



## ResilMesh Platform Reference Implementation

<b>Deliverable Number</b>	<b>D3.1</b>
This document defines the ResilMesh framework architecture within its pipeline flow, the technologies involved and its extensibility.	
<b>Deliverable Leading:</b>	<b>SLP</b>
<b>Due Date:</b>	<b>29/02/2024</b>
<b>Submitted Date:</b>	<b>28/02/24</b>
<b>Author(s)</b>	<b>J. Oliviera, C. Diver, M. Otic, B.Lee</b>
<b>Reviewer(s):</b>	<b>GMV, JR</b>

## Revision History

Version	By	Date	Changes
A	B. Lee	28/02/24	Updated according to reviews: - references as end notes. - fig 5 added - justification for component selection - new template
A2	J. Oliviera, B.Lee, M. Otic, C Diver	7/02/24	Extended with goals and requirements
A1	J. Oliviera	5/1/2024	First Draft

## 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

<b><u>RESILMESH PLATFORM REFERENCE IMPLEMENTATION</u></b> .....	<b>1</b>
TABLE OF CONTENTS .....	3
<b><u>TERMINOLOGY</u></b> .....	<b>4</b>
<b><u>ABBREVIATION LIST</u></b> .....	<b>5</b>
<b><u>EXECUTIVE SUMMARY</u></b> .....	<b>6</b>
<b><u>INTRODUCTION</u></b> .....	<b>6</b>
PURPOSE .....	6
<b><u>GOALS AND PRINCIPLES</u></b> .....	<b>7</b>
CYBERSECURITY MESH .....	8
OPEN XDR ARCHITECTURE .....	10
DEVELOPING CYBER RESILIENT SYSTEMS .....	11
<b><u>RESILMESH PLATFORM ARCHITECTURE</u></b> .....	<b>12</b>
CONCEPT .....	12
IMPLEMENTATION PHASING .....	15
AGGREGATION AND COLLABORATION MESH LAYERS.....	15
ANALYTICS LAYER.....	19
OPERATIONS LAYER.....	20
<b><u>ANNEX 1 – INITIAL IMPLEMENTATION</u></b> .....	<b>23</b>
THE TECHNOLOGIES .....	23
THE PIPELINE.....	23
MODULES .....	23
FRAMEWORK DEMO.....	26
<b><u>REFERENCES</u></b> .....	<b>28</b>

# Terminology

**Platform** – The various software that compose the system base.

**Message Broker** – The software taking care of queueing the events, the modules fetch the message broker and push back to it.

**Node** – A specific point of the pipeline where data pass through, like hosts, routers.

**Pipeline** – The flow of data through the framework from the start node to the end.

**Frontend** – The end user interface (UI).

**Module** – A piece of software of the framework holding a specific logic, in the Resilmesh architecture, this is sitting between the Message Broker and the next module or the retention store. The modules are the extendibility of the framework.

## Abbreviation List

**APT** – Advanced Persistent Threat

**ASN** – Autonomous System Number, a number allocated for a Network Operator.

**AS** - Autonomous System, same concept as previous.

**CACAO** - Collaborative Automated Course of Action Operations

**CLI** – Command Line Interface

**CI** – Critical Infrastructure

**CTI** – Cyber Threat Intelligence

**ECS** – Elastic Common Schema

**FL** - Federated Learning

**GUI** – Graphical User Interface

**IS** – Information System

**NSA** – Network Situation Awareness

**OCA** – Open Cyber Security Alliance

**OCSF** - Open Cybersecurity Schema Framework

**OXA** – Open XDR Architecture

**OT** – Operational Technology

**NS** – Nameserver

**NSA** - Network Situation Awareness

**MISP** – Malware information Sharing Platform

**POC** – Proof of Concept

**SIEM** – Security Incident and Event Management

**SOA** – Start of Authority

**STIX** – Structured Threat Information eXchange

**SOAPA** – Security Operations and Analytics Platform Architecture

**XDR** – eXtended Detection and Response

# Executive Summary

Resilmesh will develop a *reference implementation* of a Security Operations and Analytics Platform Architecture (SOAPA) to support organisations achieve higher levels of security and resilience in their critical infrastructures. A reference implementation is used to identify alternate candidate technologies that could be used to implement the platform and to select one set of technologies on which to base the initial implementation of the platform.

To achieve this Resilmesh embraces two key goals:

1. To improve integration and interoperability between security devices.
2. To support cyber resiliency engineering techniques both across all phases of the cyber resilience lifecycle (Prepare, Absorb, Recover, Adapt)

Resilmesh adopts several industry best practises to fulfil these goals namely i) the Gartner Cybersecurity Mesh<sup>i</sup>, the Open Cybersecurity Alliance Open XDR Architecture<sup>ii</sup> and the NIST publication *Developing Cyber-Resilient Systems*<sup>iii</sup>. These best practises provide a solid foundation for the development of the Resilmesh SOAPA which is based on and derived from the industry general SOAPA – as shown in Figures 3 & 4 below. The scope of this document does not include application functions but rather the services of the platform that the applications use. The SOAPA has four functional layers:

- Operations Layer
- Analytics Layer
- Software Connectivity Services (Collaboration mesh) Layer
- Aggregation Layer

Detailed non-functional resilience requirements are derived for each of the four layers of the SOAPA. Candidate implementation technologies are enumerated both to implement the services of each layer and also to adhere to the key objectives outlined above.

A set of technologies is identified for the principal services on each layer and a set of specific technologies is selected for construction of the reference architecture. A limited proof of concept has been successfully implemented.

## Introduction

### Purpose

The purpose of this document is to specify a reference implementation of the Resilmesh Security Operations and Analytics Platform Architecture (SOAPA). A reference implementation is used to identify alternate candidate technologies that could be used to implement the platform and to select one set of technologies on which

to base the initial implementation of the platform . The Resilmesh SOAPA platform will be based on industry accepted best practises and standards as outlined in the next chapter.

**Note that the scope of this document does not include application functions but rather the services of the platform that the applications use.**

The document outlines influencing architectural factors for the platform ( Chapter 0 ), details the platform architecture ( Chapter 0 ) and an initial reference implementation (Chapter 4), including a detailed set of requirements arising from those.

