as well as the main envisioned interfaces.

4.1 Resource Orchestration (RO)

4.1.1 Function

The Resource Orchestrator manages containerised workloads and services through declarative configurations. It dynamically manages the lifecycle of containers, including deployment, scaling, and networking, and self-healing ensuring high availability and reliability of services. It automatically allocates and manages computing resources like CPU, memory, and storage across a cluster based on resource usage and demand, optimising performance and reducing costs. The Resource Orchestrator uses declarative configurations to streamline the deployment processes and manages the secrets, allowing applications to adapt to different environments without code changes.

4.1.2 Provided services

4.1.2.1 Orchestration Services

A. Description

Cluster Management

- The Resource Orchestrator has high flexibility to scale up or down automatically based on the workload requirements to optimise the usage of resources. It also maintains cluster availability over updates and failures with minimal downtime.

Container Orchestration

- The Resource Orchestrator can manage container lifecycle, including deployment, scaling, updating and terminating. It also continuously detects containers´ health and makes them recover from the failures, such as restarting or rescheduling failed containers on other working nodes. The Resource Orchestrator has integrated service discovery and load balancing abilities to allow containers to communicate with each other and distribute traffic efficiently. The Resource Orchestrator supports automated provisioning of resources, including ephemeral volumes and persistent storages.

Network Management

- Resource Orchestrator defines and enforces network policies to control the communications among containers within the cluster or externally. It also provides network isolation between different applications to enhance security and reduce interference. The Resource Orchestrator supports integrating with service mesh to facilitate advanced network management.

Security

- The Resource Orchestrator provides robust mechanisms like access control policies to restrict access to the orchestration platform, only authorized users and services can perform operations. The Resource Orchestrator can securely store and manage secrets of containers without exposing credentials in configurations or codes.

B. Capabilities

Enables deployment of Resilmesh functions in a flexible and dynamic manner.

C. Type

External.

D. Consumers

- SOAPA layers

E. Pre-conditions to consume the service

-

F. Interfaces

We do not give a detailed interface breakdown but instead point to the Docker Swarm documentation.

nRO_services	**Detailed Description**	These interfaces enable the applications to be orchestrated as required for different deployments.
	From provider	RO
	To Consumer	RO
	Technology	Rest API
	API Documentation	https://kubernetes.io/docs/concepts/cluster-administration/ https://docs.docker.com/engine/swarm/
	Partners involved	TUS,

4.2 Service Mesh (SM)

4.2.1 Function

Service Mesh Platform is a dedicated infrastructure layer that enables, secures and monitors service-to-service communications within distributed applications. The Service Mesh Platform supports automatic service discovery within the network, so that services can identify and locate each other in a highly dynamic environment where services frequently change due to deployments, scaling, and failures. It has traffic management functionalities, including load balancing that distributes network traffic across multiple backend services to ensure reliable and efficient data handling, and fine-grained traffic control such as detailed telemetry, encryption settings, and specific routing rules. The Service Mesh Platform enhances the security posture of the overall application infrastructure through fine-grained access control policies, transparent TLS encryption, network segmentation, and Authentication, Authorization and Audit (AAA) tools. It offers detailed insights into the behaviour of services, including monitoring, logging, and tracing capabilities to help diagnose and resolve issues quickly.

4.2.2 Provided services;

4.2.2.1 Mesh Services

A. Description

A service mesh is an infrastructure layer designed for managing interactions between services/microservices. It helps your microservices run smoothly, securely, and stable (while telling you what is going on with them. It handles things such as discovery, load balancing, failure recovery, metrics, monitoring, rate limiting, access control, and authentication.

The following services, amongst others, are provided by the platform:

- Service Discovery
- Security

- Observability
- Routing Control

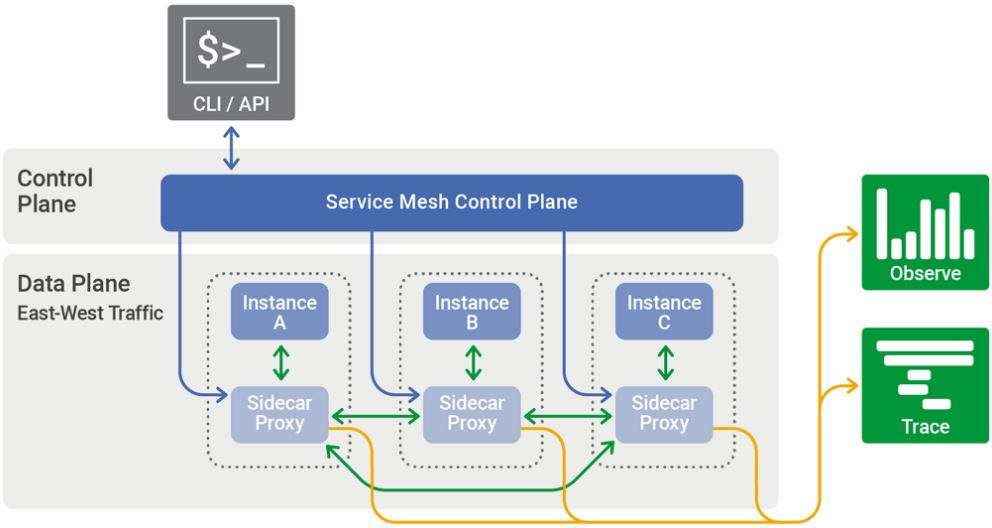

Figure 13 - Service Mesh. Source [Mesh]

B. Capabilities

Provides resilience through simplifying observability, traffic, security, and policy management

C. Type

External.

D. Consumers

- SOAPA layers

E. Pre-conditions to consume the service

A service mesh capability such as Istio or NATS must be provided

F. Interfaces

These interfaces are generic but reference is given to the NATS implementation for illustration purposes. Alternative implementations include Istio, Linkerd etc.

nSM-Discovery	**Detailed Description**	This interface allows the Resilmesh functions to discover where the other services are via a service registry
	From provider	SM
	To Consumer	Resilmesh Functions
	Technology	Rest API
	API Documentation	https://nats.io/tags/service-mesh/

		https://nats.io/blog/nats-to-implement-servicemesh-part1-service-discovery/
	Partners involved	TUS,

nSM_Security	**Detailed Description**	This interface allows the Resilmesh functions invoke security functions including end-to-end encryption (mutual TLS), authentication, authorization policies as well as service-to-service access control among the services.
	From provider	Service Mesh
	To Consumer	Resilmesh Functions
	Technology	REST API
	API Documentation	https://nats.io/tags/service-mesh/ https://nats.io/blog/nats-to-implement-service-mesh-functionality-part2-security/
	Partners involved	TUS

nSM_Observability	**Detailed Description**	This interface allows the Resilmesh to invoke observability such as metrics, tracing, and alerting functions
	From provider	Service Mesh
	To Consumer	Resilmesh Monitor
	Technology	REST API
	API Documentation	https://nats.io/tags/service-mesh/ https://dale-bingham-soteriasoftware.medium.com/using-nats-to-implement-service-mesh-functionality-part-3-metrics-tracing-alert-observability-f77cf5ab7db1
	Partners involved	TUS

nSM_Routing Control	**Detailed Description**	This interface allows the mesh to configure traffic shifting and mirroring
	From provider	Service Mesh
	To Consumer	Resilmesh Functions

	Technology	REST API and/or Fabric
	API Documentation	https://nats.io/tags/service-mesh/
	Partners involved	TUS

nSM_Loadbalancing	**Detailed Description**	This interface allows the service mesh to perform load balancing easily when 2 or more services are setup as replicas/copies
	From provider	Service Mesh
	To Consumer	Resilmesh Functions
	Technology	API REST
	API Documentation	https://nats.io/tags/service-mesh/
	Partners involved	TUS,

4.3 Event Aggregation (EA)

4.3.1 Function

This functional component ingests event logs from multiple sources and transforms the events via various operations (routing / logging / fusion / filtering / augmentation / reduction / monitoring) and then outputs the data to one or more destinations via the streaming layer. Aggregator transformation components may be linked in data pipelines giving rise to arbitrarily complex processing topologies. An example of such a pipeline is shown below for the *Vector.dev* aggregator.

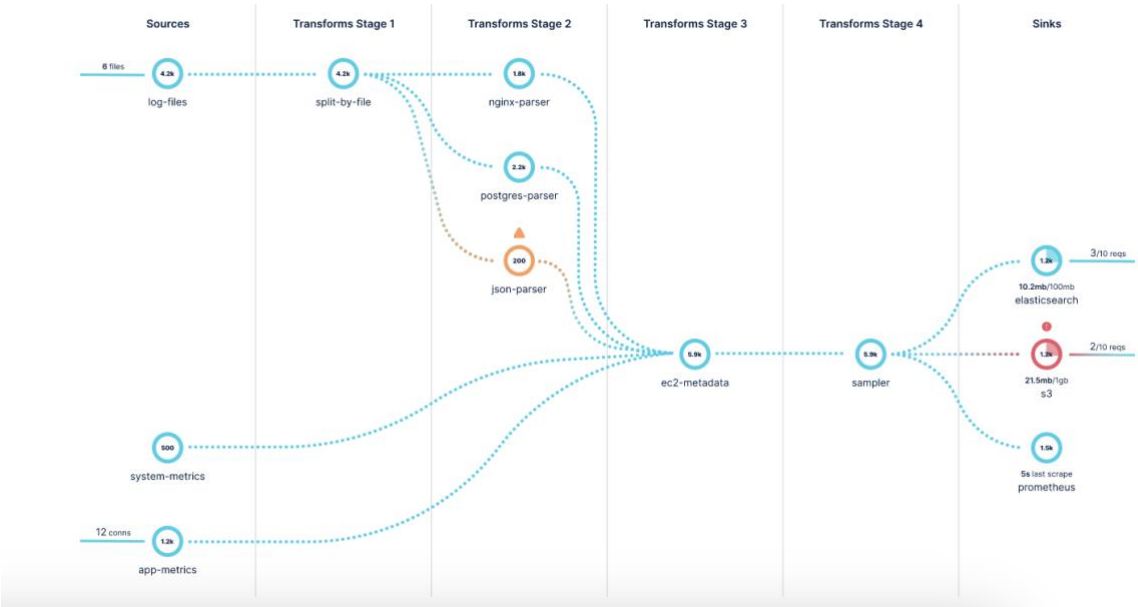

Figure 14 - Vector Orchestration

4.3.2 Provided services

The EA has three services as described below.

4.3.2.1 Event Aggregation

A. Description

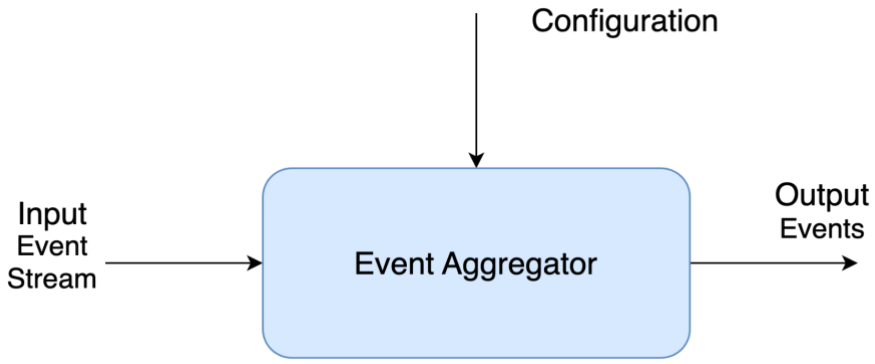

Figure 15 - Event Aggregation

B. Capabilities
Create
C. Type
Internal / External.
D. Consumers
- Decision engine

E. Pre-conditions to consume the service
Security(s) policies are defined.

F. Interfaces

nEA_Input	**Detailed Description**	This interface allows ingestion of events to the EA
	From provider	Log collection agents
	To Consumer	EA
	Technology	Rest API
	API Documentation	NA
	Partners involved	SLP

nEA_Output	**Detailed Description**	This interface allows output of events to the CEP
	From provider	EA
	To Consumer	DN. Other EA
	Technology	Rest API
	API Documentation	NA
	Partners involved	SLP

nEA_Configure	**Detailed Description**	This interface allows definition of EA pipeline configurations and addition of own transform /
	From provider	EA
	To Consumer	Resilmesh Applications

	Technology	Rest API or File
	API Documentation	NA
	Partners involved	SLP

4.4 Data Normalisation

4.4.1 Functions

Data Normalisation is the processing step used to map the events to a common format schema, it's the transformation of heterogeneous data coming from different sources into a single unified schema, so that they can be properly consumed by the different components of the platform, since they will know what to expect from the consumed data. For such, we will use ECS, which has several fields that can be used for the majority of use cases. Data Normalization can be also accomplished by SIEMs such as Wazuh, via decoders that transform the data to the SIEM data format.

4.4.2 Provided services

- **Standardisation**
  The events across the platform's pipeline will have a standard format schema

Capabilities
Common event schema
Type
External
Consumers
Threat Awareness Plane
Pre-conditions to consume the service
-
Interfaces

nRO_services	**Detailed Description**	this interface provides the standardisation of the events
	From provider	RO
	To Consumer	RO
	Technology	ECS
	API Documentation	https://www.elastic.co/guide/en/ecs/current/index.html
	Partners involved	SLP, GMV, TUS, UMU

4.5 Message Broker (MB)

4.5.1 Functions

Guarantee the correct flow of the events being processed by the Resilmesh platform by queueing them according to their state (normalised, enriched, etc). Essentially, every component of the platform will rely on the Message Broker for delivering their processed events, so that other components will do the same and so on, until the whole pipeline is finished and the event is ready for the Analytics Layer.

4.5.2 Provided services

Load Balancing
The Message Broker has the capability of reliably queueing events in such a manner that they can be consumed by any number of components. If the component is replicated, as in a Docker Compose or Kubernetes architecture, the events will be load balanced/distributed among these replicas.

Horizontal Scalability
The Message Broker is an essential piece of technology that aligns with the capability of horizontally scaling the components, since the events will be distributed among the consumers/subscribers, if the load of the platform increases, we increase the components replicas.

Capabilities
Leverages the resilience of the platform

Type
External

Consumers
Threat Awareness Plane

Pre-conditions to consume the service
NATS Clients, see: https://nats.io/download/#clients

Interfaces

nRO_services	**Detailed Description**	NATS.io implements the client-server paradigm, any client can connect to the server by passing the connection string in the format "nats://<IP>:<PORT>"
	From provider	any
	To Consumer	any
	Technology	NATS protocol over TCP
	API Documentation	https://docs.nats.io/
	Partners involved	SLP, GMV, TUS, UMU

4.6 Event Stream Processing (ESP)

4.6.1 Function

This component provides a capability for real-time processing of continuous data streams.

