**Grant Agreement No.:** 101095542
**Call:** HORIZON- HLTH-2022-IND-13
**Topic:** HORIZON-HLTH-2022-IND-13-01
**Type of action:** HORIZON-RIA

CYLCOMED

Cyber-security toolbox
for connected medical devices

# *D5.3 CYLCOMED Final Toolbox Package*

Revision: v.0.16

| | |
|---|---|
| Work package | WP5 |
| Task | T5.1, T5.2, T5.3, T5.4, T5.5 |
| Due date | 31/08/2025 |
| Submission date | 31/08/2025 |
| Deliverable lead | EVIDEN |
| Version | 0.16 |
| Authors | (EVIDEN) |
| Reviewers | Hrvoje Ratkajec (XLAB) |

| | |
|---|---|
| Abstract | Deliverable D5.3 presents the final CYLCOMED Cybersecurity Toolbox, consolidating advancements across four layers: AI-driven data collection/analysis (LADS/LOMOS), device integrity/management (OTA/Mender), identity/access/protection (LuS4MED/C-OPA/FE4MED), and security dashboard. It details architecture, innovations, pilot integrations, outcomes and lessons learned for secure CMD deployment. |
| | |
|---|---|
| Keywords | Cybersecurity, Toolbox, CMD |

# DOCUMENT REVISION HISTORY

| Version | Date | Description of change | List of contributor(s) |
|---|---|---|---|
| V0.1 | 04/07/2025 | Initial table of contents | Esteban Armas (EVIDEN) |
| V0.2 | 04/07/2025 | Section 4 — initial draft | Ricardo Ruiz (RGB) |
| V0.3 | 04/07/2025 | Section 4 — updates | Ricardo Ruiz (RGB) |
| V0.4 | 07/07/2025 | Table of contents — update | Esteban Armas (EVIDEN/ATOS) |
| V0.5 | 18/07/2025 | Section 2 — Section 3 - initial draft | Hrvoje Ratkajec (XLAB) |
| V0.6 | 01/08/2025 | Deliverable First Review | Hrvoje Ratkajec (XLAB) |
| V0.7 | 01/08/2025 | Section 5 — update | Gabriele Cerfoglio (MARTEL) |
| V0.8 | 05/08/2025 | Section 5 — additional content | Juan Carlos Pérez Baun (EVIDEN/ATOS) |
| V0.9 | 05/08/2025 | Section 5 — further update | Juan Carlos Pérez Baun (EVIDEN/ATOS) |
| V0.10 | 06/08/2025 | Document review and comments | Duško Milojević (KU Leuven) |
| V0.11 | 12/08/2025 | Sections 1 & 2 | Esteban Armas (EVIDEN/ATOS) |
| V0.12 | 18/08/2025 | Sections 6 | Esteban Armas (EVIDEN/ATOS) |
| V0.13 | 22/08/2025 | Sections 3 | Jose Antonio Nicolas Navarro<br>Hristo Koshuntanski (EVIDEN/ATOS) |
| V0.14 | 25/08/2025 | Deliverable Second Internal Review | Hrvoje Ratkajec (XLAB) |
| V0.15 | 25/08/2025 | Update : References, Section 6 and Section 7 | Esteban Armas Vega (EVIDEN/ATOS) |
| V0.16 | 28/08/2025 | Update of all sections | All partners involved |
| | | | |



Funded by Horizon Europe Framework Programme of the European Union

## Disclaimer



## Copyright notice



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

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


Funded by Horizon Europe Framework Programme of the European Union

# Executive summary

This public deliverable (D5.3) concludes WP5 “Cybersecurity Toolbox Design and Implementation” by presenting the final architecture, packaging, and integration status of the CYLCOMED Toolbox. This document is the final iteration of the CYLCOMED Cybersecurity Toolbox, developed under WP5, to enhance the cybersecurity of connected medical devices. The document provides updates across the four core layers of the Toolbox:

- **Data Collection and Analysis**: This layer includes two AI tools—LADS (network anomaly detection) and LOMOS (log-based anomaly detection). Final enhancements to LADS and LOMOS include scalable burst/low-rate DoS detection, GPU-optimised self-supervised training for anomaly scoring, and full integration with hospital logs/systems for real-time threat detection.
- **Device Integrity and Service Management**: This layer comprises tools such as Mender (for OTA updates) and the RGB external board (for SoC Integrity Check). Completed tools include Mender for OTA updates, SoC Integrity Check on Raspberry Pi, and hybrid deployments for legacy devices, validated with IEEE 11073 interoperability and TRL 6 achievement in Pilot 1 simulations.
- **Identity & Access Management:** This layer features LuS4MED (SSI-based authentication), C-OPA (policy-driven authorisation), and FE4MED (data protection via CP-ABE encryption). LuS4MED is fully implemented for SSI-based VC issuance/verification, integrated with C-OPA for policy-driven authorisation; FE4MED provides modular CP-ABE encryption/decryption, ensuring policy-driven access and GDPR-compliant data protection across pilots.
- **Security Dashboard:** This layer consists of the unified dashboard for data visualisation. The fully operational dashboard now includes real-time visualisation, alerts, and integration with LADS & LOMOS for unified monitoring and response.