## Goals and Principles

Resilmesh will develop a SOAPA to support organisations achieve higher levels of security and resilience in their critical infrastructures. The dramatic growth of these systems has created major challenges for security teams:

- CI attack surfaces have increased. CI's are *complex*– they contain multiple infrastructure layers with many types of components (edge, cloud, IoT etc). This creates dependencies between organisation business processes (missions) and the hardware and software assets that support them, which in turn facilitates multiple attack entry points (vectors). CI's are *heterogeneous* i.e., composed of many different technologies (e.g., due to the blurring of Information and Operational Technology (IT/OT) boundaries. This further increases the range of potential attack vectors. CI's are *dispersed* over wide geographical areas (cloud/edge/endpoint) making traditional perimeter-based security approaches increasingly ineffective and creating yet more attack vectors.
- Attacks have become more complex and sophisticated. Advanced persistent threats (APT's) with a focus on specific targets over an extended time period are particularly sinister. They typically seek to exfiltrate information or impede critical aspects of a mission or organization. They are increasingly based on multi attack vector approaches including, for example, cyber, physical, and deception vectors. They are often carried out by nation state adversarial actors.

**To address these challenges Resilmesh embrace two key goals:**

1. To improve integration and interoperability between security devices.
2. To support cyber resiliency engineering techniques both across all phases of the cyber resilience lifecycle (Prepare, Absorb, Recover, Adapt)

The Resilmesh SOAPA will adopt several best practices from industry sources to support these goals :

- The Gartner Cybersecurity Mesh: which is essentially about interoperability and collaboration between siloed security tool and controls. In their own words: *“The rapid evolution and sophistication of cyberattacks and the migration of assets to the hybrid multicloud creates a perfect storm. IT leaders must integrate*

*security tools into a cooperative ecosystem using a composable and scalable cybersecurity mesh architecture approach”*

- Open Cybersecurity Alliance (OCA) Open XDR Architecture : The goal of this project is to facilitate interactions between security products, using open standards and APIs and with a special focus on Detection and Response. This project aims to define the architecture of an ideal eXtended Detection and Response approach.
- NIST publication SP 800 160 R2 Developing Cyber-Resilient System: This document describes a set of resiliency engineering goals, principles and techniques to *“architect, design, develop, maintain, and sustain the trustworthy cyber systems trustworthiness of systems with the capability to anticipate, withstand, recover from, and adapt to adverse conditions, stresses, attacks, or compromises that use or are enabled by cyber resources. From a risk management perspective, cyber resiliency is intended to reduce the mission, business, organizational, or sector risk of depending on cyber resources.”*

This document defines a reference implementation based on the above best practices to realise the Resilmesh SOAPA architecture.

## Cybersecurity Mesh

The Cybersecurity mesh identified the following challenges:

- **The perimeter has become more fragmented.** Many applications and data are no longer in the company-owned data centre, and users are accessing cloud-based applications from anywhere. In a traditional data centre, network perimeter security was a common mechanism for controlling access. Within a distributed environment that supports assets everywhere and access from anywhere, identity and context have become the ultimate control surface.
- **Many organisations are adopting a multicloud approach and need a consolidated security approach:** Organizations tend to consume services from more than one cloud provider. With every cloud provider supporting a different set of policies, creating a consistent security posture across cloud providers is challenging. The huge on-premises estate of services found in most organizations only compounds the challenges.
- **Attackers don't think in silos but organisations often deploy siloed security controls:** A proper defensive posture requires eliminating silos and inefficiencies, both from an organizational perspective as well as within technology. Many security tools work within their own view of the world with minimal interoperability – or even awareness – of other tools

The cybersecurity mesh approach aims to address these challenges by integrating security controls into, and extending them to span, even widely distributed assets. It does this by defining *‘several foundational layers - i.e. enabling services - to act as a force multiplier when integrating different security products.’*



### Cybersecurity Mesh Architecture

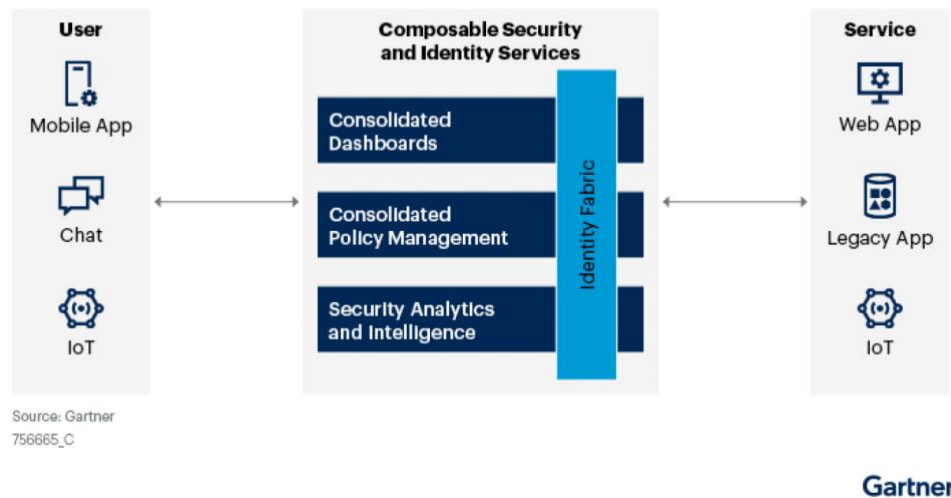


Figure 1 Gartner Cybersecurity Mesh

These are:

- **Security analytics and intelligence:** Combines data and lessons from other security tools, and provides analyses of threats and triggers appropriate responses.
- **Distributed identity fabric:** Provides capabilities such as directory services, adaptive access, decentralized identity management, identity proofing and entitlement management.
- **Consolidated policy and posture management:** Can translate a central policy into the native configuration constructs of individual security tools or, as a more advanced alternative, provide dynamic runtime authorization services.
- **Consolidated dashboards:** Offers a composite view into the security ecosystem, enabling security teams to respond more quickly and more effectively to security events.