- *Stream processing* (SP) involves the real-time handling of data, where computation occurs directly as data is generated or received. Most data is produced incrementally over time as a sequence of events. In stream processing, applications maintain a constant presence for executing logic, performing analytics, and running queries, with data continuously passing through them. When an event is received from the stream, a stream processing application responds accordingly, potentially initiating an action, modifying an aggregate or statistic, or storing the event for future use.
- *Complex event processing (CEP)* is a generalisation of traditional stream processing for aggregating, processing, and analysing data streams in order to gain real-time insights from events as they occur. However whereas traditional stream processing is concerned with finding low-level patterns in data, such as the number of mouse clicks within a fifteen-minute window CEP can make high-level inferences about complex events within the business domain using models of causality and conceptual hierarchies. CEP is therefore suitable for tasks such as event correlation.

4.6.2 Provided services

4.6.2.1 Complex Event Processing

A. Description

This function aggregates a lot of different information that identifies and analyzes cause-and-effect relationships among events in real time by querying data before storing it within a database or, in some cases, without it ever being stored. Events can be received from different sources.

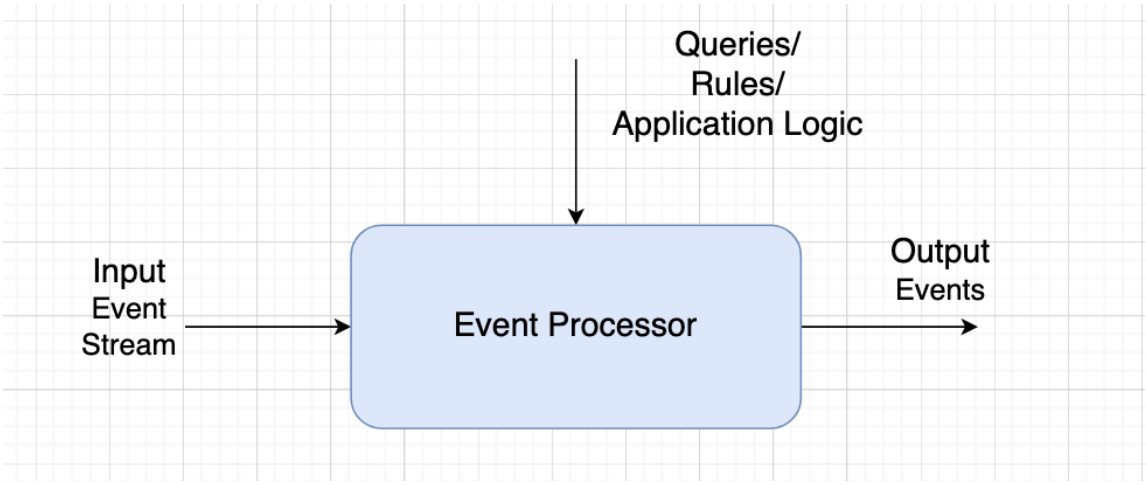

Figure 16 - Complex Event Processing

B. Capabilities

Correlation of Alerts to detect unusual patterns or events

C. Type

External

D. Consumers

Resilmesh applications

E. Preconditions to consume the service

F. Interfaces

nESP_CEPInput	**Detailed Description**	This interface allows ingestion of events to the CEP
	From provider	CEP
	To Consumer	Resilmesh Applications e.g. Elasticsearch and others
	Technology	Rest API
	API Documentation	NA
	Partners involved	TUS,

nESP_CEPOutput	**Detailed Description**	This interface allows output of events to the CEP
	From provider	CEP
	To Consumer	Resilmesh Applications e.g. Elasticsearch and others

	Technology	Rest API
	API Documentation	NA
	Partners involved	TUS

nESP_CEPConfigure	**Detailed Description**	This interface allows definition of CEP applications processing logic i,e, queries and rules
	From provider	CEP
	To Consumer	Resilmesh Applications
	Technology	Rest API
	API Documentation	NA
	Partners involved	TUS

4.6.2.2 Stream Processing

A. Description

The stream processing services enables the creation of individual 'stream processors' to operate on events in the stream and also allows the creation of a stream processor topology to carry out composite stream processing on the events. An example is Kafka streams that are built on top of the Kafka messaging broker. Events are ingested and output via the message broker API's.

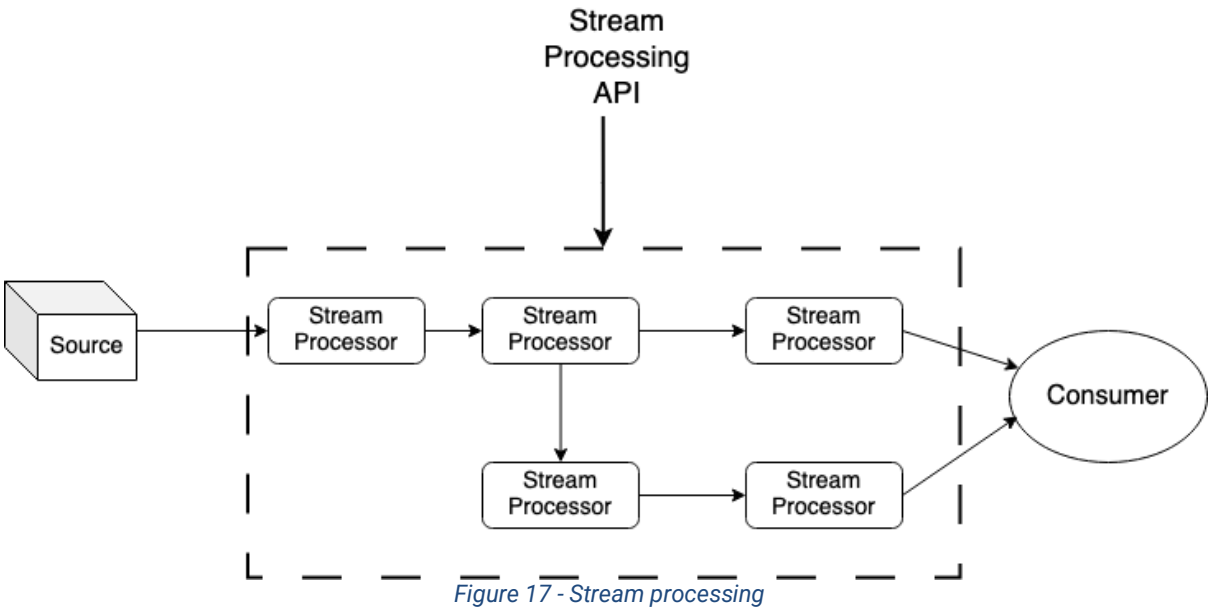

Figure 17 - Stream processing

B. Capabilities

Creates and deploys event stream processing applications

C. Type

Internal / External.

D. Consumers

- Decision engine

E. Pre-conditions to consume the service

Security(s) policies are defined.

F. Interfaces

nESP_Streamprocessc	**Detailed Description**	This interface allows the creation and deployment of stream processing applications
	From provider	Stream Processor
	To Consumer	Resilmesh Applications e.g. Elasticsearch and others
	Technology	Rest API
	API Documentation	https://kafka.apache.org/20/documentation/streams/developer-guide/dsl-api.html
	Partners involved	SLP,

4.7 Event Enrichment (EE)

4.7.1 Function

This function will enrich the events with contextual information from the Silent Push API. Silent Push scans, clusters, scores and enriches the global IPv4 range in a first-party database that outputs Indicators Of Future Attack (IOFA) – domain, IP and URL data that explains the relationship between billions of observable data points across the internet. The enriched data will make events more relevant through contextual information, depending on the type of the event (IP, Domain, etc) such as: whois, DNS records, ASN, Name Server, Certificates, Subnet information etc

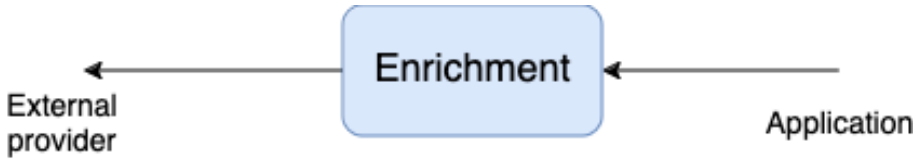

Figure 18 - Event Enrichment

4.7.2 Provided services

4.7.2.1 Event Enrichment

A. Description

This service enriches the incoming events with a callout to the Silent Push API.

B. Capabilities

enriches events to indicates possible suspicious IP/ASN/URL.

C. Type

Internal / External.

D. Consumers

- Various applications

E. Pre-conditions to consume the service

License agreements defined with SLP

F. Interfaces

nSLP-Enrich	**Detailed Description**	This interface from SLP enriches events with various risk and reputation scores to indicate possibly risky indicators of attack
	From provider	SLP
	To Consumer	Resilmesh Applications
	Technology	REST API
	API Documentation	https://docs.silentpush.com/
	Partners involved	SLP

nRCTI-Enrich	**Detailed Description**	This internal interface allows Resilmesh applications to call out enrich events.
	From provider	SLP
	To Consumer	Resilmesh Applications
	Technology	REST API
	API Documentation	NA
	Partners involved	SLP

4.8 Security Incident and Event Manager (SIEM)

4.8.1 Function

Security Information and Event Management (SIEM) operates within the realm of computer security, integrating software solutions and services that merge security

information management with security event management. SIEM serves as the central element in a standard Security Operations Center (SOC), which serves as the focal point for addressing security concerns across an organisation. It conducts immediate analysis of security alerts generated by both applications and network hardware. SIEM solutions are offered by vendors as software packages, appliances, or managed services, serving the dual purpose of logging security information and generating compliance reports.

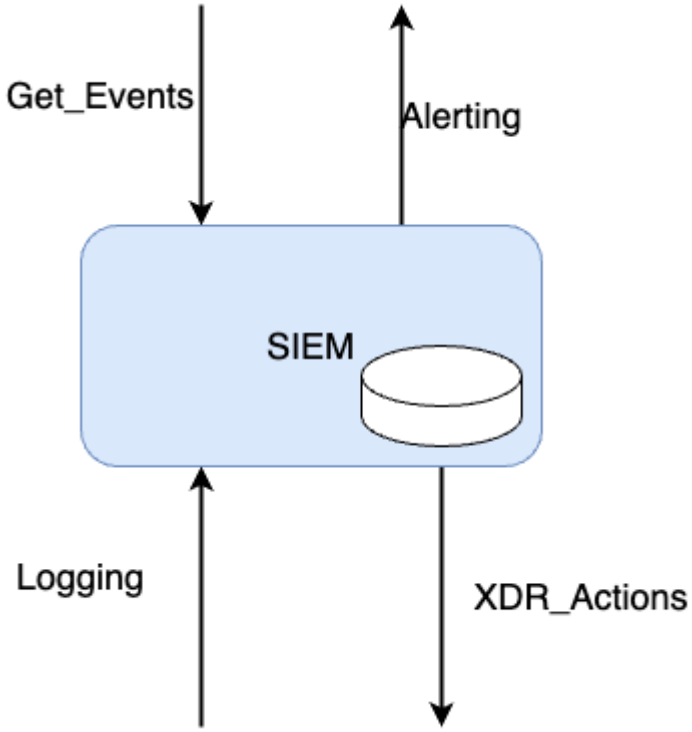

Figure 19 - SIEM Functional Architecture

The Resilmesh SIEM is based the the open source SIEM/XDR application, Wazuh[3].

4.8.2 Provided services

4.8.2.1 Event Logging

A. Description
Wazuh collects, analyses, and stores logs from endpoints, network devices, and applications. The Wazuh agent, running on a monitored endpoint collects and forwards system and application logs to the Wazuh server for analysis.

B. Capabilities
Security log analysis; Vulnerability Detection

C. Type
Internal

D. Consumers
Analytics applications

[3] https://wazuh.com/platform/overview/

E. Preconditions for service
Wazuh installed

F. Interfaces

nSIEM-LogCollection	**Detailed Description**	This interface ingests logs from the endpoints and stores on Wazuh or r Elastic
	From provider	End Points
	To Consumer	SIEM
	Technology	Rest API
	API Documentation	https://documentation.wazuh.com/current/getting-started/use-cases/log-analysis.html#log-data-collection
	Partners involved	TUS, SLP,

4.8.2.2 Event Correlation and Alerting

A. Description
Wazuh ruleset detects security events and anomalies in log data. These rules are written in a specific format and they trigger alerts when certain conditions are met. The rules are defined based on certain criteria like log fields, values, or patterns to match specific log entries that may indicate security threats. Wazuh provides a wide range of pre-built rules covering common security use cases. Additionally, administrators can create custom rules tailored to their specific environment and security requirements.

B. Capabilities
Intrusion detection

C. Type
Internal

D. Consumers
Alarm Dashboard (Kibana) ; Analytics functions; Mitigation Manager

E. Preconditions for service
Wazuh installed

F. Interfaces

nSIEM-Alerting	**Detailed Description**	This interface output alerts that indicate suspect or anomalous behaviour.
	From provider	SIEM

	To Consumer	Mitigation Manager, Alarm presentation in Elastic, Analytics functions
	Technology	Rest API
	API Documentation	https://documentation.wazuh.com/current/getting-started/use-cases/log-analysis.html#rules-and-decoders
	Partners involved	TUS, SLP,

4.8.2.3 XDR Actions

A. Description

Wazuh also provides an Extended Detection and Response (XDR) platform with a comprehensive security solution that detects, analyzes, and responds to threats across multiple IT infrastructure layers. Wazuh collects telemetry from endpoints, network devices, cloud workloads, third-party APIs, and other sources for unified security monitoring and protection.

B. Capabilities

Threat Hunting; File Integrity Monitoring; Behavioural Analysis; Endpoint mitigation actions.

C. Type

Internal / External.

D. Consumers

CASM;

E. Pre-conditions to consume the service

Wazuh installed.

F. Interfaces

nSIEM_XDR_Action	**Detailed Description**	This interface allows the WAZUH XDR manager to invoke detection and response on the endpoint.
	From provider	SIEM
	To Consumer	EDR agent
	Technology	REST API, or Fabric
	API Documentation	https://wazuh.com/platform/xdr/
	Partners involved	TUS, SLP

4.8.2.4 Get_Events

A. Description

Elasticsearch is a distributed search and analytics engine optimized for speed and relevance on production-scale workloads.

B. Capabilities
Event storage and retrieval
C. Type
Internal
D. Consumers
CASM;NDR THF, NSE,MM
E. Pre-conditions to consume the service
Elastic installed.