Overall, the CYLCOMED Toolbox now provides a practical, modular foundation for CMD cybersecurity that fits real hospital constraints, aligns with GDPR and MDR/IVDR regulations, and can be extended. This deliverable captures the final, high-level architecture and integration status needed for wider adoption, with references to previous project deliverables for requirements, pilot context, and validation results



Funded by Horizon Europe Framework Programme of the European Union

# Table of Contents





Funded by Horizon Europe Framework Programme of the European Union

Funded by Horizon Europe Framework Programme of the European Union

# List of Figures





Funded by Horizon Europe Framework Programme of the European Union

# List of Tables





Funded by Horizon Europe Framework Programme of the European Union

# Abbreviations

| | |
|---|---|
| **ABE** | Attribute-Based Encryption |
| **AI** | Artificial Intelligence |
| **API** | Application Programming Interface |
| **AWS** | Amazon Web Services |
| **CMD** | Connected Medical Devices |
| **CP** | Ciphertext Policy |
| **CPU** | Central Processing Unit |
| **DLT** | Distributed Ledger Technology |
| **FE4MED** | Functional Encryption for Medical Data |
| **FTP** | File Transfer Protocol |
| **FM** | FlowMeter |
| **GDPR** | General Data Protection Regulation |
| **GUI** | Graphical User Interface |
| **GPU** | Graphics Processing Unit |
| **HIL** | Hardware in Loop |
| **HIS** | Hospital Information System |
| **HW** | Hardware |
| **IAM** | Identity and Access Management |
| **ICU** | Intensive Care Unit |
| **IaC** | Infrastructure as Code |
| **IdM** | Identity Management |
| **IoT** | Internet of Things |
| **IP** | Internet Protocol |
| **JSON** | JavaScript Object Notation |



Funded by Horizon Europe Framework Programme of the European Union

| | |
|---|---|
| **JWT** | JSON Web Token |
| **LADS** | Live Anomaly Detection System |
| **LOMOS** | Log Monitoring System |
| **LuS4MED** | Ledger uSelf for Medical Data |
| **MHP** | Mediaclinics Health Platform |
| **NLP** | Natural Language Processing |
| **NMT** | Neuromuscular Transmission |
| **OPA** | Open Policy Agent |
| **OTA** | Over the Air |
| **PCAP** | Packet Capture |
| **REST** | REpresentational State Transfer |
| **RBPI** | Raspberry Pi |
| **SIEM** | Security Information and Event Management |
| **SIL** | Software in Loop |
| **SIOP** | Self-Issued OpenID Provider |
| **SSI** | Self-Sovereign Identity |
| **TCP** | Transmission Control Protocol |
| **VC** | Verifiable Credential |
| **VP** | Verifiable Presentation |
| **vNIC** | Virtual Network Interface Card |



Funded by Horizon Europe Framework Programme of the European Union

# 1 Introduction

This deliverable (D5.3) reports the final design and integration status of the CYLCOMED Toolbox, building upon previous deliverables (D5.1 [1], D5.2 [2]), at the end of WP5 Cybersecurity Toolbox Design and Implementation. It consolidates the architecture, component specifications, deployment artefacts, and interoperability profiles that enable secure, privacy-preserving operation of Connected Medical Devices (CMDs) in both hospital and telemedicine realms.

D5.3 documents the final versions of the toolbox components, LADS (network anomaly detection), LOMOS (log monitoring), Connected Medical Devices & Services Management Tools (incl. OTA updates), Identity & Access Management (LuS4MED, C-OPA), Data Protection (FE4MED), Device Integrity (RGB external module), and the Security Dashboard. It captures the implementation outcomes during the integration/feedback loops with the two pilots (Pilot 1: Cybersecurity in Hospital Equipment and Pilot 2: Cybersecurity for Telemedicine Platforms), better described in the deliverables D6.1 [3] and D6.2 [4], and aligns them with applicable regulatory, ethical and security requirements identified earlier in the project (D3.1 [5], D3.2 [6])

## 1.1 Purpose and scope of this document

D5.3 provides a general public description of the CYLCOMED Toolbox. It explains the architecture of the toolbox, how each component works, how they interoperate, and how they are packaged and deployed in on-premise and hybrid environments.

In scope.

- Final system high-level architecture for the toolbox across Pilots 1 and 2.
- Component design and implementation updates for T5.1–T5.5:
    - T5.1: AI-based log monitoring (LOMOS) and network anomaly detection (LADS)
    - T5.2: Connected Medical Devices & Services Management (incl. OTA)
    - T5.3: Identity & Access Management (LuS4MED, C-OPA) and Data Protection for CMDs (FE4MED)
    - T5.4: CMDs integrity (RGB external module)
    - T5.5: Cybersecurity Dashboard
- Packaging & deployment artefacts and supported deployment models.
- Implementation lessons learned and final limitations/known issues relevant for exploitation and scale-up.