Some **key principles or guidelines** for the cybersecurity mesh approach are:

- Centralized management and distributed enforcement
- Collaborative approach between integrated security tools and detective and predictive analytics
- **Composability, scalability and interoperability for security controls:** Note that Gartner's concept of 'composable' seems to be *'technologies that ease integration via plug-in APIs that allow extensions and customization.'* Moreover *"Integration can be achieved through a mix of open standards and interfaces, proprietary APIs, and point integrations (e.g., ad hoc integrations between vendors' tools).*
- **Cross-Domain Security Analytics:** For example, a security product may have its own native analytics engine that calculates a risk score for a particular session, user or transaction. A broader security analytics and intelligence tool

could then use this risk score, applying it in a different context from that originally intended.

## Open XDR Architecture

The OXA concept is represented in Figure 2. It notably provides a library of mapping files so that every cybersecurity product can speak to another one. Benefits for users include:

- Industry best practices are available as playbooks
- Ability to plug solutions easily without waiting vendor integration
- CTI-enabled products



Figure 2 Open XDR Architecture

OXA relies on:

- STIX/TAXII for threat intelligence dissemination along security components
- OpenC2 for ordering generic actions that should be executed on security components
- CACAO for defining the community orchestrated best practices on response strategies
- ECS/OCSF for ingesting logs from security components
- A open REST/API to create an interface between security products and the XDR back-end side

## Developing Cyber Resilient Systems

SP800 160 v2 R1 sets out to “address security, safety, and resiliency issues from the perspective of stakeholder requirements and protection needs using established engineering processes”. Most significantly from our point of view it identifies fourteen *cyber resiliency techniques* which are described in Appendix D3 of the document. We have identified five (5) of these techniques that we consider pertinent to ResilMesh and the table below outlines how these apply- the specific techniques from the standard are indicated in italics.

*Table 1 Cyber Resiliency Techniques applicable to ResilMesh*

ResilMesh Feature	Implementation Approach	Description
<b>Context Awareness</b> Ensure the organisation has sufficient overview of its environment to support situational awareness  Contextual Awareness	Resource awareness	Maintain information about systems assets, resources, services and their connectivity.
	Mission Awareness	Maintain information about enterprise business functions and their dependencies on IS resources and services as well as the threat status of resources.
	Threat Awareness	Ensure awareness of threat actors and adversaries. Maintain ongoing observation and analysis of threat event and indicators and enable better prediction of cyber security threats.
	Risk based response	Ensure that threat response/mitigation actions are chosen to prioritise those IS capabilities required to ensure enterprise critical mission continuity.
<b>Adaptive</b> Ensure ResilMesh can be used for a variety of application domains, computing topographies, and mission types.	Heterogeneous Infrastructures	Ensure ResilMesh can support organisations of different types, domains and size.
	Flexible function allocation and composition	Ensure that security functions can be flexibly located where required across the dispersed IS infrastructure. Enable analytic functions to be pipelined/composed to support flexible aggregation and processing of events.

Coordinated Protection Dynamic Positioning Adaptive Response	Interoperability	Ensure that diverse security function and tools can share information and actions
	Orchestration	Ensure that critical cyber operations are automated and course of action playbooks are prepared to ensure collaboration between diverse tools to ensure effective detection, analysis and response.
	Zoning	Ensure that ResilMesh security functions support the using of security zoning approaches such as those of Zero Trust and Industrial Control System (ICS) zones <sup>iv, v</sup> as defined in IEC 62443.
Security Analytics Analytic Monitoring	Sensor Fusion	Combine monitoring data from different sources at different points in the system as well as with externally provided CTI
	Threat Monitoring	Monitor and analyse system components to look for indication of adversary activity
	Forensics and Behavioural Analysis	Analyse indicators of compromise and behaviour and other evidence of adversary presence or activity
	Situational Awareness	Correlate and analyse data to monitor the ongoing organisation wide situation al awareness

## ResilMesh Platform Architecture Concept

A SOAPA is defined as<sup>vi</sup>

*“A multi-layered heterogenous architecture designed to integrate disparate security analytics and operations tools. This architecture glues incongruent security analytics tools together to improve threat detection, and then tightly-couples security analytics with operations tools to accelerate and automate risk mitigation and incident response”,*

and is depicted in Figure 3:

### SOAPA: Security Operations and Analytics Platform Architecture

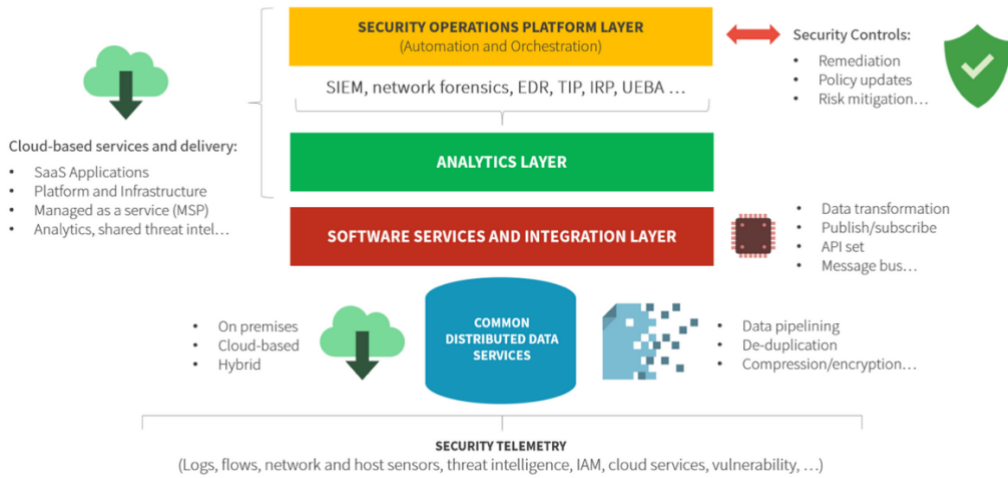


Figure 3 SOAPA logical diagram

As can be seen from the definition and the goals of the SOAPA concept align very closely with the foundational Resilmesh goal outlines in the previous chapter. The Resilmesh mapping to the general SOAPA logical architecture of Figure 3 is shown below in Figure 4.