F. Interfaces

nSIEM_Get_Events	**Detailed Description**	This interface allows the applications such as NDr and NSE to retrieve events from SIEM logs
	From provider	SIEM
	To Consumer	CASM NDR THF, NSE, MM
	Technology	REST API, or Fabric
	API Documentation	https://github.com/elastic/elasticsearch
	Partners involved	TUS, All

4.9 AI-based detector (AID)

4.9.1 Function

The AI-based detector (AID) functional block is a system component for anomaly detection across a spectrum of IT and OT applications and network protocols. Leveraging machine learning (ML) and artificial intelligence (AI) techniques, the system offers anomaly detection models that can be deployed on endpoints, edges, or in the cloud based on specific requirements.

This component will include multi-view deep learning approaches, including fusion-based and alignment-based methods, to effectively handle the inherent heterogeneity present in mixed technology domains. For instance, a multi-view anomaly detector within Resilmesh may combine data from disparate sources such as an OT PLC (view), Modbus network data (view), and IT IDS data (view) to detect potential cross-node attacks.

The implementation extends to both centralised (*Figure 20*) and distributed, edge/endpoint-based anomaly detectors (*Figure 21*). The output generated by these detectors comprises events forwarded to the correlator/SIEM (Security Information and Event Management) or features transmitted to other models, ensuring a comprehensive and integrated approach to anomaly detection within diverse technological environments.

It has the following main features:

- Detect suspicious events and attacks at edge and potentially cloud

- Support multi-view anomaly detection for blending heterogeneous data sources, crucial for mixed IT/OT critical infrastructures
- Implement hierarchical feature fusion (e.g., Autoencoders) and decision fusion (e.g., ensemble learning) models for edge-based anomaly detection

4.9.2 Provided services

4.9.2.1 AI based anomaly detection in OT environment

A. Description

This service provides AI based anomaly detection.

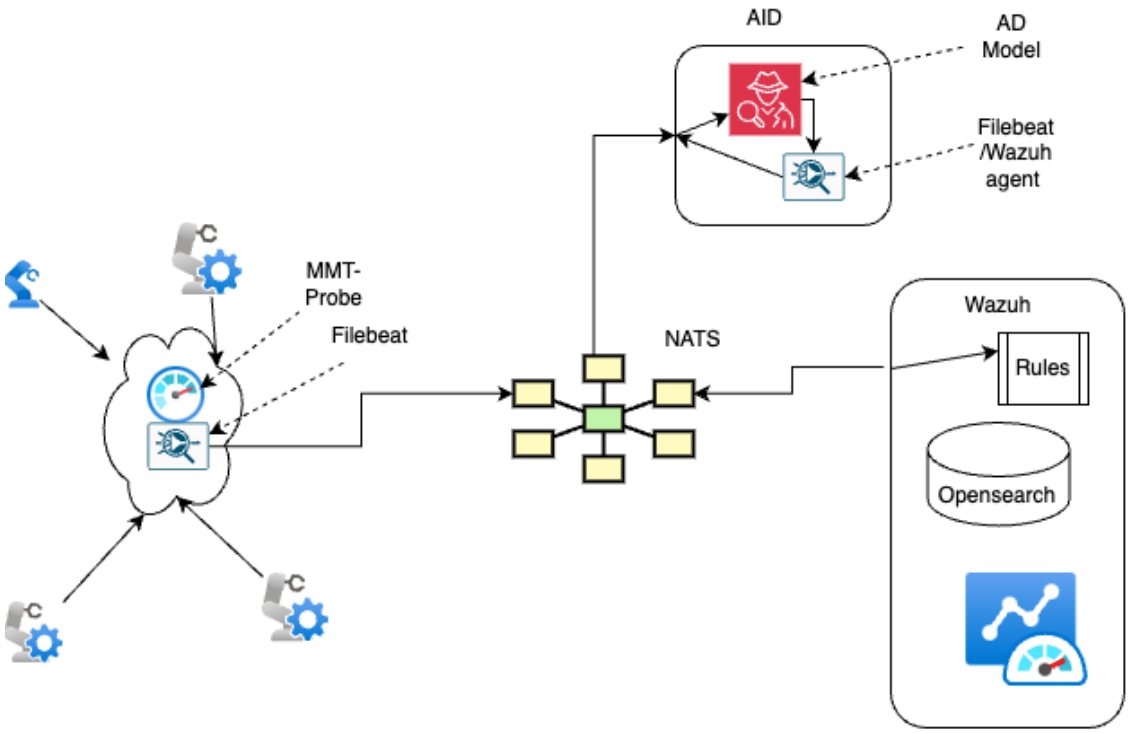

Figure 20 - AID Architecture centralized

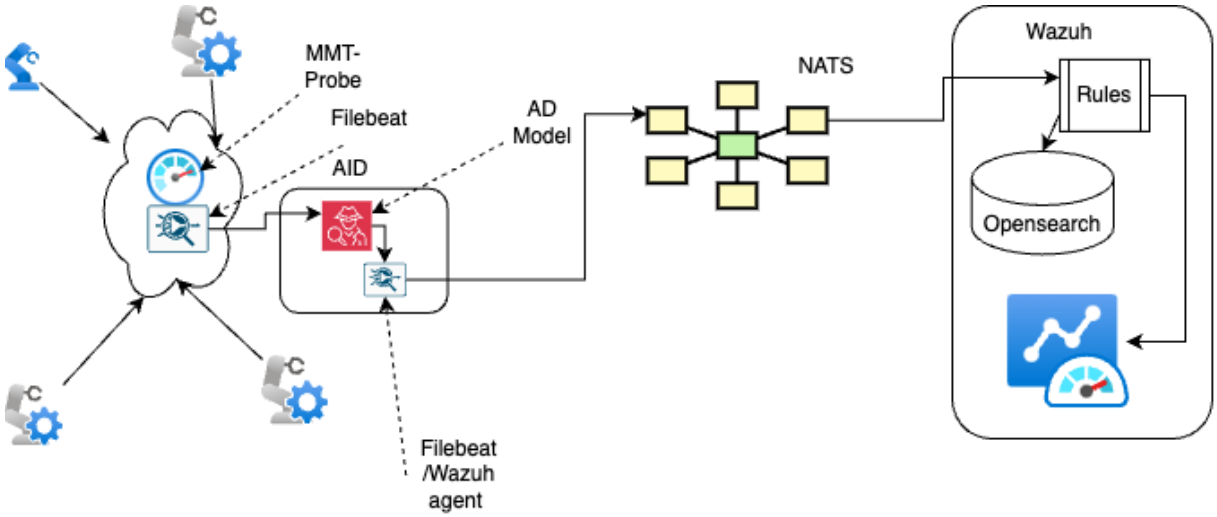

Figure 21 - AID Architecture distributed

B. Capabilities

Anomaly detection, multi-view data fusion.

C. Type

Internal.

D. Consumers

- AIC
- Applications

E. Pre-conditions to consume the service

Pre-processed and streamed features.

F. Interfaces

nAID_IngestFeatures	**Detailed Description**	This interface ingests pre-processed and streamed features.
	From provider	Event stream processor (ESP)
	To Consumer	AID
	Technology	REST API or Method call
	API Documentation	NA
	Partners involved	JR, MONT, UMU

nAID_DetectionEvent	**Detailed Description**	This interface outputs the result of the anomaly detection process in the form of security events and alerts.
	From provider	AID
	To Consumer	AIC, Application
	Technology	REST API, or Method call
	API Documentation	NA
	Partners involved	JR, TUS, MONT, UMU, Application Developer

4.9.2.2 Flow Processor

A. Description

This service, implemented by the UMU Monitoring Sensor and Flow Processor software modules, is used to monitor traffic from the network and calculate in real-time the flow features needed either for training the detection model through FL or to feed the detection model, once installed in the AI-based Detection Engine, with real-time data to perform the inference and alert about attacks taking place in the infrastructure.

The currently supported set of features and the description of each one can be consulted in the next table:

Table 5 - Features supported by the flow-processor subcomponent

F1	Uplink/source IP
F2	Downlink/destination IP
F3	Uplink/source port
F4	Downlink/destination port
F5	Duration in microseconds
F6	Number of packets sent in Uplink
F7	Number of packets sent in Downlink
F8	Ratio between Uplink and Downlink packets
F9	Number of packets sent in one second in Uplink
F10	Number of packets sent in one second in Downlink
F11	Number of bytes sent in Uplink
F12	Number of bytes sent in Downlink
F13	Number of bytes sent in one second in Uplink
F14	Number of bytes sent in one second in Downlink
F15	Maximal packet size in Uplink
F16	Maximal packet size in Downlink
F17	Minimal packet size in Uplink
F18	Minimal packet size in Downlink

F19	Average packet size in Uplink
F20	Average packet size in Downlink
F21	Variability of size between packets in Uplink
F22	Variability of size between packets in Downlink
F23	Maximal time-to-live (TTL) in Uplink
F24	Maximal time-to-live (TTL) in Downlink
F25	Minimal time-to-live (TTL) in Uplink
F26	Minimal time-to-live (TTL) in Downlink
F27	Average time-to-live (TTL) in Uplink
F28	Average time-to-live (TTL) in Downlink
F29	Variability of TTL between packets in Uplink
F30	Variability of TTL between packets in Downlink
F31	Average time between consecutive packets (IAT) in Uplink
F32	Average time between consecutive packets (IAT) in Downlink
F33	Min time between consecutive packets (IAT) in Uplink
F34	Min time between consecutive packets (IAT) in Downlink

B. **Capabilities**
Monitors and calculates the flow features needed for training and detection

C. **Type**
Internal.

D. **Consumers**
- FL Agents
- AI-based Detection Engine

E. **Pre-conditions to consume the service**
Generate traffic on the interfaces being monitored.

F. **Interfaces**
This service is internal and interfaces with the AI-based Detection Engine to send the flow features in real-time.

nAID_Send_Flow	**Detailed Description**	This interface is used to send the flow features in real-time
	From provider	AID
	To Consumer	PPFL
	Technology	NATS
	API Documentation	N/A
	Partners involved	UMU, JR

4.10 Privacy preserving model training (PPFL)

4.10.1 Function

The privacy preserving model training is responsible for training a AI-based detection and attack classification model using Federated Learning along with privacy-preserving algorithms based on data perturbation mechanisms and Privacy-Enhancing Technologies (PETs) to prevent membership inference and model poisoning attacks during the federated training process.

Once trained, the model is installed through a REST API message in the real-time AI-based Detection Engine, where it receives the traffic flows from the Event Stream Processor (ESP) module through NATS to perform inference, predict an attack type per flow, and output an alert per flow detected as an attack also to NATS.

For the Federated Learning training setting, several aggregation algorithms (e.g., FedAvg, FedProx) will be analysed and evaluated, as well as different privacy-preserving mechanisms (Differential Privacy based on noise-adding mechanisms such as Laplace or Gaussian-based), so that the tradeoff between accuracy and privacy is maximised.

The subcomponents of the assets, namely: FL Aggregator, FL Agent and the AI-based Detection Engine will be served as Virtual Network Functions (VNFs) so that they can be orchestrated (deployed and configured dynamically) based on policies, where specific configuration parameters, such as the number of training rounds to be executed or the aggregation algorithm to be used, can be specified.

For the first evaluation of the asset, the UMU Monitoring Sensor and Flow Processor will be used to monitor, process and ingest the traffic flow features needed for detection. Currently supported features include 34 metrics such as average packet size, minimum and maximum TTL values, variability of inter-arrival time (IAT), etc. These metrics are used to feed the FL Agents during the FL process to serve as training data, and during inference phase to feed the AI-based Detection Engine with flows to predict using the previously trained detection model.

4.10.2 Provided services

4.10.2.1 Privacy-preserving training

A. Description

The Federated Learning process workflow that would be used to train the anomaly detection model can be consulted in the following figure. As can be seen, a fixed number of training rounds are executed. During each round, the agents will train their local models upon local training data, apply a certain noise-adding mechanism, and share the resulting protected model updates with a central aggregator that will mix

them using a certain aggregation function (e.g., FedAvg). At the end of the final round, the final model is shared with the AI-based Detection Engine so it can be used for real-time anomaly detection.

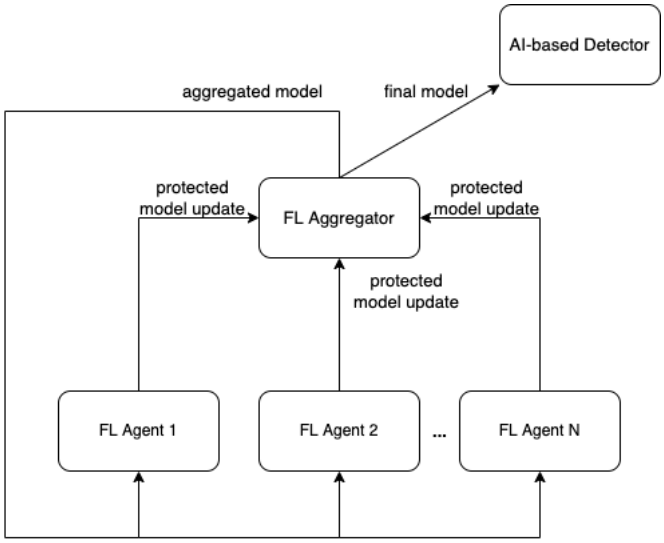

Figure 22- Privacy Preserving training service flow

B. **Capabilities**
Performs federated learning based on the deployed agents

C. **Type**
Internal.

D. **Consumers**
- AI-based Detector

E. **Pre-conditions to consume the service**
Provide training data for each agent in the Federated Learning scheme.

F. **Interfaces**
This service is internal to train the detection model and only provides one interface to send the model to the detection engine which is part of the same asset.

nPPFL_Install_Detection_Model	**Detailed Description**	This interface is used to send the trained detection model to the AI-based Detection Engine
	From provider	PPFL
	To Consumer	PPFL
	Technology	REST API
	API Documentation	NA
	Partners involved	UMU

4.10.2.2 Real-time detection

A. Description

Using the detection model trained in the last FL process, the asset provides this service to perform real-time detection from the network traffic flow it receives from the ESP or Flow Processor. For each flow detected as attack, it creates an alert in JSON format which contains:

1. The datetime when the attack was detected.
2. The source IP
3. The source port
4. The destination IP
5. The destination port
6. The attack type
7. The model's confidence of the prediction
8. The accuracy achieved by the model at testing time
9. The flow's features

These alerts are fed back to NATS using a separate subject to be consumed by mitigation agents (to enforce a certain mitigation to stop the attack) and further functional components in the RESILMESH architecture

B. Capabilities

Performs real-time detection and attack classification from real-time received flow features.

C. Type

Internal / External.

D. Consumers

- Mitigation agents
- AIC
- Applications

E. Pre-conditions to consume the service

Provide real-time flow features through the required NATS subject.