Out of scope.

- Clinical protocols and patient-facing procedures (reported in WP6 Integration and Validation with Real-world Applications -Pilots deliverables).
- Full legal/ethical analysis and conformity assessment dossiers (covered in WP2).
- Pilot validation results beyond the implementation evidence (summarised only where needed; full results are in D6.2 [4]/D6.3 [7])



Funded by Horizon Europe Framework Programme of the European Union

## 1.2 Deliverable structure

D5.3 is the final deliverable of WP5. It is organised to give a clear, high-level view of the CYLCOMED Toolbox at the end of implementation, without exposing sensitive internals. Having said that, the document is structured as follows:

- Section 2 — CYLCOMED Toolbox: Final Architecture.
  Presents the Toolbox's final architecture, its components, and supported deployment options across Pilots 1 and 2
- Sections 3–6 — Toolbox Components
  Present the key features and innovations of each component within the toolbox. Additionally, detail the integration outcomes within the pilots, and at the end of each section, lessons learned are presented.
- Section 7 — Conclusions and Future Work
  Present the milestones achieved during the development of WP5, and based on the achieved milestones, propose future work to keep scaling up the solution of CYLCOMED.
- Section 8 — References
  Sources and previous project deliverables as appropriate.

## 1.3 Relation to Previous Deliverables

This deliverable relates to prior outputs as follows:

- D3.1 & D3.2 - Requirements and specifications: Provided the baseline functional and non-functional requirements that guided the Toolbox architecture, scope, and priorities across T5.1–T5.5.
- D2.2 - Ethical, Legal and Data Protection Requirements: Established the constraints that guided the design of the toolbox.
- D5.1 - Toolbox prototype (first iteration): Established the initial component set, packaging approach, and early interfaces.
- D5.2 - Toolbox prototype (second iteration): Expand the features and further develop the interoperability interfaces and integration blueprints to enable continuous integration and testing within the project pilots.
- D6.1 - Pilots design: Defined the pilot environments and constraints that shaped deployment choices, hardening measures, and acceptance criteria used in D5.3.
- D6.2 - Pilot reports (initial phase): Supplied implementation evidence and validation feedback from simulated and real-world settings.



Funded by Horizon Europe Framework Programme of the European Union

# 2 CYLCOMED Toolbox: Final Architecture

This section consolidates the third iteration of the CYLCOMED Toolbox architecture at the end of WP5. It briefly outlines how components fit together across Pilot 1 and Pilot 2, the high-level data flows among them, and the deployment options adopted.

## 2.1 Evolution of the Toolbox Architecture

Throughout WP5, the architecture evolved through three iterations, reflecting a consistent development from basic prototypes to a fully integrated, deployable system focused on healthcare cybersecurity constraints. The initial prototype, as reported in D5.1 [1], established core components independently, including LADS, LOMOS, LuS4MED, FE4MED. Containerised packaging was validated on dedicated VMs, with Pilot 2 exploring a cloud-first approach in a partner-managed sandbox to align with initial D6.1 planning [2] for telemedicine interoperability. Early integrations focused on one-way telemetry, routing logs and network flows to analytics modules without cross-tool correlation, providing a baseline for isolated tool testing.

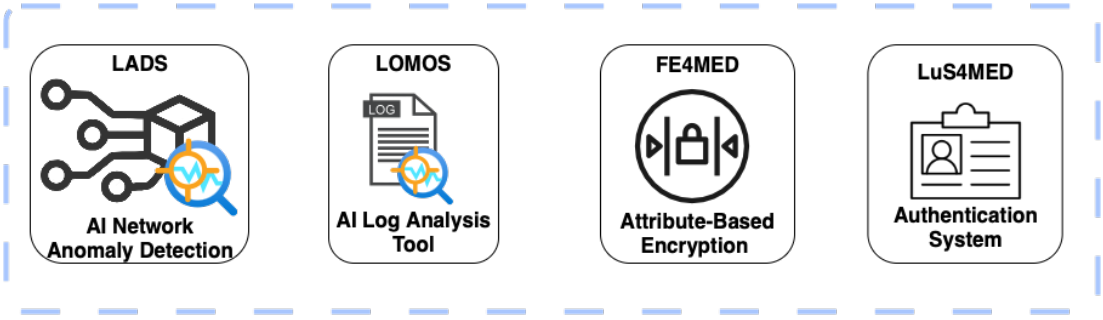


*Figure 1 Early Version of Toolbox Components*

The second iteration, detailed in D5.2 [2], marked a significant advancement. LADS and LOMOS were aligned, enabling joint anomaly narratives that enhanced detection accuracy. FE4MED was modularised into edge encryptors (RBPI for Pilot 1 and an Android library for Pilot 2), complemented by a server-side decryptor/service. LuS4MED's SSI flows were drafted for credential issuance and presentation, with C-OPA positioned as the policy enforcement proxy. OTA management via Mender was adopted for secure device and service updates with rollback capabilities in controlled environments. Critically, the architecture shifted from cloud-first to hybrid for Pilot 1, and on-prem and isolated deployments for Pilot 2 hospital environments, addressing regulatory and ethical constraints per D6.1 [3] and initial feedback in D6.2 [4], ensuring compliance with GDPR/MDR while maintaining scalability.