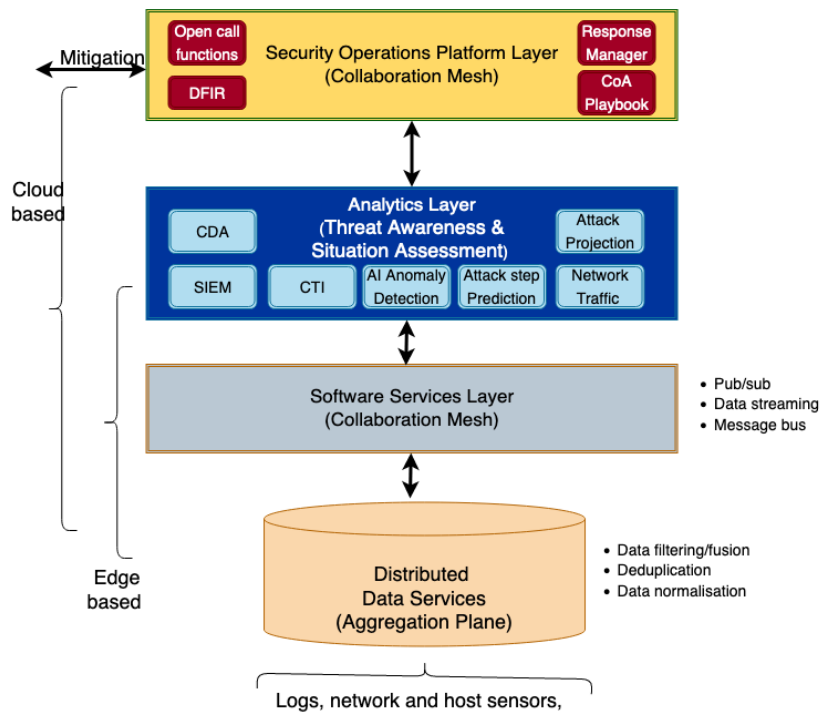


Figure 4 Resilmesh SOAPA architecture

A preliminary version of the Resilmesh functional architecture is shown in which may

further aid in understanding the Resilmesh concept. Be aware however that this will evolve.

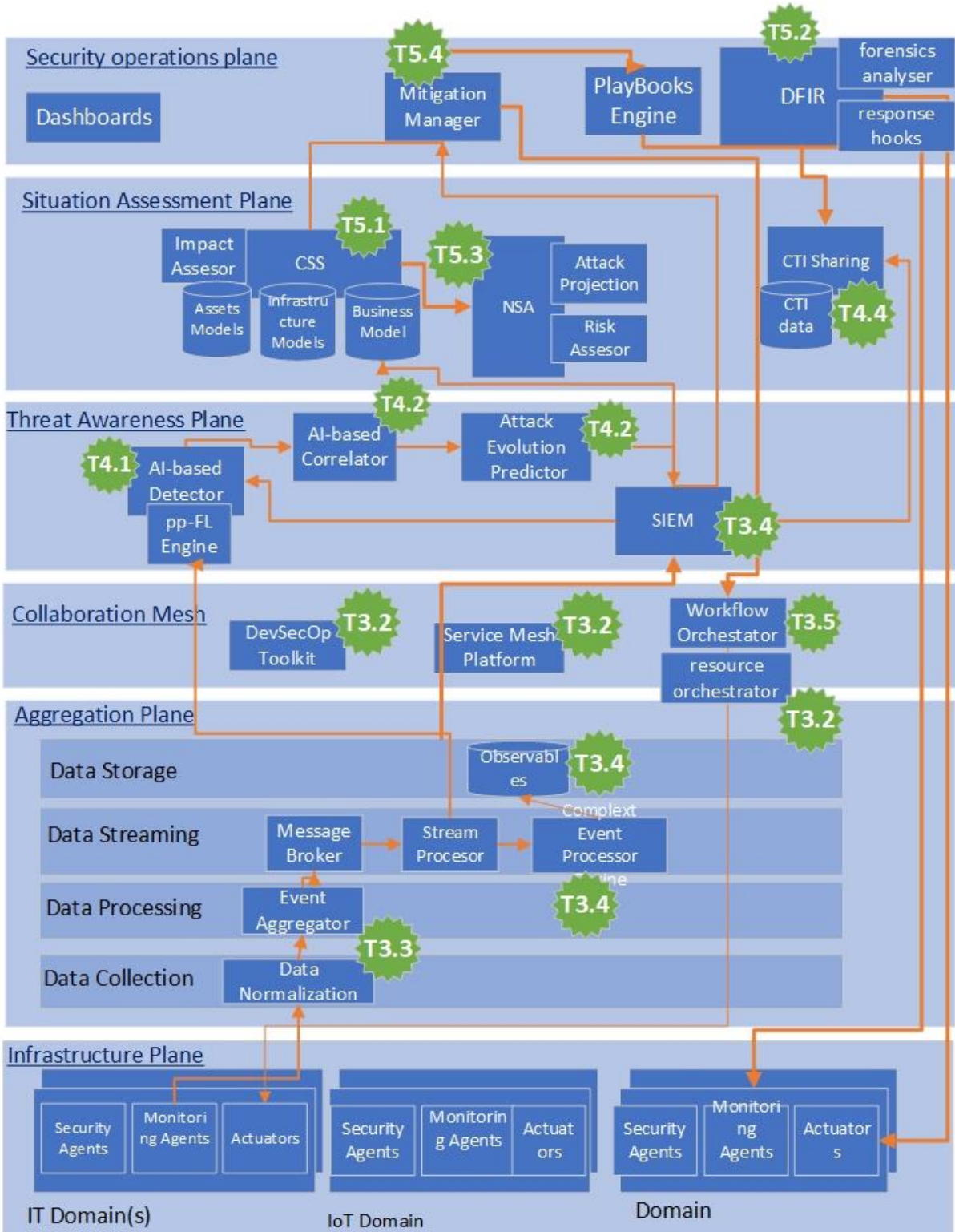


Figure 5 Resilmesh preliminary functional architecture

## Implementation Phasing

There are two phases for delivery of the SOAPA framework as addressed in WP3. These are :

- Phase 1 – M15 – MS3
- Phase 2- M24 – MS6

The intention is to focus on the ‘happy path’ scenarios in Phase 1 and to consider the corner cases during Phase 2. However more work remains to be done to decide exactly what should go in each phase. The planning here is therefore preliminary and will be refined as requirements become clearer.