F. Interfaces

nPPFL_Perform_Inference	**Detailed Description**	This interface receives the real-time flow data for attack prediction
	From provider	Event stream processor (ESP) / Flow Processor
	To Consumer	PPFL
	Technology	NATS
	API Documentation	NA

	Partners involved	JR, MONT, UMU

nPPFL_Attack_Alert	**Detailed Description**	This interface sends the prediction (attack type) for each evaluated flow
	From provider	PPFL
	To Consumer	AIC, Application
	Technology	NATS
	API Documentation	NA
	Partners involved	JR, TUS, MONT, UMU, Application Developer

4.11 AI Correlation (AIC)

4.11.1 Function

Event correlation is the process of finding the relationships between events. Correlation creates context between individual events and information previously collected in real-time, and also normalises it for subsequent processing. The primary purpose of alert correlation is to identify the most significant events in the security dataset. Security event correlation should increase the quality of information about events while decreasing their number and interpreting multiple alarms.

The main directions for **application** of AI methods to correlate security events includes:

- classify security events for *event detection, event grouping,* and *event pattern extraction*
- *Intrusion detection* which deals with multi-stage and targeted attacks or *anomaly detection* to notify the security administrator about misuses and deviations from normal behaviour, respectively.
- I*ntrusion/attack projection* based on incoming events, which allows early detection of intruder targets.

There are three main areas of event correlation **methods**:

- *Similarity-based* methods are based on the idea that similar events can have the same root cause or the same type, and the found links depend on the inherent similarity between attributes of each event. Similar alerts are usually aggregated into a composite so-called *meta/hyper-alert*.
- *Causal-based* methods focus on the causal structure of an event sequence, when previous steps determine the ones that follow.
- *Data mining* is a process of discovering significant patterns, especially in a large amount of data

Different methods are appropriate for different applications.

Event correlation **AI-models** compromise the following approaches:

- *Rule-based correlation models* – similarity rules , causal rules , composite rules and rule mining models
- *Semantic correlation models* – signature language-based , event embedding and ontology learning models.
- *Graphical correlation models* – knowledge provenance graphs and probabilistic graphical models.
- *Machine learning correlation models* – shallow and deep learning models.

AIC can be decomposed to three sub functional components

- **AI Pruning (AIP)**- This component uses, primarily similarity based, methods to reduce the flow of information to be consumed e.g. to be presented to the SOC threat analysts.
- **AI-Root Cause Analysis (AI-RCA)** - This component uses primarily similarity based methods to help identify root cause of an anomaly, very often as part of a set RCA approaches.

AIC is most often included as a component in a higher level application.

AIC is complementary to non AI based rule based correlation.

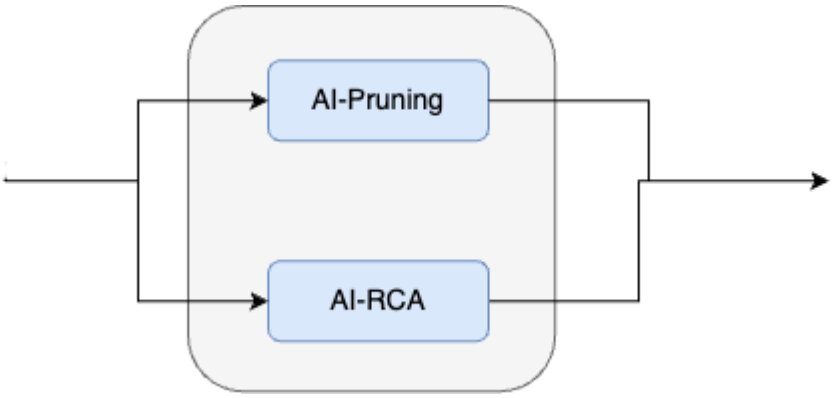

Figure 23 - AI Correlation

4.11.2 Provided services

4.11.2.1 AI Correlation

A. Description

This service provides generic AI correlation.

B. Capabilities

Alert grouping, prediction

C. Type

Internal.

D. Consumers

- Applications

E. Pre-conditions to consume the service

-

F. Interfaces

nAIC_IngestEvents	**Detailed Description**	This interface ingests security events and alerts to the correlation process.
	From provider	AIC
	To Consumer	Alert Stream
	Technology	REST API or Method call
	API Documentation	NA
	Partners involved	TUS, MONT,

nAIC_CorrelationResult	**Detailed Description**	This interface outputs the result of the correlation process. The output depends on the type of correlation method and AI approach used. Typically it will be a meta-alert of some type or an event prediction.
	From provider	AIC
	To Consumer	Application
	Technology	REST API, or Method call
	API Documentation	NA
	Partners involved	Application Developer

4.12 Threat Hunting and Forensics (THF)

THF provides functions for

- **TTP-based threat hunting**: THF supports the use of TTP-based hunting techniques for cyber attack investigation. It contains a number of sub-functions that support steps of the hunting process.
  - Manual Threat Hunting - This capability provides a manual based methodology implemented using Wazuh features including notebooks, dashboards, different forms of search as well as anomaly detection and other analysis features.
  - Enhanced Threat Hunting- This provides an LLM based conversational agent to automate parts of the manual methodology to provide a more friendly interface for threat hunter.
- **Robotic Forensics**: This capability provides an structured methodology to support forensic investigation of robotic systems by defining a set of

standardized steps to analyze key aspects of the robot and its operation, including log files, configuration integrity, and system behaviour. The methodology is triggered by relevant alerts and presented to the operator through the Resilmesh dashboard to guide the investigation process led by the user.

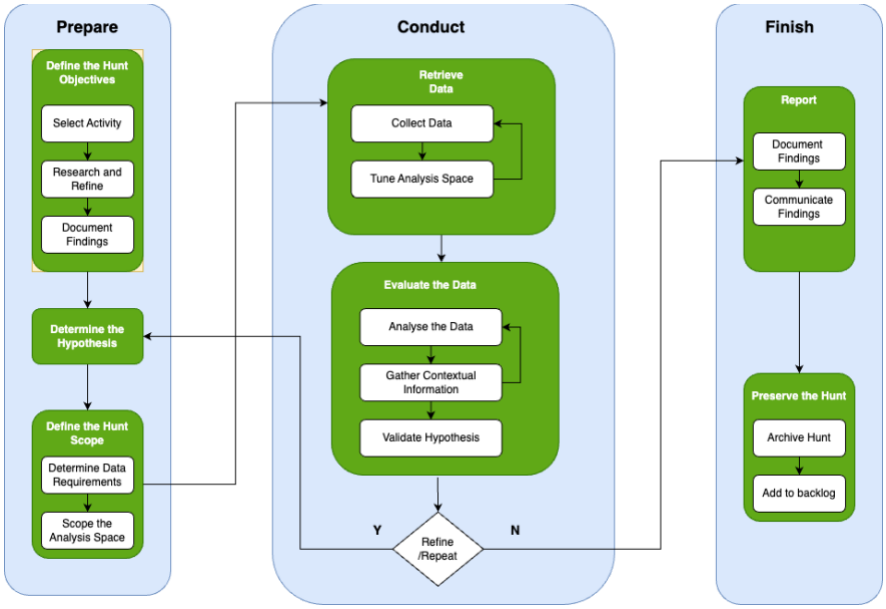

Figure 24 – Resilmesh Threat Hunting Methodology

4.12.1 Provided services

THF does not provide any API based services but does consume a number of other services

A. Interfaces

nISIM_AssetSearch	**Detailed Description**	This interface allows the CASM to retrieve asset details from ISIM
	From provider	THF
	To Consumer	Infrastructure and Services Information Model (ISIM)
	Technology	REST API,
	API Documentation	NA
	Partners involved	TUS, MUNI

nSIEM_Get_Events	**Detailed Description**	This interface allows the THF to retrieve alert and threat data from the SIEM (Wazuh)
	From provider	THF
	To Consumer	SIEM
	Technology	REST API
	API Documentation	https://documentation.wazuh.com/current/user-manual/indexer-api/index.html
	Partners involved	TUS

nRCTI_ECS2STIX	**Detailed Description**	This interface allows the THF to convert behavioural observations to STIX format
	From provider	THF
	To Consumer	SIEM
	Technology	REST API
	API Documentation	NA
	Partners involved	TUS

4.13 Robust Cyber Threat Intelligence (RCTI)

4.13.1 Function

This function contains two sub-functions

- **CTI sharing** internally and externally. This is implemented in the first instance using MISP and may also be extended to sharing via STIX2
- **Indicator of Behaviour (IoB) creation** – IoB are activities or events that indicate possible anomalous behaviour and are seen as the next step in improving threat detection and response. These could include anomalies in network traffic, unexpected file modifications, or irregular user activities. This function builds on work carried out in the Open Cyber Security Alliance (OCA) that aims to create a standardised approach for representing cyber threat actor behaviours in a shareable format. focus on patterns of behaviour associated with malicious cyber activity. By understanding the behaviour patterns innovative solutions can

be developed to enable shared behaviour sets.

- **Privacy Preserving CTI** – When adding new internal events to the CTI sharing platform, this component allows the enforcement of privacy policies, which ensure that sensitive information is anonymised in order to prevent data exfiltration. Privacy policy files are designed to address specific data attributes with granular privacy preserving techniques, and they can be switched on the fly based on organizational needs.

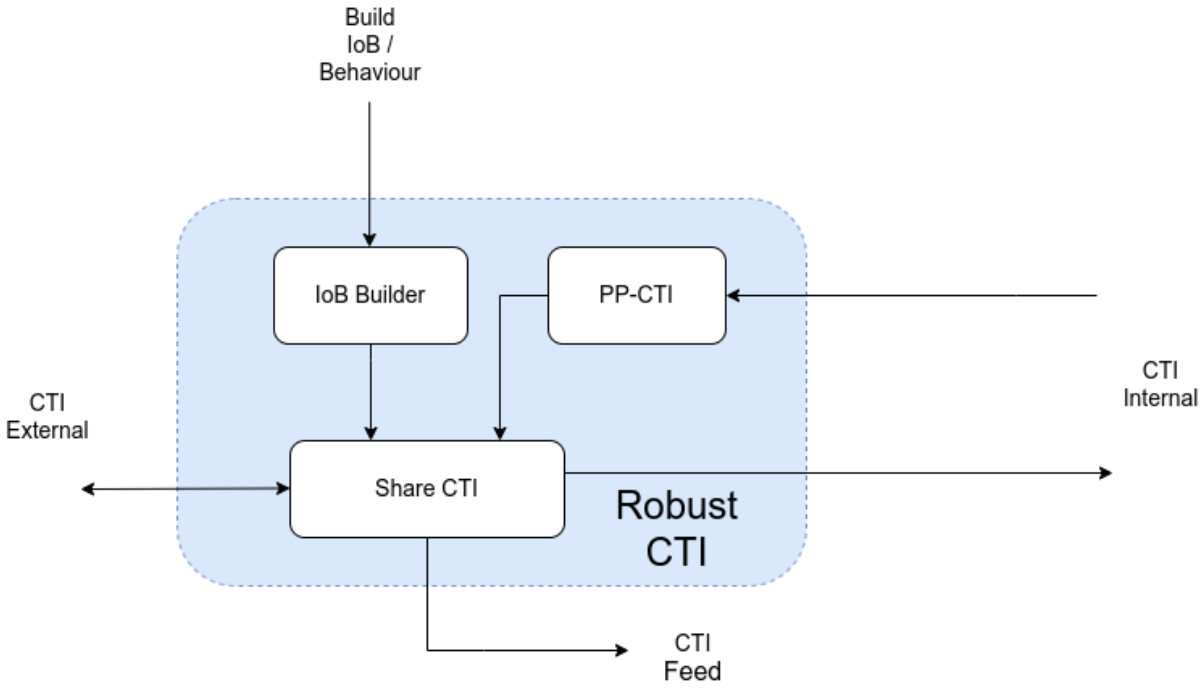

Figure 25 - Robust CTI Architecture

4.13.2 Provided services

4.13.2.1 CTI Sharing

A. Description

This service shares CTI internally and externally

B. Capabilities

store and share CTI

C. Type

Internal / External.

D. Consumers

- Various applications

E. Pre-conditions to consume the service

Sharing agreements defined with external partners.

F. Interfaces

nRCTI-ExtCTI	**Detailed Description**	This interface allows Resilmesh to share CTi with external organisations via MISp (or other CTI platform). Appropriate security arrangements must be defined. This is implemented in the first place for MISP
	From provider	External CTI ProviderI
	To Consumer	RCTI
	Technology	Rest API
	API Documentation	https://www.circl.lu/doc/misp/automation/ ,
	Partners involved	SLP

nRCTI-IntCTI	**Detailed Description**	This interface allows Resilmesh applications to pull CTI . This is implemented in the first place for MISP using PyMISP
	From provider	RCTI
	To Consumer	Resilmesh Application
	Technology	Rest API
	API Documentation	https://www.circl.lu/doc/misp/pymisp/
	Partners involved	TUS

nRCTI-IntFeed	**Detailed Description**	This interface enables local feeds into MISP. Feeds are remote or local resources containing indicators that can be automatically imported into MISP at regular intervals
	From provider	Resilmesh Applications
	To Consumer	RCTI
	Technology	REST API
	API Documentation	https://www.circl.lu/doc/misp/managing-feeds/
	Partners involved	TUS

4.13.2.2 IoB construction

A. Description

This service allows the construction of STIX2.1 objects to describe a chain of suspected adversarial behaviours.

B. Capabilities

generate STIX IoB objects

C. Type

Internal / External.

D. Consumers

- THF

E. Pre-conditions to consume the service

-

F. Interfaces

nSLP-Enrich	**Detailed Description**	This interface enables the construction of STIX IoB behavioural object sets.
	From provider	RCTI
	To Consumer	THF
	Technology	REST API
	API Documentation	https://github.com/opencybersecurityalliance/oca-iob
	Partners involved	TUS

4.13.2.3 Privacy Preserving CTI

A. Description

This service extends the CRTI component with data anonymisation capabilities, for externally published CTI.

B. Capabilities

Generate privacy policies, anonymize CTI events.

C. Type

Internal / External.

D. Consumers

- External CTI Sharing platforms

E. Pre-conditions to consume the service

-

F. Interfaces

nRCTI_IntPPCTI	**Detailed Description**	This interface enables pushing CTI events to the CTI Sharing platform, preprocessing them according to specific privacy policies

	From provider	Resilmesh Application
	To Consumer	RCTI
	Technology	REST API
	API Documentation	N/A
	Partners involved	TUS, UMU

4.14 Infrastructure and Service Information Model (ISIM)

4.14.1 Function

The information model captures and represents all the entities of interest in the environment. The environment consists of many components of various types, which have to be reflected. The information model interconnects the pieces of information on the assets in the environment.