Funded by Horizon Europe Framework Programme of the European Union

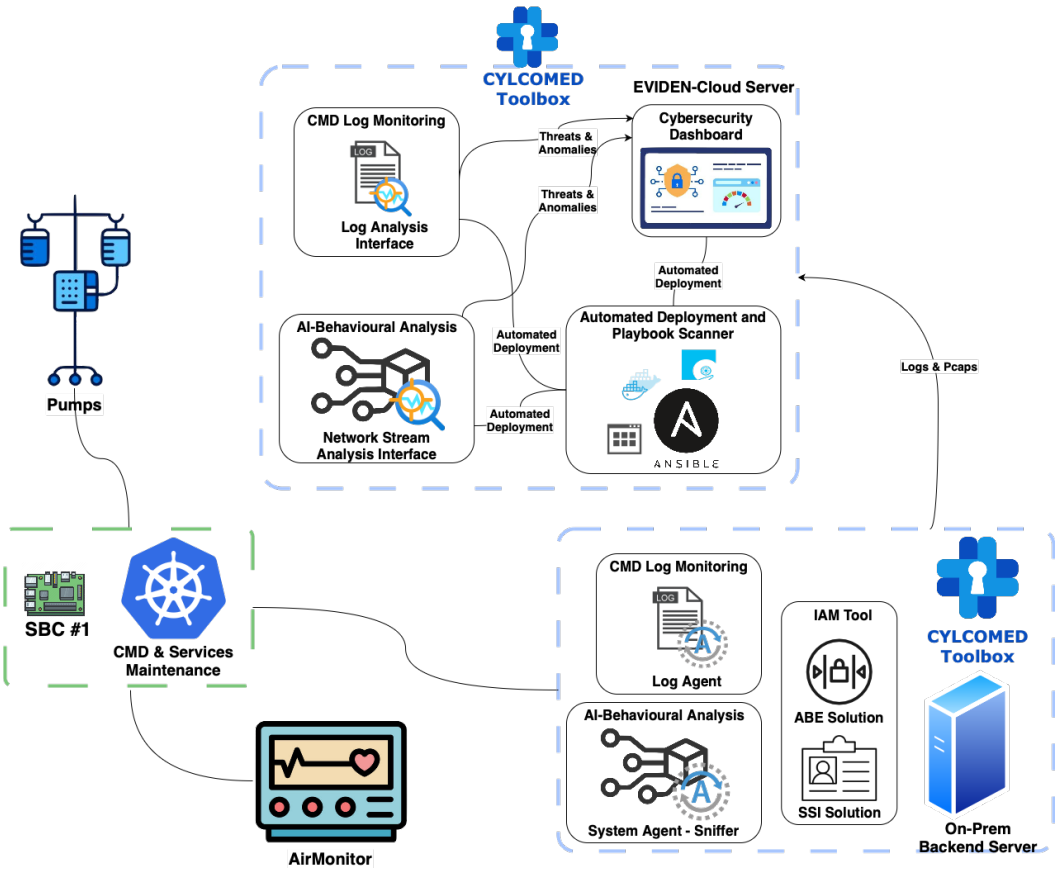


*Figure 2 Final Architecture - Hybrid Deployment - Pilot 1 [2] [4]*

This final integrated baseline, presented in this document, achieves end-to-end orchestration of the toolbox. LADS and LOMOS publish events to the Dashboard, which alerts in real time. Identity and data protection are enforced comprehensively through LuS4MED (SSI; Agent API, first in Section 5.1), feeding into C-OPA (policy evaluation and token checks; Policy/Bundle API, first in Section 5.2) and onward to FE4MED (encrypt/decrypt REST API, first in Section 5.3). Device integrity reporting from the RGB external board (Integrity Report API, first in Section 4.1) now surfaces directly into the Dashboard (Section 6). Deployment methods have been used for on-prem and hybrid validation setups, automated via containers and playbooks, and will be validated across D6.3 real/simulated scenarios [7]. This evolution underscores a cybersecurity-first engineering approach, prioritising modularity for legacy compatibility, zero-trust principles for data flows, and resilience against evolving threats like those in ICU simulations or telemedicine transits.