## Aggregation and Collaboration Mesh Layers

### Requirements for these layers

*Table 2 Non Functional Requirements*

Slogan	Explanation	System Component
Ensure support for heterogeneous critical infrastructures	Ensure ResilMesh can support organisations of different types, domains and size.	System
Ensure support for flexible function allocation and composition	Ensure that security functions can be flexibly located where required across the dispersed IS infrastructure.	System
Cooperative security ecosystem	Ensure that heterogeneous security tools can be integrated into a cooperative ecosystem	System
Enable fusion of sensor security data	Combine monitoring data from different sources at different points in the system as well as with externally provided CTI	Aggregation
Distributed service deployment	Provide a capability to deploy service components across the system distributed infrastructure. (support for edge)	Mesh
Support event processing flows	Enable composition of end-to-end security event processing data flows.	Aggregation Mesh
Standardised data collection format	Incoming security event data must be normalized to an industry standard format for sharing between tools.	Aggregation

Distributed AI-based Anomaly Detection	Deploy and train models for anomaly detection as required across a data processing pipeline including on edge and cloud.	Analytics
Federated Learning	Provide support for deployment of federated learning frameworks for edge anomaly detection. FL shall support zone awareness.	Analytics
Security Event Message Streaming	Provide support for message streaming of security event.	Analytics
Security Event Aggregation	Provide support for security event collection, routing, logging, filtering storage and routing.	Aggregation

### Reference Implementation of these layers

We can reimagine these layers from the above logical views in Figure 4 and Figure 5 and to a more implementation focused view in the figure below

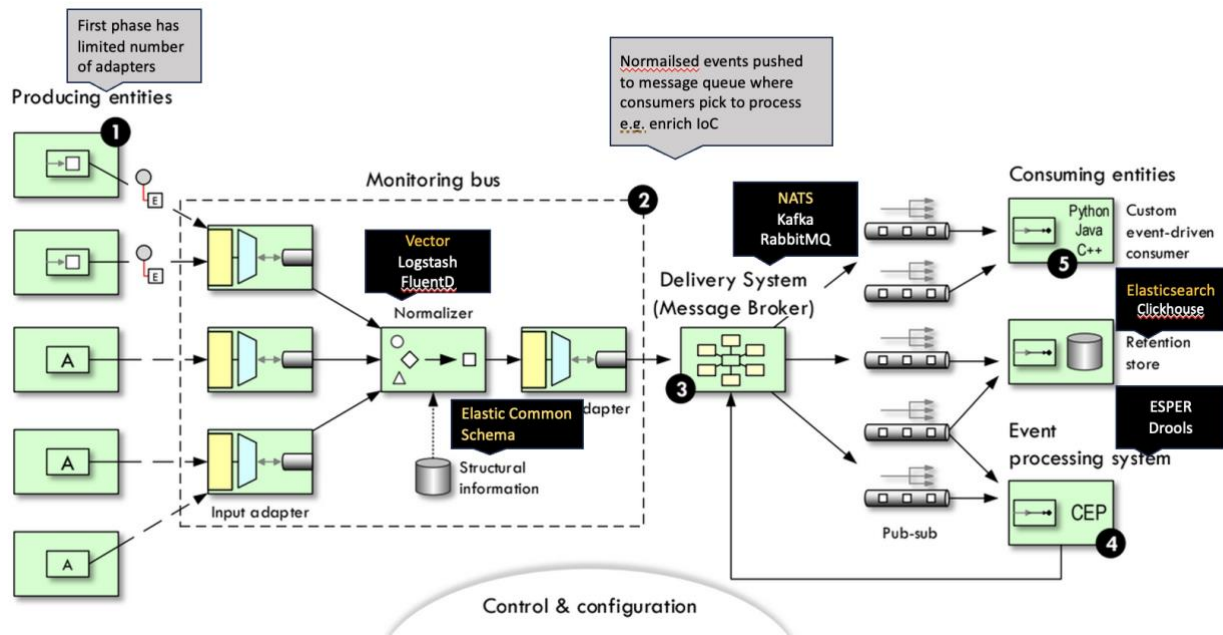


Figure 6 Resilmesh Aggregation and Service Layer Implementation Architecture

Here you can see the (1) events logs feeding the (2) monitoring bus which normalizes them and push to the (3) message broker, then the (4) event processing system such as the enrichment can fetch the events, process and push them back to another queue in the (3) message broker, this forth and back interaction between the (4) event processing and the (3) message broker goes on and on until all the necessary



contextual information is added to the events, finally, the events are stored in the (5) retention store.

As well as showing the implementation architecture the diagram shows some alternative technologies to implement the components. A more complete list is given in the table. Where applicable, preferred components are highlighted in bold:

*Table 3 Candidate Technologies for Aggregation and Collaboration Mesh Layers*

Component	Tools	Comment
Sensors	Zeek, Suricata, Anomaly Detector models	A number of such sensors will be used
Collection Agents	Beats, nxlog, Windows Event Log, syslog-ng, rsyslog	A number of such collection agents will be used
Normalisation	<b>Vector</b> , FluentD, Logstash	<p>Vector is more lightweight with respect to both CPU and memory, setup, and deployment. This choice makes sense primarily in the context of the project early stages.</p> <p>With the Vector VRL language, the data transformations are easier to prototype and test. Monitoring support is built-in and the data pipelines can be debugged in real-time. Even though Vector does not support as much connectors and adapters as Logstash, it supports the currently needed ones well, e.g. Elasticsearch, or Clickhouse.</p>
Messaging	<b>NATS</b> , RabbitMQ, Kafka	<p>Nats is more lightweight with respect to both CPU and memory, setup, and deployment. This choice makes sense primarily in the context of the project early stages</p> <p>Nats also offers an inbuilt service mesh creation capability that could be leveraged to provide collaboration mesh capabilities.</p> <p>Thanks to the high interoperability of both Kafka and NATS, they can essentially serve as drop-in replacements for each other. Only minor changes might be necessary at the producer/consumer side, since the passed-around messages are essentially binary BLOBs.</p>
Stream Processing	Spark, Storm, Flink, Kafka Streams, Nifi	Ph1/2 (unclear if needed )

Enrichment	Silent Push	Default
Retention	Elasticsearch, Clickhouse , S3	<p>Elasticsearch was designed to power better search. It can efficiently return search results, accounting for things such as spelling mistakes. This makes it very useful for full-text search, log and event data analysis, real-time application monitoring, analytics – <b>This is more in line with the Resilmesh use cases</b></p> <p>ClickHouse is an open source columnar database management system designed for high-performance online analytical processing (OLAP) tasks, s known for its ability to process large volumes of data in real-time, providing fast query performance and real-time analytics. Its columnar storage architecture enables efficient data compression and faster query execution, making it suitable for large-scale data analytics and business intelligence applications.</p> <p><b>Clickhouse might be used in a later project phase such as in the open calls e.g. for real time event streaming aggregation.</b></p>

## Deployment and Orchestration of Distributed Components

This relates to:

**Automated deployment of functional component** where desired across the entire framework .This is the most basic capability required and is valid for all scenarios and cases. A sub case is deployment of distributed anomaly detectors and associated learning framework– distributed anomaly detectors can be considered as a variation of an Intrusion Detector System or other sensor and also an example of edge AI.