The information model is to be materialised in a database that will serve as a data repository for other components of T5.1 (e.g., CASM, CSA, NSE). The other components will have access to the database for reading and editing. However, there is also a need to fill the database with data from external sources, namely when deploying the overall system in a new environment. Thus, the component needs to implement a connector to an external service or repository and store the data about the infrastructure to ISIM in bulk.

Following the Cyber Defense Matrix [CDM], all assets of the following categories are considered:

- Devices,
- Networks,
- Applications,
- Data,
- Users.

For each category of assets, their enumeration will be collected from existing databases, repositories, or services, or collected via a set of custom tools. Data collection is split between asset management tasks (ISIM) and attack surface discovery tasks (CASM). ISIM collects the data on assets, primarily from asset inventories like the NetBox tool [NetBox]. Widely used asset management systems and asset descriptions in non-IT domains (e.g., IoT, OT) will be provided by partners during the development.

CRUSOE data model [Komarkova2018] served as a foundation for this component, although it was updated to better fit the Resilmesh use cases. CRUSOE was intended to store data coming from network monitoring tools. The list of changes goes as follows:

- Devices - referred to as *Node* and *IP* in CRUSOE, can be merged into a *Device* entity,
- Networks - CRUSOE nowadays only considers the hierarchy or subnets, but not separate networks, the change will be trivial in this regard,
- Applications - CRUSOE models only network services associated with the IP, which will be generalised,
- Data - not modelled in CRUSOE, adding them and connecting them to devices would not be complicated,
- Users - already modelled in CRUSOE, although not yet used in practice.

Moreover, ISIM stores information about critical services and business processes/missions and their mapping to assets to keep track of which assets support each critical service or missions. These mappings will be provided by CSA and, if possible, stored alongside other data in the ISIM database. In case this is not feasible, there will be a separate repository of mappings in the CSA component. Models used in the CRUSOE Decide component will be used for start.

ISIM should also support risk assessment accomplished by other components. For this purpose, the data model will allow computing centralities for nodes in the network. These data are accessible by other components, e.g, CSA and NSE, using a REST API and a GraphQL API.

Another important aspect in risk assessment is the representation of vulnerabilities in network environments. The data model should be used for vulnerabilities from the National Vulnerability Database (NVD) [NVD], which implies support for multiple versions of the Common Vulnerability Scoring System (CVSS) that are used in the NVD – versions 2.0, 3.x, and 4.0 [Metrics]. The data about vulnerabilities will be collected by CASM and stored in the database.

4.14.2 Provided services

Three provided services are assumed: 1) DB that stores the data according to the information model, 2) data collectors or connectors to external systems, and 3) API for the use by NSE and CSA.

4.14.2.1 Information model and DB

A. Description

The component consists of two parts - database and a data model. The database is a functional component of Resilmesh, the data model defines the structure of the data in it and provides a common language for this and other components of WP5.

To start with, the CRUSOE data model and a graph database (preferably Neo4J) will be used.

B. Capabilities

1) Persistent storage of data about assets in the environment.
2) Common language for describing and categorising the assets.

C. Type
Internal.

D. Consumers
- NSE
- CSA

E. Pre-conditions to consume the service
ISIM and CSA data models are defined.

F. Interfaces
Only an admin interface of the database for its management and/or orchestration.

4.14.2.2 Data collector / connector

A. Description
A data collector or connector capable of filling the ISIM database with data from external sources. There might be multiple connectors for multiple types of data sources.

Primarily, there will be an option to insert a bulk of asset descriptions in CSV or JSON format directly - for all asset types and domains. The structure of such CSV/JSON is to be defined.

Optionally, for selected asset types and scenarios, we may use existing data sources and directly write connectors that extract data from them. For example, a widely used asset management system in IT is Netbox - an ISIM-Netbox connector would get the relevant data from Netbox (list of assets - devices, networks, …) and store them in ISIM DB. Similar connectors could be implemented to get lists of assets of other types (users, data, …) or in other domains (IoT, OT). Such a connector is run manually or optionally scheduled to update the list of assets.

If no such service exists or is not considered for an asset type or domain, ISIM falls back to textual input described above.

B. Capabilities
Reading the description of the environment from an external source (service, repository) and storing it into the ISIM database, in bulks of multiple assets of the same or several types.

C. Type
Internal / External.

D. Consumers
- ISIM

E. Pre-conditions to consume the service
External data sources are up and running or the data on assets are available elsewhere, e.g., in a CSV or JSON file. ISIM DB is up and running.

F. Interfaces
Just input.

4.14.2.3 ISIM API

A. Description

API that allows the user/NSE/CSA to access the contents of the ISIM DB. The API implements queries expected from NSE/CSA and ensures their correct execution, conforming to the data model, etc. The API can also be used to send custom queries to the database.

B. Capabilities

Translating queries from NSE/user into valid ISIM DB queries and ensuring their proper execution. List of queries is to be specified by NSE/CSA.

C. Type

Internal

D. Consumers

- NSE
- CSA
- MM
- CASM
- other applications that may be added e.g. Open Calls.

E. Pre-conditions to consume the service

ISIM DB is up and running

F. Interfaces

There is one interface that uses two technologies – REST API and GraphQL interface – that allow other components of WP5 to access ISIM data. The difference in the use of REST API and GraphQL API is that the REST API should provide predefined endpoints that are needed by other components. On the contrary, the GraphQL allows preparing any custom queries according to GraphQL schema by developers of other components. Such queries could be prepared also in run time and then passed to ISIM.

nISIM_AssetSearch	**Detailed Description**	This interface will allow other components in WP5 (namely NSE and CSA, in some cases even the user directly) to view and manipulate the content of ISIM.
	From provider	ISIM
	To Consumer	NSE, CSA, MM
	Technology	Rest API or GraphQL
	API Documentation	REST API: https://github.com/resilmesh2/ISIM/blob/main/docs/api_reference.yaml

		GraqpQL: https://github.com/resilmesh2/ISIM/blob/main/isim_graphql/src/schema.graphql
	Partners involved	MUNI, TUS

4.15 Cyber Asset Attack Surface Management (CASM)

4.15.1 Function

The Cyber Asset Attack Surface Management (CASM) provides contextual awareness about all the assets in the organisation. It has the following main features;

1. It provides status and posture information for every asset – internal and external - in the enterprise across all technology types and it continuously monitors assets for any change of status to identify risky services running the organisations network for remediation and attack surface area reduction.
2. It enables enterprise security teams to check how assets comply with the organisation security policy i.e., it can be used to detect deviations from policy by examining relationships between objects via queries to the underlying database. Security teams can ask questions such as "Which users do not have multi factor access enabled on AWS?" or "Show me which devices do not have virus checkers installed" or "Show me all assets with highest vulnerabilities". Queries may be saved and shared.
3. It provides a capability to take action via alerting or enforcement. Users may e.g. open a trouble ticket to remedy some problem or send an email or deploy security controls etc. Actions may be manually or automatically invoked.

The function works for both IT and OT assets. It enables users to add a customised dashboard to manage their own activities.

It has the following subfunctions:

External ASM (EASM)

This function is used to map the external attack surface of an organisation i.e. it identifies and manages threats discovered in internet facing assets using independent scans of the organisation attack surface. This includes aspects of so-called 'shadow IT' which is defined as "all hardware, software or any other solutions used by employees inside the organisation which have not been approved by the IT department". EASM discovers and enumerates internet facing assets using a number of DNS enumeration techniques such as

- Brute-force of subdomains using a domain name wordlists and alteration wordlists
- Identify subdomains by reading SSL/TLS certificates, performing DNS zone transfers or checking certificate transparency logs

- Recursive subdomain discovery on identified domains

The enumeration results are stored in ISIM and the tool provides continuous monitoring by performing regular repeated enumerations and comparing and highlighting differences between enumerations. It can calculate and highlight risks associated with different assets and initiate remedial actions including decommissioning or isolating assets that don't need to be Internet facing.

Internal ASM

This function provides visibility to the organisation's internal assets that are collected in the ISIM database. It leverages the ISIM graph query language to enable security personnel to check asset status as well as relationships between assets. It will:

- discover exposures including vulnerabilities, expired certificates, etc.
- ensure compliance with regulations,
- continuously monitor all assets,
- notify security teams of changes or omissions,
- trigger actions to isolate or remediate exposures and vulnerabilities.

Alerting

This function triggers alerts and mitigation actions in response to input from ASM modules

UI (SACD)

This function provides the main dashboard and user interface functionality. It is referred to as SACD (Situation Awareness Consolidated Dashboard) in other documents. This will entail the use of (sub) function specific visualisations and UI screen as well as use of the ISIM console and visualisation functions. It provides the following services to the operator/analyst via the UI:

- discover and visualise the internal and Internet facing assets,
- enumerate the discovered Internet facing assets to uncover vulnerabilities and estimate risk,
- discover the status, compliance and risk situation of internal assets,
- initiate automated continuous monitoring of both internal and Internet facing assets,
- trigger report or remediating actions.

4.15.2 Provided services

4.15.2.1 Attack Surface Management

This service provides internal and attack surface management scans and alerting. It is triggered either by a schedule or on demand by an Operator.

A. Description

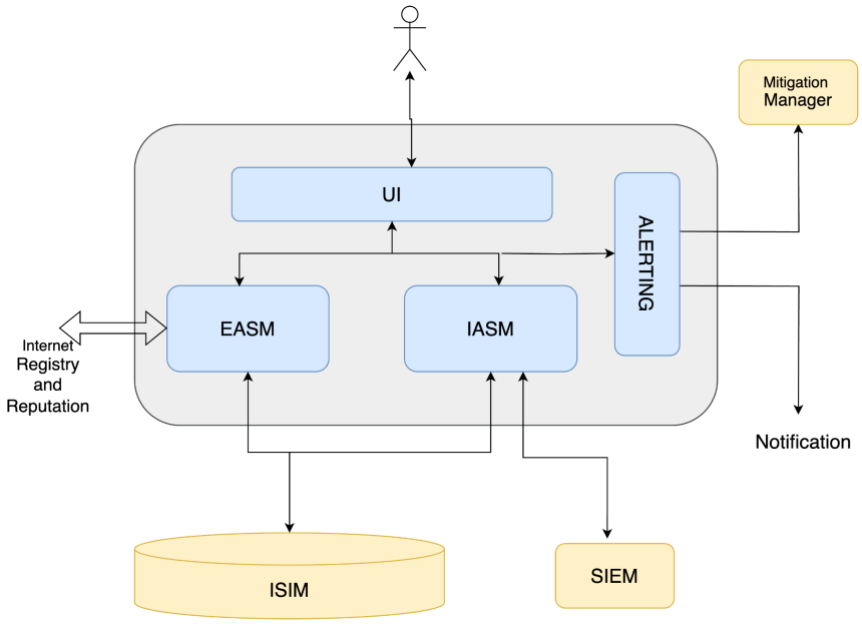

Figure 26 - CASM Architecture

B. Capabilities

Scans internal and external assets and triggers alerts

C. Type

Internal and External.

D. Consumers

Mitigation Manager; Other services, e.g., email

E. Pre-conditions to consume the service

Asset connectors exist and external interfaces keys obtained

F. Interfaces

nCASM-ExtScan	**Detailed Description**	This interface enables the EASM to scan the Internet to discover the organisations external facing services and ports and to discover shadow IT
	From provider	CASM
	To Consumer	Internet Registry and Reputation Services
	Technology	Rest API
	API Documentation	NA
	Partners involved	TUS, MUNI

nISIM_AssetSearch	**Detailed Description**	This interface allows the CASM to retrieve asset details from ISIM
	From provider	CASM
	To Consumer	ISIM
	Technology	REST API
	API Documentation	REST API: https://github.com/resilmesh2/ISIM/blob/main/docs/api_reference.yaml GraqpQL: https://github.com/resilmesh2/ISIM/blob/main/isim_graphql/src/schema.graphql
	Partners involved	TUS, MUNI

nSIEM_XDRaction	**Detailed Description**	This interface allows the CASM to check assets for software vulnerabilities and/or invoke other XDR functions such as file integrity checking etc
	From provider	CASM
	To Consumer	SIEM
	Technology	REST API
	API Documentation	https://wazuh.com/platform/xdr/
	Partners involved	TUS

nMISC_Notify	**Detailed Description**	This interface allows the orchestrator to notify SOC operators and other relevant personnel of alerting events.
	From provider	CASM
	To Consumer	Email, Slack and other such providers.
	Technology	API REST
	API Documentation	NA
	Partners involved	TUS

4.16 Critical Service Awareness / Mission Awareness (CSA)

4.16.1 Function

The critical service awareness component will provide hierarchical risk assessment to aggregate infrastructure risk into a risk for the critical service or business mission (CS/M). It will implement tools to assess, visualise, and manage such risks. To achieve this, the component will use data stored in ISIM (information model) and forward the outputs towards NSE (Network Situation Evaluation).

The overall functionality of CSA goes as follows:

1. Information on the assets in cyber infrastructure are stored in ISIM.
2. User defines which assets support which critical service or business mission and relations are there between the assets, services, and missions. The definition is stored in ISIM (or a separate CSA repository, if needed).
3. On demand or periodically, the component will (for each critical service/mission):
    a. Iterate the assets (supporting the service/mission) and risks associated with them,
    b. hierarchically assess the risks by aggregating them bottom-up,
    c. return the overall risk to each critical service or mission,
    d. (optional) save the risks into time series for the use by NSE.
4. The outputs will be used as follows:
    a. formatted for visualisation in the NSE,
    b. critical services and missions will be sorted by the risk level,
    c. (optional) key factors contributing to the overall risk scores are highlighted (root cause analysis - perhaps leave to NSE),
    d. recommendations of tasks to manage the risks will be provided.

CSA will allow the user to check whether a risk to an asset impacts a critical service or mission and which risks impact each critical service or mission. More precisely, it should allow the user to see which risks are the most severe with regard to CS/M.

CSA will rely on the data stored in ISIM, namely the enumeration of assets. However, it will also build a mapping between assets and CS/M. The data will be stored either in ISIM or in a separate repository of CSA, depending on practical issues. The outline of the mapping can be seen to the right of this text. Critical services (“Service”) is mapped on assets (in red) via AND/OR notation. The same notation is used to map services and missions (“Supportive Process”).

4.16.2 Provided services

Two services need to be provided: 1) management of the repository of mappings and 2) risk assessment.

4.16.2.1 Management of asset-CS/M mappings repository

A. Description

CRUD interface that will allow for accessing and manipulating a repository of mappings.