Funded by Horizon Europe Framework Programme of the European Union

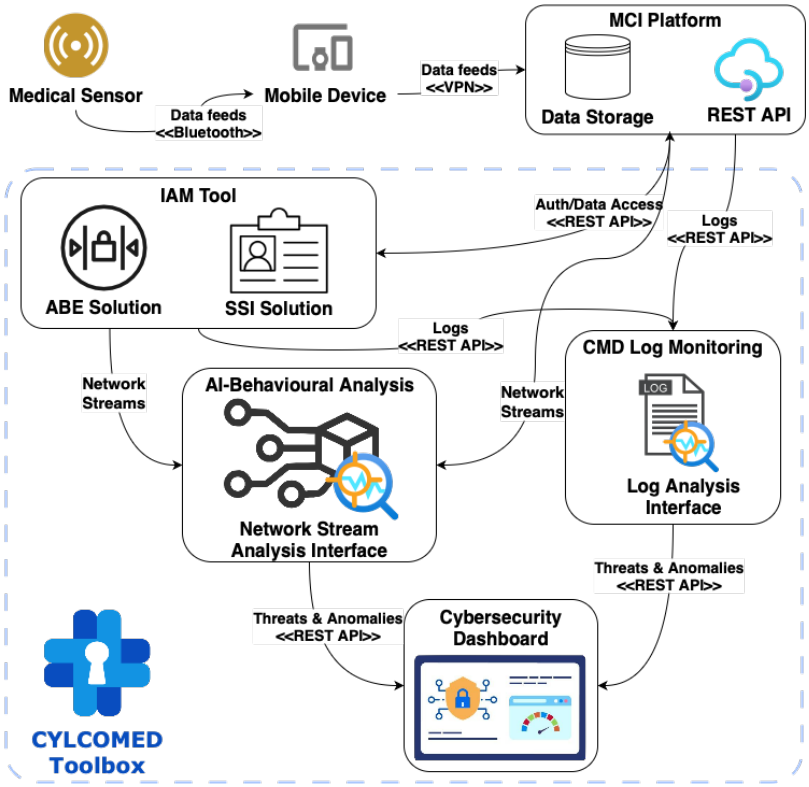


*Figure 3 Final Architecture - On-Prem Deployment - Pilot 2 [2] [4]*

## 2.2 Key Features and Innovations

The toolbox uses a layered design to provide specific protections. It supports integration between components and focuses on practical use in CMD security.

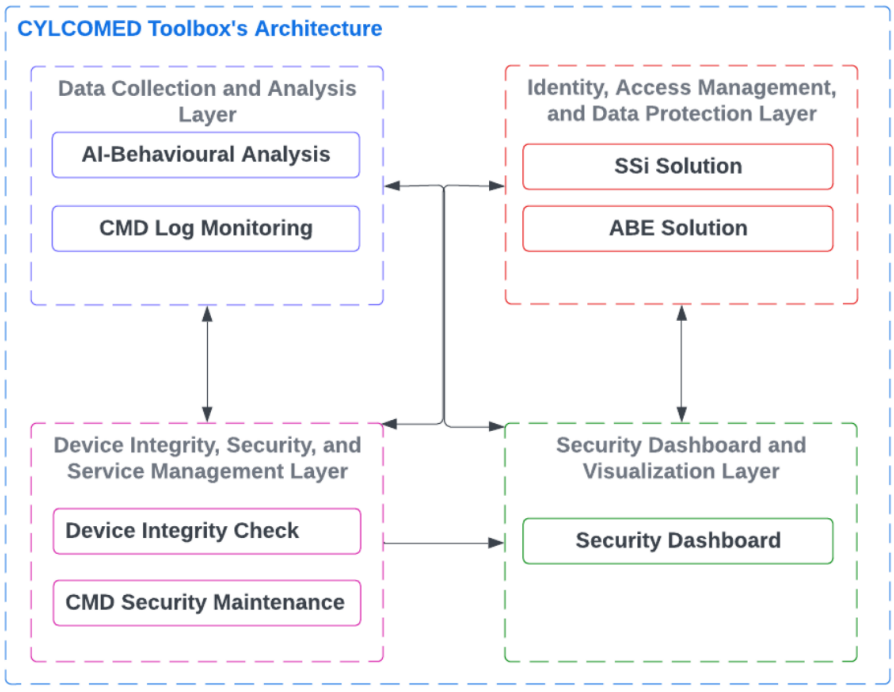


*Figure 4 Toolbox layered design [1]*



Funded by Horizon Europe Framework Programme of the European Union

The Data Collection & Analysis layer is twofold; on one hand, the LADS uses unsupervised AI to detect anomalies in network traffic, and then it sends those events to the Dashboard. On the other hand, LOMOS uses NLP to detect anomalies in system and application logs. It analyses fields like time, source, and context, and if something is detected, it sends that event to the Dashboard. Combining network and log analysis reduces false positives and helps explain incidents without predefined rules.

The Device Integrity, Security & Service Management layer includes the RGB External Board for Device Integrity Check. This Linux module connects to legacy devices like Vision Air without changing their firmware. It sends Integrity Reports using its API. CMD Security Maintenance uses OTA and IaC for updates and deployment with Mender. It exposes endpoints through Mender Management APIs in test setups. This keeps certifications intact under MDR and supports IEEE 11073 for integration. It also allows updates to old ICU equipment without full recertification.