**Creation of event processing pipelines.** It is pertinent to ask what we mean by pipeline. A number of alternatives could be envisaged,

- *Aggregation pipelines* i.e. chained functional components for security event collection, routing, logging, filtering storage and routing. Tool support will be needed to define these pipelines, deploy the functional components on the related computing assets as well configuring the individual components and the message routing etc.
- *Event processing streams* based on Spark or such. Stream processing components typically receive data from the messaging platform, perform some computation and write the result back to the messaging platform. It is unclear at this stage if we will need this functionality.

The **possible redeployment or migration** of Resilmesh functional components. This relates to requirements for dynamic positioning and collaboration mesh) and is aimed to improve the resilience of the framework itself. These requirements are self-imposed, and it is probably unlikely that we will get direct end-user requirements for this.

The table list the main technology candidates:

*Table 4 Candidate Technologies for Deployment*

Component	Tools	Comment
Execution Env.	<b>Docker</b> , Bare metal	Docker enables a more modular deployment. MONT tools may need bare metal/VM
Orchestration	<b>Kubernetes, Kubeedge</b>	Best in class
Service Mesh	<b>Nats, Istio</b>	Nats in built service mesh means we remove the need for an alternate tool.

## Analytics Layer

### Requirements

*Table 5 Analytics Requirements*

Slogan	Explanation	System Component
Cooperative security ecosystem	Ensure that heterogeneous security tools can be integrated into a cooperative ecosystem	System
Security Event Stream Processing	Provide support for real time complex event processing of security event streams.	Analytics
Distributed AI-based Anomaly Detection	Deploy and train models for anomaly detection as required across a data processing pipeline including on edge and cloud.	Analytics

### Cooperative security ecosystems

This is really about **integration of diverse security** tools via:

*Interoperability* through a mix of

- open standards and interfaces – at the analytics layer standards that include STIX, MISP, etc. A high-level architectural approach that considers how interoperability at a number of layers is the OpenXDR standard from OASIS Open Cyber security Alliance<sup>vii</sup>.
- proprietary APIs, and point integrations (e.g., ad hoc integrations between vendors’ tools) e.g. Silentpush and other tools such as HIVE

*Cross-Domain Security Analytics* by synergistically combining the outputs of different analytics tools e.g. the Network Situational Assessment (NSA-T5.3) will use the CI asset model (T5.1). Another example from Gartner is “a security product may have its own native analytics engine that calculates a risk score for a particular session, user or transaction. A broader security analytics and intelligence tool could then use this risk score, applying it in a different context from that originally intended”

**Achieving integration in this is a foundational goal of Resilmesh (as well as the Cybersecurity mesh and SOAPA activities)**

### Consolidated Dashboards

The system will obviously need different user interfaces both GUI and CLI. However we should try to have a single entry point be that a dashboard or launch pad. An example could be the dash board developed for the PROTECTIVE project<sup>viii</sup>

### Tools

*Table 6 Implementation Tools and Technologies for the Analytics layer*

Component	Tools	Comment
Standard protocols.	STIX, MISP	De facto standards
Stream Analytics	<b>Esper</b> , Drools	Esper’s SQL like query syntax makes it easier to learn and work with. Also consortium members are familiar with it.
SIEM	<b>Wazuh</b> , DSIEM, OSSIM	Wazuh offers much more functionality than DSiem- yet less that OSSIM which is overly complex for our needs.
Dashboards	Protective <sup>ix</sup>	

## Operations Layer

Note we do not consider all the operations function here as more work needs to be done as part of the Architecture activities to elaborate the related functional components

### Requirements

Table 7 Operations Layer Non Functional Requirements

Slogan	Explanation	System Component
Cooperative security ecosystem	Ensure that heterogeneous security tools can be integrated into a cooperative ecosystem	System
Consolidated dashboards	Ensure the system user interfaces are linked in a coherent manner. The system shall be accessed from a single, root, interface.	System
SIEM	Provide a security management capability that supports event correlation, alarm generation and viewing	Analytics
Command and Control interoperability	Provide support to enable security tools to exchange command and control information in a standardised format. This includes tools in the SOC as well as between SOC and distributed enforcement points.	Operations Analytics
Command and control collaboration	Ensure that the outputs from system security tools are combined synergistically i.e. so that each tools increases the effect of others	Operations Analytics
Provide orchestration for cyber operations	Ensure that critical cyber operations are automated and course of action playbooks are prepared to ensure collaboration between diverse tools to ensure effective detection, analysis and response.	Operations

## Cooperative security ecosystems

The notes above wrt Interoperability apply here also especially the Open XDR automation layer. The OpenDXL<sup>x</sup>. work is also very relevant- unfortunately this work is not supported and while it morphed into the OCA Ontology that work is also not active any more.

## SIEM

We will use an open source SIEM for event correlation.

## Orchestration

Here we are concerned with open-source orchestration engines

## Tools

Table 8 Candidate Technologies Operations Layer

Component	Tools	Comment
Standard protocols.	<b>OpenC2</b>	De-facto standard solution
SIEM	<b>Wazuh, DSIEM</b>	See previous table.
Dashboards	Protective	-
Orchestration Engine	<b>Temporal, Shuffle</b> ( <i>to be decided after further analysis</i> )	Shuffle support many of the OCA Open XDR standards and is the most likely choice for Resilmesh .