B. Capabilities

Insert a new mapping

User prepares a JSON-formatted (preferably) document describing the mapping of CS/M to assets. The CSA will receive the JSON, validate it and check if all the assets are defined in ISIM. Then, it will check if the CS/M is already defined and either inserts or updates it.

Permanently store mappings

Resolved by ISIM, unless not technically feasible - then a separate DB will be created.

Read one or more mappings

JSON-formatted descriptions of one or more mappings are returned on demand.

Modify or delete a mapping

For each modification, a validity check will be performed - all the assets need to be defined beforehand.

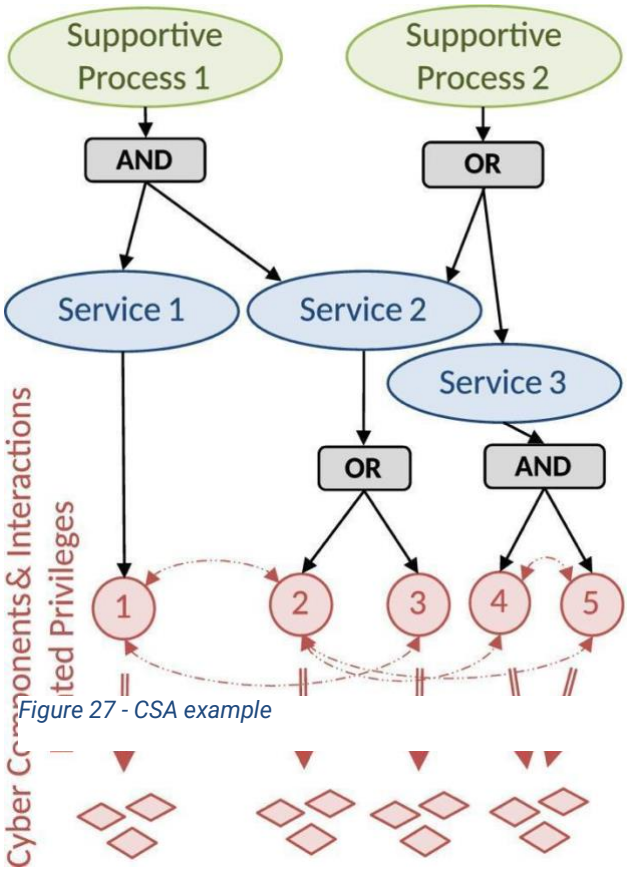

Figure 27 - CSA example

C. Type

Internal

D. Consumers

- ISIM (Information model)
- NSE (Network Situation Evaluation)

E. Pre-conditions to consume the service

ISIM and NSE are up and running, although it should be usable even without NSE (via text-based API)

F. Interfaces

nCSA_AssessSituation	Detailed Description	This interface will allow the user (via the dashboard) to read, insert, or manipulate CI/mission descriptions

	From provider	CSA
	To Consumer	ISIM
	Technology	Rest API or GrapQL
	API Documentation	NA
	Partners involved	MUNI

4.16.2.2 CS/M assessment

A. Description

The initial version will use the algorithm described in the research paper by Javorník and Husák [Javorník2022]. For example, let's assume a mission is supported by two critical services, each supported by an application running on a device. Applications or devices are found to be vulnerable - risks based on CVSS scores of the vulnerability are assigned to them. Risks to the applications and devices are then propagated upwards to the critical service - maximal, average, or otherwise calculated risk scores are estimated. Then, the same is done with missions - risk score is aggregated from the scores assigned to critical services.

Nevertheless, a more elaborated approach inspired by other existing solutions or operational needs of Resilmesh will be elaborated.

B. Capabilities

Assessing the risk for CS/M - each CS/M is assigned a risk score based on its underlying assets.

C. Type

Internal

D. Consumers

- ISIM (Information model)
- NSE (Network Situation Evaluation)

E. Pre-conditions to consume the service

ISIM and NSE are up and running.

F. Interfaces

nCSA_AssessSituation	**Detailed Description**	This interface will allow the user (via the dashboard) or orchestrator (for periodically triggered assessments) to run the risk assessment routine and output its results.
	From provider	CSA
	To Consumer	NSE, ISIM
	Technology	Rest API or GraphQL

Funded by the European Union

	API Documentation	NA
	Partners involved	MUNI, TUS

4.17 Network Detection and Response (NDR)

4.17.1 Function

Network Detection and Response (NDR) leverages both signature-based and non-signature-based analytical methods, such as machine learning, to identify suspicious network activities. NDR solutions continuously monitor and analyse raw enterprise network traffic to establish a baseline of normal behaviour. Any deviations from this baseline are flagged as potentially threatening, prompting alerts for security teams to investigate and respond to potential threats within their environment.

It contains the following capabilities:

1. **Traffic Analysis**: Analyse network traffic patterns to identify normal behaviour and potential anomalies. This includes monitoring bandwidth usage, protocols, and communication patterns.
2. **User Behavior Analysis:** Analyse user activities on the network to detect anomalous behaviour. This can include identifying suspicious login patterns, unauthorized access attempts, or unusual data access patterns.
3. **Anomaly Detection:** Implement algorithms and techniques to detect anomalies in network behavior. This can include unusual patterns of traffic, unexpected connections, or deviations from established baselines.
4. **Root Cause Analysis (RCA):** RCA aims to identify the root causes of incidents. By understanding the fundamental issues, SOC analysts can address the core problems rather than dealing with symptoms (cascading effect). This can reduce recurring incidents
5. **Explainable AI (XAI):** XAI contributes to the Anomaly Detection model transparency as it provides insights into the model's decision-making process. (aka, explainable when a particular instance is flagged as anomalous). It also helps highlighting contributing features (Feature Importance)
6. **Reporting and Visualization:** Generate reports and visualizations to present network situational awareness information to stakeholders. This can include dashboards, graphs, and charts to communicate key insights.

NDR will build on anomaly detection and root cause correlation functions described elsewhere in this document. These will be included directly as subcomponents/analysis modules and will not be consumed as services.

The NDR tool is developed based on the MMT toolset from Montimage [4]. An representative architecture of the MMT-based NDR is shown in the diagram below -

[4] https://github.com/montimage

the final version may deviate somewhat but will closely follow this structure

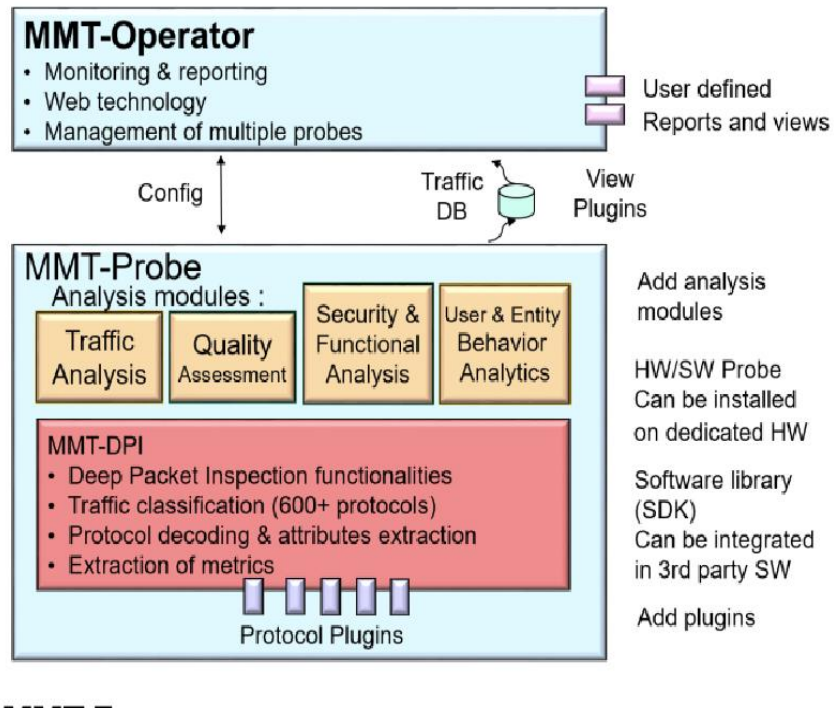

Figure 28 - NDR based on MMT

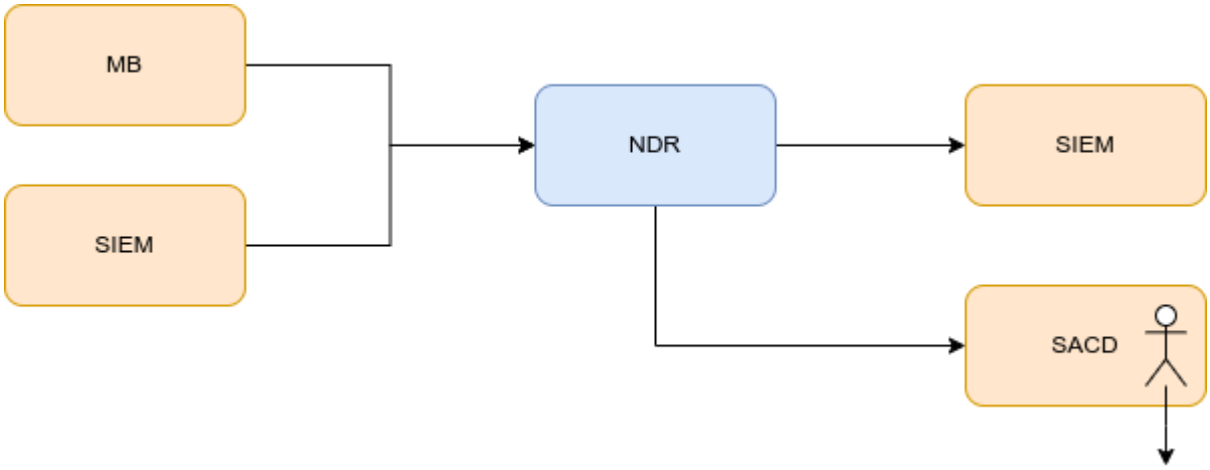

Figure 29 - NDR Functional Architecture

4.17.2 Provided services

NDR does not provide services to other components but does consume services from other components

A. Interfaces

nSIEM-Alerting	**Detailed Description**	This interface outputs alerts that indicate suspect or anomalous behaviour.
	From provider	NDR
	To Consumer	SIEM
	Technology	Rest API
	API Documentation	https://documentation.wazuh.com/current/getting-started/use-cases/log-analysis.html#rules-and-decoders
	Partners involved	TUS, MONT

nSIEM_Get_Events	**Detailed Description**	This interface allows the applications such as NDR to retrieve events from SIEM logs
	From provider	SIEM
	To Consumer	NDR
	Technology	REST API,
	API Documentation	https://github.com/elastic/elasticsearch
	Partners involved	TUS, All

nMB_Get_Data	**Detailed Description**	This interface allows the NDR component to retrieve data in real time from MB
	From provider	MB
	To Consumer	NDR
	Technology	REST API
	API Documentation	N/A
	Partners involved	SLP

4.18 Network Situation Evaluation (NSE)

4.18.1 Function

Network Situation Evaluation provides a risk assessment of the overall network based on input from other functions and can also project the attack intensity for the network.

It provides the following capabilities:

1. calculates the overall network risk based on inputs from different sources

including CSA, CASM and NDR

2. *attack intensity prediction* fuses information about the ongoing attacks from diverse sources and estimates an overall attack intensity. The overall intensity is derived from the number and severity of attacks against the whole network. The prediction can then give a warning about incoming increase or decrease of attacks
3. provides a visualisation of both current and future network risk status

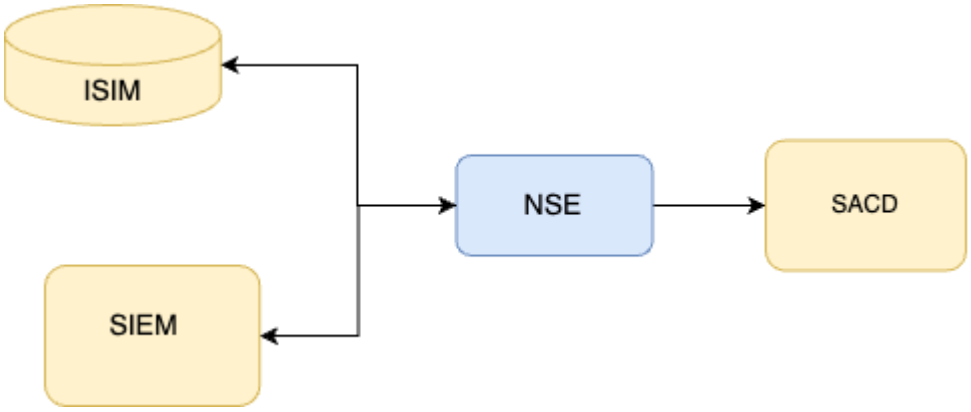

Figure 30 - NSE Functional Architecture

4.18.2 Provided services

NSE does not provide any services of itself but consumes many others as described below.

A. Interfaces

nISIM_AssetSearch	**Detailed Description**	This interface allows the NDR to retrieve asset details from ISIM
	From provider	NSE
	To Consumer	Infrastructure and Services Information Model (ISIM)
	Technology	REST API,
	API Documentation	NA
	Partners involved	MONT, MUNI

nSIEM-Alerting	**Detailed Description**	This interface output alerts that indicate suspect or anomalous behaviour.
	From provider	SIEM
	To Consumer	NSE
	Technology	Rest API

	API Documentation	https://documentation.wazuh.com/current/getting-started/use-cases/log-analysis.html#rules-and-decoders
	Partners involved	TUS, MONT,

4.19 Mitigation Manager (MM)

4.19.1 Function

The Mitigation Manager functional component is responsible for deciding which mitigation actions, if any, should be taken in response to a detected incident.

The Mitigation Manager receives security alerts forwarded from the SIEM as mitigation requests. It then interfaces with the ISIM component to analyse mission, risk and network status projection as factors in the mitigation decision process, as well as to gather the information model captures and represents all the entities of interest in the environment (such as devices, networks, applications, data, or users). The mitigation manager also interacts with the Playbooks Tool and Workflow Orchestrator components in order to enforce any mitigation playbook(s) that have been selected in response to security alerts.

The component will make use of the Optaplanner rule-based inference engine to infer the best mitigations for each particular situation, leveraging situational information on available playbooks together with the Cyber Situational Awareness data and Risk Scores provided by ISIM. Additionally, the component will employ a localized graph tracking algorithm based on the MITRE Attack Flow format in order to record ongoing attacks and predict potential next steps, information which will be fed to the logic solver to enrich the available context for calculating effective mitigations.