In the Identity, Access Management & Data Protection layer, LuS4MED handles SSI. It issues and verifies Verifiable Credentials using the LuS4MED Agent API, with a mobile wallet for users. C-OPA is the authorisation proxy using Envoy and OPA with Rego policies. Its Policy & Bundle API checks tokens and roles to control access to services like FE4MED. FE4MED uses CP-ABE encryption for attribute-based access. Its Encrypt/Decrypt REST API works at the edge (RBPI/Android) and server for decryption by allowed roles. These parts provide full data protection with decentralised identity. This reduces risks in telemedicine data sharing.

The Cybersecurity Dashboard collects anomalies from LADS/LOMOS, integrity alerts, and authorisation results. It shows trends, alerts, and logs in one view for hospital IT teams. It fits on-prem setups and hybrid environments.

The resulting system scales across the two validated deployment patterns (o-prem & hybrid). In Pilot 1, the toolbox surrounds the device-integrity module and consumes logs and network traces generated by the testbed; curated artefacts can be exported for model refinement while keeping the loop closed within the lab. In Pilot 2, the full stack runs on-premise, FE4MED/LuS4MED/C-OPA gate clinical access, and a separate virtual environment allows safe stress and adversarial testing without touching hospital production systems. Throughout, the architecture aligns with GDPR [8], MDR/IVDR [9] [10] cybersecurity expectations and ISO 14971 [11] risk-management practices.

## 2.3 Summary

After two iterations, and following all the feedback and recommendations, the final CYLCOMED architecture is a set of building blocks that work together and can run on hospital servers (on-premise) or in hybrid setups. AI-driven detection systems (LADS + LOMOS) provide broad coverage of behavioural anomalies; SSI, policy enforcement and CP-ABE ensure that clinical data are accessed only under verifiable identity and explicit authorisation; and OTA/IaC plus the device-integrity gateway closes the loop on secure lifecycle management for both modern and legacy equipment. A unified Security Dashboard ties all together into actionable situational awareness.

At the end, the current toolbox architecture provides a solution that is tailored for CMD environments, aligned with the GDPR and MDR/IVDR regulations and provides a modular system that can be adapted on any scenario of deployment.



Funded by Horizon Europe Framework Programme of the European Union

# 3 Data Collection and Analysis Layer

## 3.1 Advancements in AI-Behavioural Analysis

The focus of AI-behavioural analysis in CYLCOMED is on the detection of anomalies on network traffic in connected devices' medical platforms. The project adopted and extended an asset developed by ATOS called LADS - Live Anomaly Detection System. An asset currently applied to the domain of microservices architectures and Kubernetes clusters. LADS is a soft real-time network anomaly detection system. The anomaly detection is based on a deep learning algorithm to model patterns of benign traffic and identify anomalous behaviour based on deviations from the normal behaviour. Importantly, LADS provides anomaly explainability based on what features of network traffic are root cause (critical) for the anomaly, and what anomalous values are observed with respect to the normal ones. LADS uses its own in-house network flow extractor for effective real-time gathering of traffic telemetry on a range of relevant protocols. Further information of LADS and its demonstrators can be found at the asset's booklet[1].

### 3.1.1 Key Features and Innovations

LADS main results achieved in the project:

- *Extended telemetry and detection capacity for the* HTTP protocol behaviour in response to selected attacks, such as user enumeration and dictionary attacks, on medical platforms in the first period of the project[2]. Refer to D5.1 [1].
- *Improved detection capabilities* (and telemetry) for Distributed DoS attacks of high concern for medical platforms – slow-read DDoS and low-rate DDoS. The detection of low-rate DDoS is based on modelling anomalies of packets burst per flow but also burst of new sessions per IP node or host. The slow-read attack detection covers anomalies overly long sessions towards a single or multiple IP nodes/hosts. Also, a hybrid slow-read low-rate attack detection capacity is offered. Refer to D5.2 [2].
- *Improved architecture workflow* between the sensors and the brain module. We decoupled the brain (AI module) from the telemetry sensor and allowed the brain to work with any number of sensors. In fact, the brain became agnostic to the actual number of sensors deployed in a given infrastructure. This decoupling enables better scalability (e.g., adding sensors without reconfiguring the brain), adaptability to different infrastructure sizes (e.g., small lab setups vs. large hospital networks), and deployment flexibility (e.g., distributed sensors across segments without impacting processing). We note that as a future work we plan to introduce control of sensors' availability. Refer to D5.2 [2].
- *Successful integration of RabbitMQ* message broker for efficient workflow messaging of telemetry data from the sensors to the brain module. The integration allowed to achieve the necessary level of abstraction of telemetry data from sensors generation for AI processing. It also facilitated parallel execution of models to get advantage of multi-CPU environments.
- *Successful automation of model training* on premise triggered by an operator, and *automated configuration and deployment* of the newly trained models for prediction.