# Annex 1 – Initial Implementation

This annex describes a preliminary proof of concept implementation of the aggregation and collaboration mesh layers of the Resilmesh platform.

## The Technologies

We have mainly 3 technologies composing the lower layers:

### Vector

Vector is the observability pipeline tool; it will be responsible for collecting the logs from different sources and normalizing them.

The log sources, such as Syslog, Windows Log servers, etc have their own format and schema, hence, normalizing will be the process of converting these different formats into a single one, so it will be able to be read by all the components of the framework.

### NATS

NATS is the queue system, our message broker, it's responsible for aggregating the events according to their state such as normalized, enriched, etc.

Every component of the framework will fetch NATS for the specific event they need, process them and push it back to the message broker using the same normalized format, so that the next component can do the same, until the pipeline is finished.

### Elastic

Elastic is our retention storage, it will handle all the events once they are processed by all the components of the pipeline, it's the end node of the framework.

We will take advantage of all the search capabilities Elastic can provide; it will act as the frontend of the project.

## The Pipeline

The pipeline is as described in Figure 6.

## Modules

The extensibility of the framework will rely on the event process system highlighted in the figure 1, any piece of software can implement their own specific logic and push the events back to the message broker as long as the normalized format is respected.

### SLP Enrichment Module

Silent Push will provide the enrichment engine, which will be implemented as a module in the framework. The enriched data will consist of several contextual information, depending on the type of the event (IP, Domain, etc), the most important ones being:

- **Whois:** The registration info of a domain.

- **DGA:** How likely it is that a domain was created by an automated generation algorithm.
- **DNS Records:** All the records found for the domain (A, NS, SOA, etc)
- **Nameserver**
  - **Reputation:** The ratio of blacklisted domains, taken from the total number of domains using a nameserver.
  - **Entropy:** A score that includes recency, frequency, and the number of NS changes.
  - **Density:** How many domains are used by a specific nameserver.
- **ASN**
  - **Name:** The name of the Autonomous System (AS)
  - **Rank:** A ranking of ASNs seen to host threats listed on feeds, calculated using a weighted formula based on the type of threat observed.
  - **Reputation:** The ratio of blacklisted IPs, taken from the total number of IPs that have been observed as being active within an ASN, in the last 30 days.
  - **Takedown Reputation:** A reputation score based on the time it takes for the ASN owner to react to takedown requests related to malicious URLs.
  - **IPs in ASN:** The number of IP addresses available in this ASN.
- **Certificates:** The certificate details for the domain such as issuer, validity, etc.
- **Geolocation:** Geographical location of the IP.
- **Subnet**
  - **Reputation:** The ratio of blacklisted IPs, taken from the total number of IPs that have been observed as being active within a particular subnet in the last 30 days.
  - **Number of IPs:** The number of IP addresses available in this subnet
  - **Active IPs:** The number of IP addresses seen as active A records in this subnet.

A Silent Push enriched domain would look like:



```

{
  "domain_urls": {
    "results_summary": {
      "alexa_rank": 44062,
      "alexa_top10k": false,
      "alexa_top10k_score": 0,
      "dynamic_domain_score": 0,
      "is_dynamic_domain": false,
      "is_url_shortener": false,
      "results": 0,
      "tranco_rank": 42662,
      "tranco_top10k": false,
      "tranco_top10k_score": 0,
      "url_shortener_score": 0
    }
  },
  "domaininfo": {
    "age_score": 0,
    "info": "Domain registered before 20170101",
    "is_new": false,
    "is_new_score": 0,
    "last_seen": 20240105,
    "registrar": "CSC CORPORATE DOMAINS, INC.",
    "whois_age": 13805,
    "whois_created_date": "1986-03-19 05:00:00",
    "zone": "com" },
  "ip_diversity": {
    "asn_diversity": "1",
    "asns": [
      7162
    ],
    "ip_diversity_all": "2",
    "ip_diversity_groups": "2"
  },
  "ns_reputation": {
    "is_expired": false,
    "is_parked": false,
    "is_sinkholed": false,
    "ns_reputation_max": 0,
    "ns_reputation_score": 0,
    "ns_srv_reputation": [
      {
        "ns_server_domain_density": 1509,
        "ns_server_domains_listed": 1,
        "ns_server_reputation": 0
      }
    ]
  },
  "nschanges": {
    "results_summary": {
      "changes_0_7_days": 0,
      "changes_30_90_days": 0,
      "changes_7_30_days": 0,
      "changes_last_30_days": 0,
      "changes_last_7_days": 0,
      "changes_last_90_days": 0,
      "ns_entropy": 0,
      "ns_entropy_score": 0,
      "num_changes_all": 0,
    }
  },
  "sp_risk_score": 0
}

```

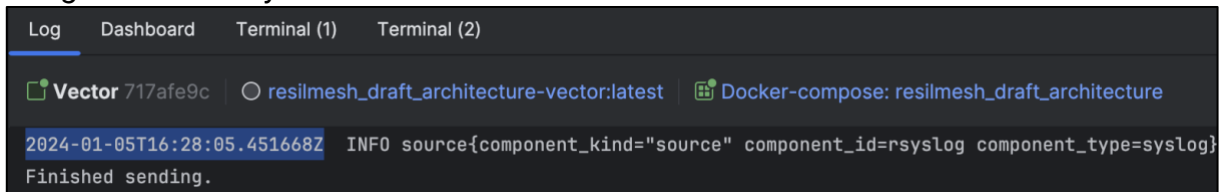
The addition of this module to the framework will be paramount, since will make the events way more relevant, threat analysts can take conclusion just by analysing the enriched data. Also, the other modules can leverage their functionalities relying on the enriched information, it can act as a starting point.

## Framework Demo

In this session, we will show the whole framework in action, simulating an event being processed by the ResilMesh pipeline.

Bear in mind all screenshots below were extracted from a POC instance, built only for demonstration purposes.