The kind of mitigation actions to be enforced as part of the playbooks might include:

- Network filtering (diverting/dropping/blocking/mirroring network traffic)
- Channel protection
- Ansible script execution
- Host quarantining
- Outdated software updating
- Service restoration
- SIEM indicator updates
- SIEM rule updates (such as YARA or Sigma rules)
- CTI Sharing

4.19.2 Provided services

4.19.2.1 Mitigation Service

A. Description

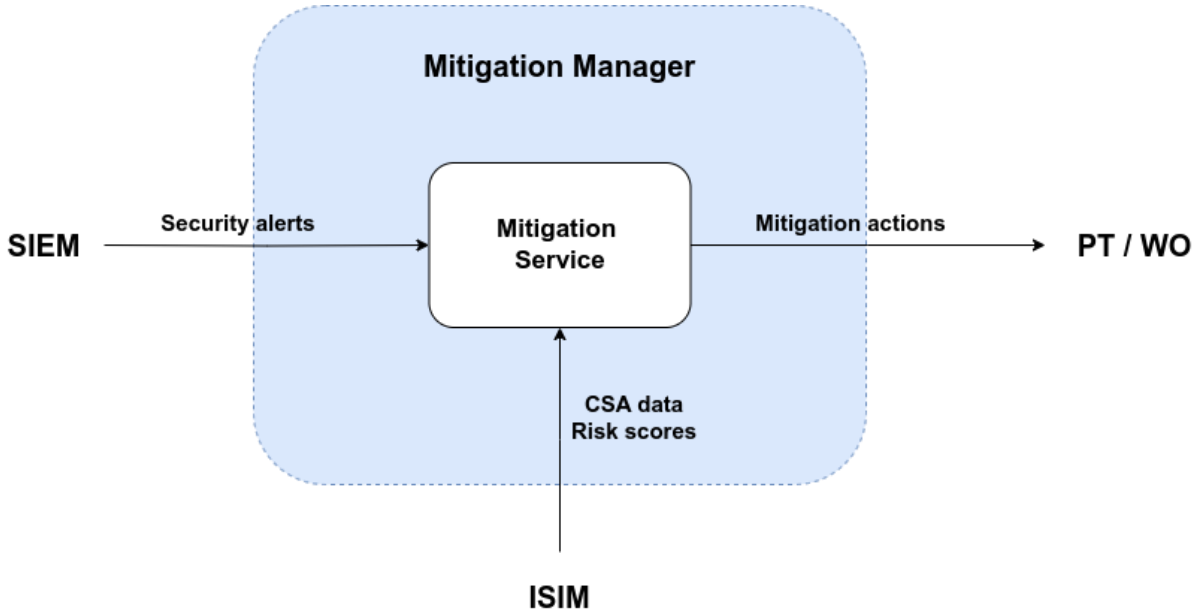

Figure 31 - Mitigation Manager

This service implements the Mitigation Manager function.

B. Capabilities

Enforce mitigation actions (playbooks)

C. Type

Internal / External.

D. Consumers

- PT, WO

E. Pre-conditions to consume the service

Mitigation playbooks are available

F. Interfaces

nPT_out	**Detailed Description**	This interface allows interacting with the Playbooks Tool component, in order to trigger the enforcement of one or more selected playbooks.
	From provider	Mitigation Service

	To Consumer	Playbooks Tool
	Technology	REST API
	API Documentation	N/A
	Partners involved	UMU

nISIM_in	**Detailed Description**	This interface allows obtaining Cyber Situational Awareness and Risk Score information from the ISIM component.
	From provider	ISIM
	To Consumer	Mitigation Manager
	Technology	REST API
	API Documentation	N/A
	Partners involved	UMU, MUNI

nSIEM_in	**Detailed Description**	This interface allows the Mitigation Engine to receive alerts that will trigger the reactive mitigation process.
	From provider	SIEM
	To Consumer	MM
	Technology	PUB/SUB
	API Documentation	N/A
	Partners involved	UMU

nWO_out	**Detailed Description**	This interface allows the MM to enforce mitigations through Workflow Orchestrator component
	From provider	MM
	To Consumer	WO
	Technology	REST API
	API Documentation	N/A
	Partners involved	UMU-MUNI

nAIBAST_out	**Detailed Description**	This interface allows the AI based automated security testing to be triggered as part of a mitigation enforcement by MM
	From provider	MM
	To Consumer	AI based automated security testing
	Technology	REST API
	API Documentation	N/A
	Partners involved	UMU, JR

4.20 Playbooks Tool (PT)

4.20.1 Function

The Playbooks Tool component is a CoA playbook engine intended to execute specific workflows that contains as a set of actions to be enforced as countermeasures to mitigate the attacks. Those workflows can contain several steps to enforce OpenC2 actions and any other step needed in the workflow, such as deploying additional rules in SIEM.

Additionally, the playbook can include in its steps the invocation of Security Orchestrator that, in turn, can enforce certain security/privacy policies, beyond common-basic actions as defined in OpenC2. It might include for instance the deployment and configuration of certain virtual security functions in the network.

The PT should be ideally able also to define graphically standardised workflows (using standard CACAO).

The kind of mitigation actions to be enforced as part of the playbooks might include:

- Network Filtering: divert, drop-block, mirror network traffic.
- Set-up Channel protection
- Run Ansible Scripts
- Quarantining host(S)
- Update outdated software
- Restore services.
- Update Indicators in SIEM
- Add new rules in SIEM, e.g. yara rule, sigma rules for detection.
- Share CTI

4.20.2 Provided services

4.20.2.1 Playbook Service

A. Description

Manages the playbook database and handles incoming playbook requests.

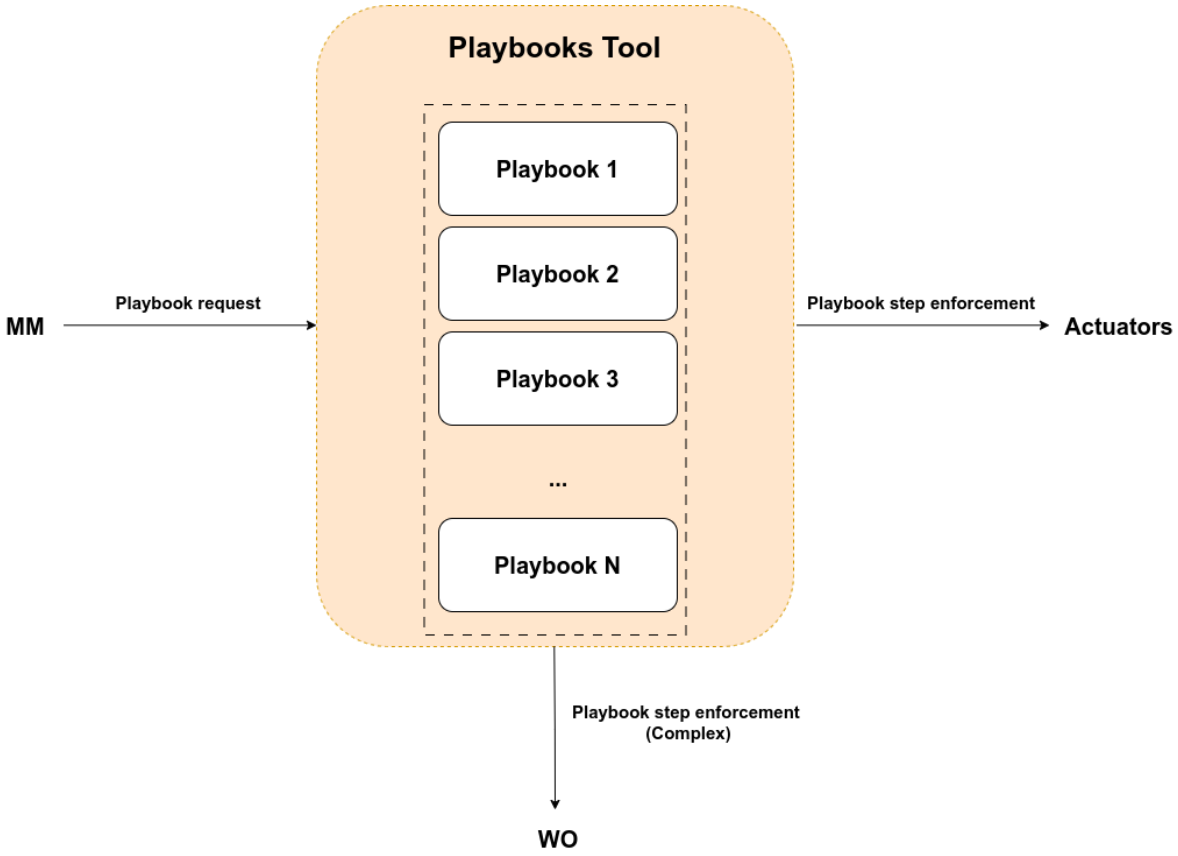

Figure 32 - Playbook Tool

The PT service is the tool that implements the PT functional component functionality.

B. Capabilities

Enforce and define CoA playbooks

C. Type

Internal / External.

D. Consumers

- Mitigator, Orchestrator,

E. Pre-conditions to consume the service

Abstract mitigation playbooks are defined (e.g. in CACAO)

F. Interfaces

nMM_runPlaybook	**Detailed Description**	This interface allows interacting with the Playbook engine in order to trigger the enforcement of 1 or more selected playbooks.
	From provider	Mitigation Service
	To Consumer	Playbook engine
	Technology	Rest API or Fabric
	API Documentation	N/A
	Partners involved	UMU

nRO_orchestrate	**Detailed Description**	This interface allows the PT to enforce complex actions through the Resource Orchestrator
	From provider	PT
	To Consumer	RO
	Technology	Rest API, Temporal interface
	API Documentation	N/A
	Partners involved	UMU, MUNI

nActuator_enforce	**Detailed Description**	This interface allows the PT to enforce mitigations actions such as OpenC2
	From provider	PT
	To Consumer	Actuator
	Technology	Rest API, OpenC2 and others
	API Documentation	https://docs.oasis-open.org/openc2/oc2ls/v2.0/oc2ls-v2.0.html
	Partners involved	UMU, MUNI

4.21 Workflow Orchestrator (WO)

4.21.1 Function

- Orchestration and automation to support actions taken as part of Courses of Action (CoA) playbooks
- Playbooks are advertised and requested through the *Playbook Engine* interface
- Playbooks can be triggered either on-demand or scheduled periodically
- The *Orchestrator* component can provide the data back to the *Mitigation Engine* creating a constant feedback loop
- Workflow orchestration consists of two parts:
    - high-level, low-code, visual workflow orchestration in Shuffler.io
    - low-level, all-code, complex task-as-a-workflow orchestration in Temporal.io
- The workflow orchestration and automation should be easily extensible, simple to use, durable, not hard to maintain, not hard to define, and perform complex tasks. There is no existing orchestration engine that fulfils all of the mentioned requirements; therefore, the orchestration consists of the *Playbook Engine* component and the *Orchestrator* component.
- The *Playbook Engine* component represents the standardised interface for playbook management. The playbooks will be easy to view, edit, and use from a high-level point of view.
- The *Orchestrator* component will perform complex and/or time-demanding parts of the workflow. It will ensure that the execution of the actions defined via a workflow is timely, durable, and complete. Typical complex tasks are e.g.:
    - building an environment for analysing network traffic and performing the analysis,
    - building an environment for disk analysis, and performing the analysis.
- It provides information about the result of actions and data obtained during their execution.
- It uses a defined subset of tools from Target Actuators to perform the actions.

4.21.2 Provided services

4.21.2.1 Orchestration and Automation platform (Shuffler.io + Temporal.io)

A. Description

The Playbook Engine defines use cases for which it needs cooperation from the orchestrator, typically more complex actions that are difficult to implement in Shuffler. In the orchestrator, we implement the actions by breaking them down into individual steps. Based on the request from the Playbook Engine, the Orchestrator will execute the actions on the Target Actuators and monitor them. It will provide information about the progress and outcome of the actions back to the Mitigation Engine.

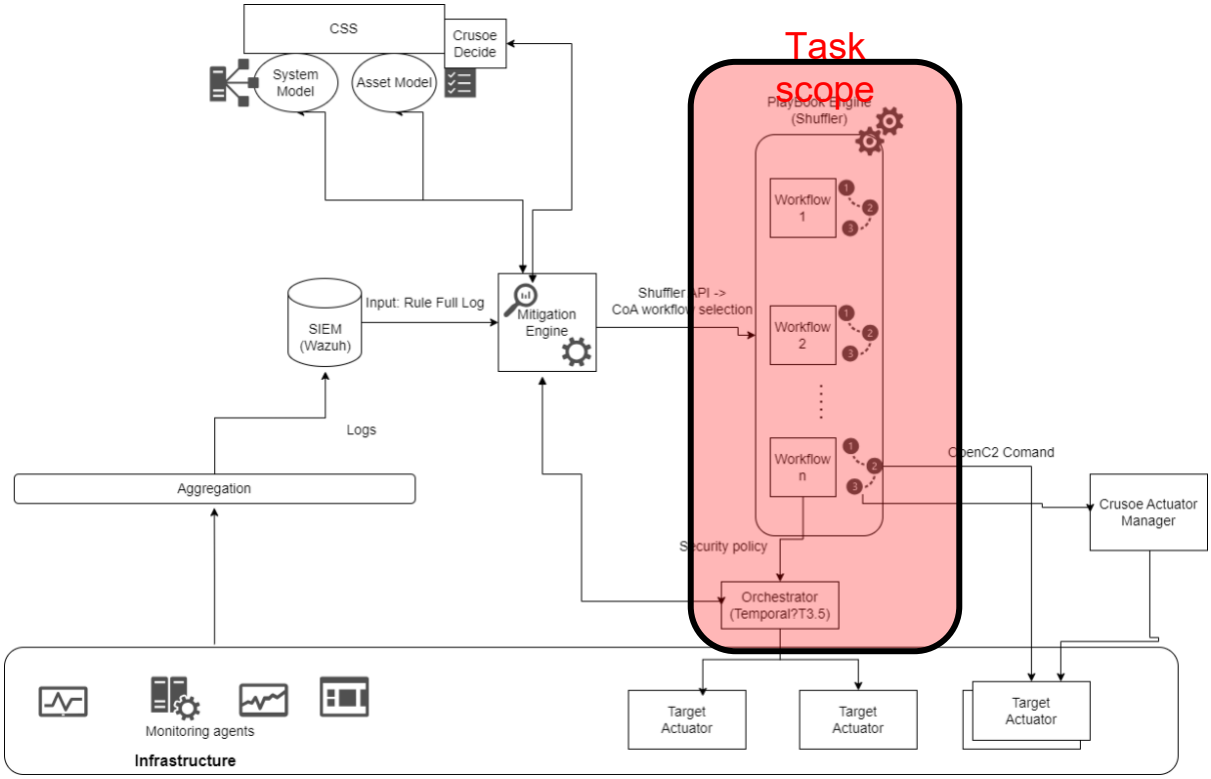

Figure 33 -Orchestration Service

B. Capabilities

Manage playbooks and execute actions

C. Type

Internal / External.