[1] https://booklet.evidenresearch.eu/lads

[2] https://booklet.evidenresearch.eu/lads#:~:text=Demonstration%20videos



Funded by Horizon Europe Framework Programme of the European Union

- *Design of comprehensive GUI–Dashboard* – showing the value of the data generated by LADS. Several visual analytics are offered per protocol, per port, per flow, etc. To facilitate comprehension of anomalies detected.

The last three results of the message broker, automated on premise training, and the design of dashboard were addressed in the last reporting period, while the improved detection capability for the DDoS attacks and the decentralised architecture workflow were addressed in the second year of the project. We note that Rsyslog service was used initially before we migrated to the message brokering solution.

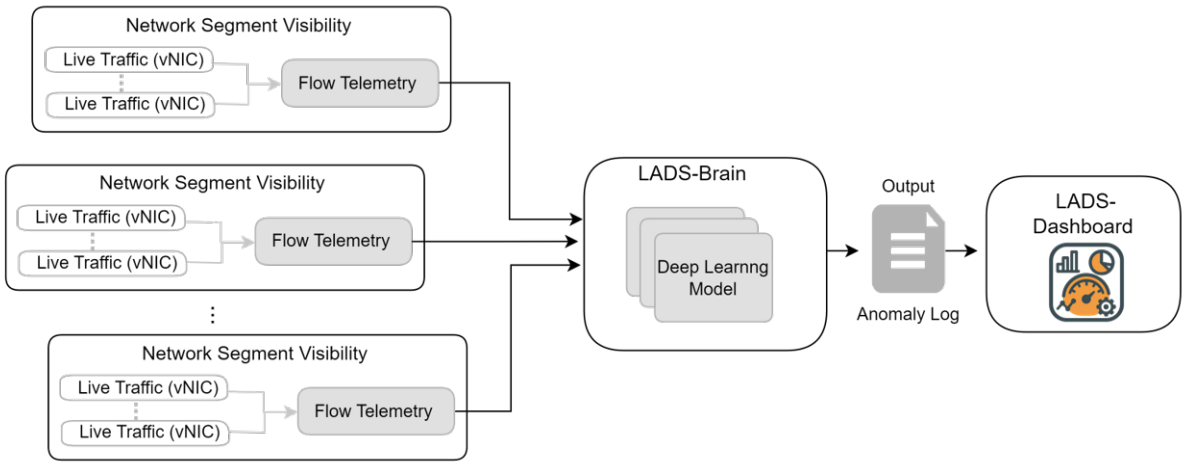


*Figure 5 LADS Workflow*

Figure 5 shows the final workflow of LADS realised in the project. It shows the distribution of telemetry sensors across segments of visibility of medical platforms, and association of a brain module to a set of sensors. The outcome of the LADS brain is the anomaly log file that contains info of all anomalies detected by the system. A Dashboard module, optional in the architecture, provides visual analytics on the anomalies detected. Anomalies contain telemetry data, and additional meta data indicating score of anomaly, critical features causing the anomaly useful for further decision making. The three modules Sensor, Brain and Dashboard have been dockerized. We note there is a fourth docker component, the RabbitMQ, that is required to handle telemetry messaging.

### 3.1.2 LADS Dashboard

The dashboard has been entirely developed in *Python*, using *Streamlit* open-source library as the web application framework to facilitate the rapid development of an interactive tool for real time user interaction and visualization [12].

For data processing and analysis, the dashboard is powered by *pandas*. This library is used for manipulating and analyzing data, providing data structures such as *DataFrame* and *Series*, along with typical data operations like reading, cleaning and aggregation.

In terms of data visualization, two main libraries are used: *Altair* and *Plotly*. *Altair* is a declarative statistical visualization library that enables the creation of interactive charts like line plots, bar charts and histograms [13]. *Plotly* is complementary to *Altair* and provides interactive visualizations with advanced charts such as pie charts and *Sankey* diagrams [14].

To enhance user experience, the library *Streamlit AgGrid* is integrated, allowing for interactive data tables that include features like sorting, filtering and selection [15]. Other dependencies include *numpy*, which is commonly used with *pandas*.

The architecture of the dashboard can then be categorized into several layers:



Funded by Horizon Europe Framework Programme of the European Union

- **Frontend**: *Streamlit* as the user interface, managing LADS inputs and displaying a variety of charts and tables.
- **Backend/Data Layer**: *Pandas* processes various data files, such as CSVs, preparing the data for visualization and analysis.
- **Visualization Layer**: *Altair* and *Plotly* are responsible for generating both interactive and static charts for exploratory data analysis and reporting

### 3.1.3 Pilot Integration Outcomes

Several iterations with the pilot environments have been performed to ensure the necessary support for successful integration of LADS. Importantly, we managed to connect the telemetry sensors with the relevant network interfaces of the environments of the two pilots (Pilot 1: Cybersecurity in Hospital and in Pilot 2: Cybersecurity for Telemedicine Platforms). Furthermore, we supported pilot owners in the training of LADS deep learning models and deploying those to test the system. We faced some issues in accessing normal traffic of the pilots, but after some interactions, the pilot owners managed to train the models. Another issue we faced was the integration of the Dashboard to visualise the results of anomalies. Current system setup requires the Dashboard to be present on the same environment where the Brain in deployed, so that log file of anomalies can be read and visualised. After few iterations with pilot owners, we managed to integrate the Dashboard in the pilot environments so that results can be visualised and easily understood.