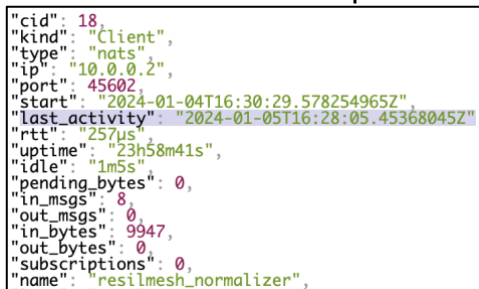
1. A log is received by the Vector server:



```

Log Dashboard Terminal (1) Terminal (2)
Vector 717afe9c | resimesh_draft_architecture-vector:latest | Docker-compose: resimesh_draft_architecture
2024-01-05T16:28:05.451668Z INFO source{component_kind="source" component_id=rsyslog component_type=syslog}
Finished sending.
  
```

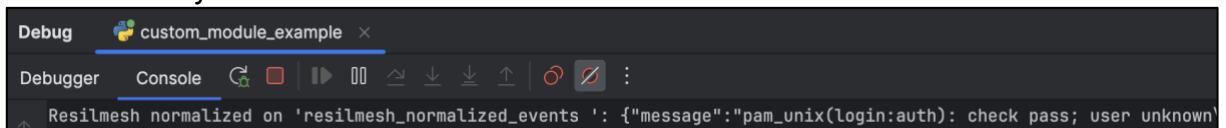
2. Immediately, the same log is normalized and pushed to the NATS "resimesh\_normalizer" queue in the Message Broker:



```

"cid": 18,
"kind": "Client",
"type": "nats",
"ip": "10.0.0.2",
"port": 45602,
"start": "2024-01-04T16:30:29.578254965Z",
"last_activity": "2024-01-05T16:28:05.45368045Z",
"rtt": "257µs",
"uptime": "23h58m41s",
"idle": "1m5s",
"pending_bytes": 0,
"in_msgs": 8,
"out_msgs": 0,
"in_bytes": 9947,
"out_bytes": 0,
"subscriptions": 0,
"name": "resimesh_normalizer",
  
```

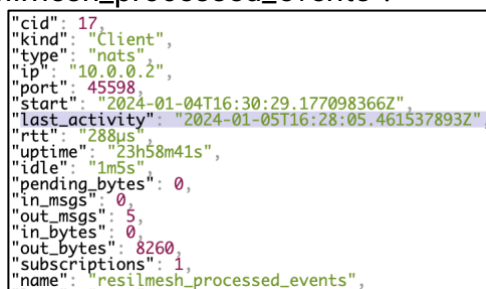
3. A module fetches the "resimesh\_normalizer" queue, retrieve the event and process it successfully:



```

Debug custom_module_example x
Debugger Console
Resimesh normalized on 'resimesh_normalized_events': {"message":"pam_unix(login:auth): check pass; user unknown"}
  
```

4. Which is pushed back to the Message Broker into another queue "resimesh\_processed\_events":



```

"cid": 17,
"kind": "Client",
"type": "nats",
"ip": "10.0.0.2",
"port": 45598,
"start": "2024-01-04T16:30:29.177098366Z",
"last_activity": "2024-01-05T16:28:05.461537893Z",
"rtt": "288µs",
"uptime": "23h58m41s",
"idle": "1m5s",
"pending_bytes": 0,
"in_msgs": 0,
"out_msgs": 5,
"in_bytes": 0,
"out_bytes": 8260,
"subscriptions": 1,
"name": "resimesh_processed_events",
  
```

4.1. This step can be repeated by other modules: pull, process, push back...

5. Finally, after the whole flow, we have the processed event stored into Elastic:

```

→ localhost:9200/resilmesh-processed-events/_search?from=5300&size=2
// 20240105163822
// http://localhost:9200/resilmesh-processed-events/_search?from=5300&size=2

{
  "took": 35,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 5301,
      "relation": "eq"
    },
    "max_score": 1.0,
    "hits": [
      {
        "_index": "resilmesh-processed-events",
        "_id": "zFF0ZowBZL31Y1688Dj",
        "_score": 1.0,
        "_ignored": [
          "message.keyword"
        ],
        "source": {
          "message": {
            "\message": "\pam_unix(login:auth): check pass; user unknown\u0000<85>Jan 5 16:27:46 login[821]: pam_unix(login:auth): check pass; user unknown\u0000<85>Jan 5 16:27:49 login[821]: FAILED LOGIN (1) on '/dev/pts/1' FOR 'UNKNOWN', Authentication failure\u0000<85>Jan 5 16:27:53 login[821]: FAILED LOGIN (2) on '/dev/pts/1' FOR 'UNKNOWN', Authentication failure\u0000<85>Jan 5 16:27:54 login[821]: pam_unix(login:auth): check pass; user unknown\u0000<85>Jan 5 16:27:57 login[821]: pam_unix(login:auth): check pass; user unknown\u0000<85>Jan 5 16:28:02 login[821]: pam_unix(login:auth): check pass; user unknown\u0000<85>Jan 5 16:28:05 login[821]: pam_unix(login:session): close_session - error recovering username\u0000<85>Jan 5 16:28:05 login[821]: PAM 4 mo\u0000<85>Jan 5 16:28:05 login[821]: PAM service(login) ignoring max retries; 5 > 3\u0000"
          }
        },
        "source_type": "nats",
        "subject": "resilmesh-processed-events",
        "timestamp": "2024-01-05T16:28:05.461663971Z"
      }
    ]
  }
}

```

# References

---

- <sup>i</sup> <https://www.gartner.com/en/information-technology/glossary/cybersecurity-mesh>
- <sup>ii</sup> <https://github.com/opencybersecurityalliance/oxa>
- <sup>iii</sup> <https://csrc.nist.gov/pubs/sp/800/160/v2/r1/final>
- <sup>iv</sup> IEC/TS 62443-1-1 Industrial Networks – Terminology, Concepts and Models <https://bit.ly/3yL1g0y>
- <sup>v</sup> ENISA , Zoning and conduits for railways, <https://bit.ly/3A4uMjm>
- <sup>vi</sup> <https://www.techtarget.com/esg-global/blog/openc2-can-accelerate-security-operations-automation-and-orchestration/>
- <sup>vii</sup> <https://github.com/opencybersecurityalliance/oxa>
- <sup>viii</sup> <https://gitlab.com/protective-h2020-eu/protective-node/-/wikis/user-guide/User-Guide#pdh>
- <sup>ix</sup> <https://gitlab.com/protective-h2020-eu>
- <sup>x</sup> <https://www.opendxl.com/index.php?media/141-opendxl-integration-planning-pdf/>