D. Consumers

Mitigation Engine, external clients.

E. Pre-conditions to consume the service

Playbook Engine needs to define a playbook providing the required workflow.

F. Interfaces

Playbook Engine Interface (Shuffler.io)	**Detailed Description**	This interface allows the Mitigation Engine and external clients to send a request for a playbook execution. The playbook will represent a predefined use

		case and will be accompanied with any necessary parameters and features.
	Interface provider	Playbook Engine
	Interface consumer	Mitigation Engine, external clients
	Technology	REST API, UI
	API Documentation	NA
	Partners involved	

Orchestrator Interface (Temporal)	**Detailed Description**	This interface allows the Playbook Engine to send a request for orchestration of a complex action. The action will belong to a predefined set of supported use cases and will be accompanied by any necessary parameters and features.
	Interface provider	Orchestrator
	Interface consumer	Playbook Engine
	Technology	gRPC API, REST API
	API Documentation	https://docs.temporal.io/
	Partners involved	

Mitigation Engine Interface	**Detailed Description**	This interface allows the Orchestrator to send the feedback on the orchestrated actions back to the Mitigation Engine.
	Interface provider	Mitigation Engine
	Interface Consumer	Orchestrator
	Technology	REST API
	API Documentation	NA
	Partners involved	

Target Actuator Interface	**Detailed Description**	This interface allows the Orchestrator to manage devices required to carry out the orchestrated actions, including operations such as deploying a virtual environment,

		setting up a monitoring infrastructure, and carrying out data analysis.
	Interface provider	Target Actuators
	Interface consumer	Orchestrator
	Technology	REST API
	API Documentation	NA
	Partners involved	

4.22 Artificial Intelligence based automated security testing (AIBAST)

4.22.1 Function

AI technologies, such as **Reinforcement Learning (RL)** and **Large Language Models (LLMs)**, inject new levels of automation and insight into security-testing activities.

RL was adopted in Phase 1 of the component development. In an RL set-up, an *agent* interacts with a target environment and receives rewards or penalties based on its actions. Over many iterations it learns the optimal behaviour that maximises cumulative reward—effectively uncovering weak points across the entire attack surface. Although RL delivered deep technical insight, we observed integration issues and decided to pivot to a more flexible technology stack.

In Phase 2 we switched to LLM-driven workflows to capitalise on their broad usability and rapid integration capabilities. An LLM can ingest natural-language prompts and instantly produce executable test cases, exploit scripts, or remediation advice.

AIBAST tool has the following main features:

- Performs an automated security testing using AI
- Improves resilience preparedness of cyber security teams
- Supports capacity building for security test teams via a guided penetration testing approach.

4.22.2 Provided services

4.22.2.1 Artificial Intelligence based automated security testing Service

A. Description

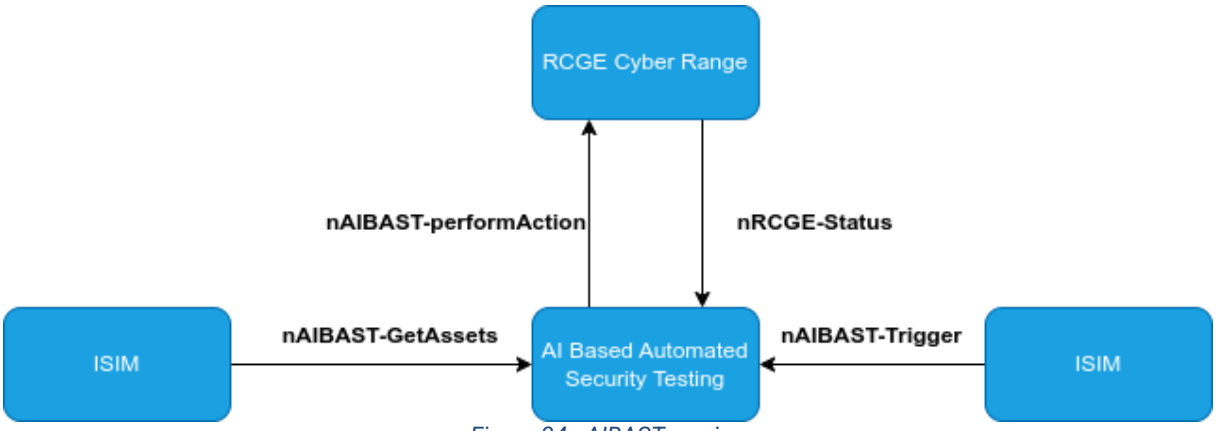

Figure 34 - AIBAST service

B. Capabilities

C. Type

D. Consumers

Cyber range testing environment

E. Pre-conditions to consume the service

Operation modes, actions and reward functions for chosen environment are defined.

F. Interfaces

nAIBAST-GetAssets	**Detailed Description**	This interface allows the AI based automated security testing to obtain information about the assets used in the environment
	From provider	ISIM
	To Consumer	AI based automated security testing
	Technology	REST API
	API Documentation	NA
	Partners involved	

nAIBAST-PerformAction	**Detailed Description**	This interface allows the AI based automated security testing to interact with the components of the environment to test.
	From provider	AI based automated security testing

	To Consumer	RCGE cyber range
	Technology	NA
	API Documentation	NA
	Partners involved	YAMK

nRCGE-Status	**Detailed Description**	This interface allows the AI based automated security testing to obtain some status information from the RCGE.
	From provider	RCGE cyber range
	To Consumer	AI based automated security testing
	Technology	NA
	API Documentation	NA
	Partners involved	YAMK

nAIBAST-Trigger	**Detailed Description**	This interface allows the AI based automated security testing to be triggered as part of a mitigation enforcement by MM
	From provider	MM
	To Consumer	AI based automated security testing
	Technology	REST API
	API Documentation	N/A
	Partners involved	UMU

6 Architecture Extensions and Open Challenges

This section of the document outlines extension points in Resilmesh that can be leveraged by Open Call parties to add to the Resilmesh functional scope. These extensions leverage the Integration Reference Points(IRPs) described in Section 3.

We consider two main extension categories as described below.

6.1 Extension to new domains and systems

This task will address extensions to Resilmesh via the **collaboration mesh IRP's** . The collaboration mesh consists of two sub-parts:

1. A **connectivity mesh** to describe the *interconnection* of the systems' components. This term includes not only the actual network connectivity but also the end-to-end aggregation data processing pipeline. This applies primarily to the two layers shown in the Architecture diagram earlier i.e. the Collaboration Mesh and the Aggregation Plane. For this subject area we are interested in proposals:

- that demonstrate how the connectivity mesh capability can provide secure adaptivity to improve the resilience of the Resilmesh platform by improve resilience engineering techniques such as redundancy and dynamic positioning as described in the NIST cyber resilience engineering guidelines – see table 3 below.
- We expect a Kubernetes-based service mesh implementation. The task shall also indicate and demonstrate how their implementation will address known security issues that could arise in the use of Kubernetes service mesh such as mTLS configuration issues, sidecar injection vulnerabilities, lack of ingress/egress controls, certificate and key management risks etc.
- The proposal should be grounded in a specific IT or OT application scenario.

Table 6 -NIST cyberresilience engineering guidelines.

Technique	Description
Redundancy	Provide multiple protected instances of critical resources
Dynamic Positioning	Flexible function component allocation and composition

2. An **interworking mesh** based on the use of open protocols, standards and best practices to enable ease of *integration* and *cooperation* between security applications/controls including third party tools. In this task we are looking for proposals that will extend the interoperability in the security application layers i.e. Threat Awareness, Situation Assessment and Security Operations- between

Resilmesh components and other tools along the lines of the Open XDR Architecture (OXA) principles including the use of 'Meshroom' tool.

Proposals should address <u>one</u> of *connectivity mesh* or *interworking mesh*

6.2 New Analytic Algorithms and Architectures

Zone based Anomaly detection architectures: This area is concerned with exploring the use of novel approaches to edge anomaly detection as well as alerts aggregated by zone. A security zone is a logical grouping of physical, data, and application assets sharing common security requirements. Zones have long been a central feature of Industrial Control System (IDS) networks (https://bit.ly/3yL1g0y) . They are now also becoming mainstream in IT security solutions as a part of the Zero Trust approach i.e. secure device enclaves such as the Google Cloud Platform 'service perimeter' or network segment etc.
This task will therefore explore, for a particular use-case or scenario, how zone based anomaly detection may be implemented in Resilmesh. This relies on the capability to associate a number of assets as a single unit for analysis purposes and may involve extension of the ISIM asset management tool or the NSE network risk management tool as well as possible extension to Wazuh dashboards

Novel edge AI AD architectures and algorithms: The deployment of edge-based AI opens many possibilities for experimenting with different algorithms and architectures, taking into consideration the needs of the domain and the data. Some possible approaches might be

- Use Ensemble methods
- Distributed deep learning
- Incremental learning
- Edge Agent/ic architectures
- Edge-to-Edge Collaborative Anomaly Detection
- Multi-modal and multi-rate data sources fusion

User and Entity Behaviour Analytics: UEBA shifts the focus of detection from Indicator of Compromise (IoC) approaches to focus on higher level Indicators of Behaviour (IoB). UEBA can apply to both endpoint and network traffic behaviours. One approach here could be to extend the Resilmesh NDR functional component with network behaviour analytics such as those identified in the Network Traffic Analysis category in the Mitre D3FEND taxonomy (https://d3fend.mitre.org/). UEBA analytics for IIoT/OT infrastructure in particular are of interest.

These are suggestions and there can be other approaches.

7 Conclusion

This document has outlined the final version of the Resilmesh Functional Architecture, designed to assist infrastructure providers in enhancing resilience across diverse and distributed operational environments.

The architecture is organized into several planes to address the functional requirements defined in D2.1. The architecture is inspired by the SOAPA model and incorporates best practices from NIST Special Publication 800-160 related to cyber resiliency techniques, such as Contextual Awareness, Analytic Monitoring, Coordinated Protection, Dynamic Positioning, and Adaptive Response.

The architecture is divided into multiple functional planes, including Infrastructure, Aggregation, Collaboration, Threat Awareness, Situation Assessment, and Security Operations. Each plane includes specific functional components that have been thoroughly described, along with their key services and expected interfaces. The document also details the primary workflows supported by the architecture.

Furthermore, this document identifies potential extension points within the architecture and outlines open challenges that could guide future development of the Resilmesh platform, particularly through open call specifications in WP8.

8 References

[SOAPA] security operations and analytics platform architecture https://www.csoonline.com/article/566687/security-operations-activities-to-watch-in-2019.html

[Endsley1995] ENDSLEY, Mica R. Toward a theory of situation awareness in dynamic systems. Human factors, 1995, 37.1: 32-64.

[Husak2020] HUSÁK, Martin; JIRSÍK, Tomáš; YANG, Shanchieh Jay. SoK: contemporary issues and challenges to enable cyber situational awareness for network security. In: Proceedings of the 15th International Conference on Availability, Reliability and Security. 2020. p. 1-10.

[Gutzwiller2020] GUTZWILLER, Robert; DYKSTRA, Josiah; PAYNE, Bryan. Gaps and opportunities in situational awareness for cybersecurity. Digital Threats: Research and Practice, 2020, 1.3: 1-6.

[Husak2018] HUSÁK, Martin, et al. Survey of attack projection, prediction, and forecasting in cyber security. IEEE Communications Surveys & Tutorials, 2018, 21.1: 640-660.

[Husak2022] HUSÁK, Martin, et al. CRUSOE: A toolset for cyber situational awareness and decision support in incident handling. Computers & Security, 2022, 115: 102609.

[Ahmad2021] AHMAD, Atif, et al. How can organizations develop situation awareness for incident response: A case study of management practice. Computers & Security, 2021, 101: 102122.

[Khraisat2019] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (12 2019). Survey of intrusion detection systems: techniques, The ResilMesh software architecture is constructed according to the Security Operations and Analytics Platform Architecture (SOAPA and challenges. Cybersecurity, 2. doi:10.1186/s42400-019-0038-7

[McMahan2017] McMahan, B., Moore, E., Ramage, D., Hampson, S., & Arcas, B. A. y. (20--22 Apr 2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. In A. Singh & J. Zhu (Eds.), Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (pp. 1273–1282). Retrieved from https://proceedings.mlr.press/v54/mcmahan17a.html

[Singh2022] Singh, P., Singh, M. K., Singh, R., & Singh, N. (2022). Federated Learning: Challenges, Methods, and Future Directions. In S. P. Yadav, B. S. Bhati, D. P. Mahato, & S. Kumar (Eds.), Federated Learning for IoT Applications (pp. 199–214). doi:10.1007/978-3-030-85559-8_13

[Ruzafa2023] Ruzafa-Alcázar, P., Fernández-Saura, P., Mármol-Campos, E., González-Vidal, A., Hernández-Ramos, J. L., Bernal-Bernabe, J., & Skarmeta, A. F. (2023). Intrusion Detection Based on Privacy-Preserving Federated Learning for the Industrial

IoT. IEEE Transactions on Industrial Informatics, 19(2), 1145–1154. doi:10.1109/TII.2021.3126728

[Kairouz2021] Kairouz, P. et al. (2021). Advances and Open Problems in Federated Learning. Retrieved from http://ieeexplore.ieee.org/document/9464278

[CVE] https://www.cve.org/

[NVD] https://nvd.nist.gov/

[CDM] https://cyberdefensematrix.com/

[NetBox] https://docs.netbox.dev/en/stable/

[Komarkova2018] KOMÁRKOVÁ, Jana, et al. CRUSOE: Data model for cyber situational awareness. In: Proceedings of the 13th International Conference on Availability, Reliability and Security. 2018. p. 1-10.

[Metrics] https://nvd.nist.gov/vuln-metrics/cvss

[Javornik2022] Javorník, Michal, and Martin Husák. "Mission-centric decision support in cybersecurity via Bayesian Privilege Attack Graph." Engineering Reports 4.12 (2022): e12538. https://onlinelibrary.wiley.com/doi/pdf/10.1002/eng2.12538
[Istio] https://istio.io/latest/about/service-mesh/
[Oltsik] Jon Oltsik, 2016 Goodbye SIEM, Hello SOAPA https://bit.ly/3DMQKcB
[Mesh] What is a services Mesh, NGINX https://www.nginx.com/blog/what-is-a-service-mesh/
[Vector] https://vector.dev/docs/about/under-the-hood/architecture/pipeline-model/