## 3.2 CMD Log monitoring

Traditional log monitoring solutions are limited to rule-based (manual) analysis of time series data. In contrast, Log Monitoring System (LOMOS) makes use of state-of-the-art Natural Language Processing (NLP) architectures (e.g. LogBERT) to model log streams and capture their normal operating conditions. This monitoring system does not depend on any manually defined rules or human intervention but relies on that behavioural model to automatically detect deviations that would represent any kind of abnormal situations, including potential security threats (e.g. brute force attacks, DoS attacks).

### 3.2.1 Key Features and Innovations

**Key features**

LOMOS examines system or application logs, providing valuable insights into the current and past status of monitored assets. It uses self-supervised deep learning methods, such as Mask Language Modelling and Hypersphere Volume Minimization, mapping normal samples to their representations.

LOMOS identifies log templates that correspond to the log, searching for structure by identifying parameters (IDs, services, ports, etc.) and converting unstructured logs into structured log templates organized in a tree structure, including wildcards for parameters that differ from log to log. LOMOS observes the sequence of templates to understand normal behaviour, providing an anomaly score (including aggregation and specific counts) that should be low for normal logs.

**Technical innovations**

The LOMOS workflow is conceptually composed of four primary components (see Figure 6). The first component is the log parser, responsible for extracting events (or log templates) from



Funded by Horizon Europe Framework Programme of the European Union

unprocessed log messages. Following this, the anomaly detector trainer comes into play, in a self-supervised fashion, training the model using historical log messages. The third component, the anomaly detector, utilizes the previously trained model and new data that has been pre-processed by the log parser to calculate anomaly scores on a per-log basis. These anomaly scores, along with the timestamp and anomalous log line, are then presented on the dashboard (Grafana). It's worth noting that a single LOMOS deployment has the capability to handle multiple log sources using distinct models.

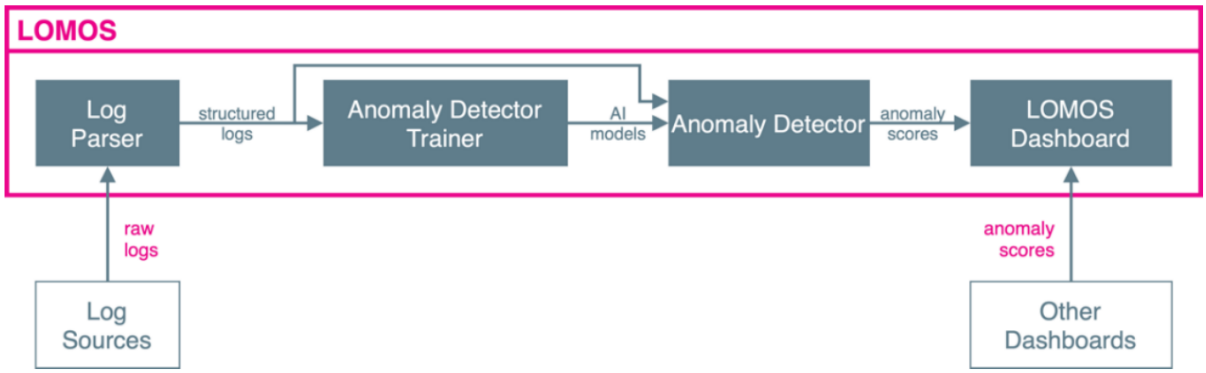


*Figure 6 Workflow of the LOMOS log anomaly detection system based on self-supervised learning*

The system utilizes the Drain method [16] to process raw logs through the Log Parser component. In that, the events are often called log templates.

The anomaly detector trainer takes log templates as input. Currently, the LogBERT [17] model is implemented and extended. LogBERT is an appropriate method because it is a self-supervised machine learning method for window log-based anomaly detection. Therefore, it does not require labelled data for training. The log templates, more specifically, their IDs, are grouped into time or index-based windows. Sequences of IDs are input into the BERT-like [18] transformer model. The trained model has a fixed vocabulary, so new log templates cannot be added without retraining the model. Because the model is trained in a self-supervised manner on normal data, it does not require per-log labelling for training. The model is expected to be robust enough so that even a small number of anomalies in the training data should not significantly impact its performance. However, the more the data reflects the normal bahaviour, the better the model will be. Graphics Processing Unit (GPUs) are needed to train a model like LogBERT.

The anomaly detector utilizes a model acquired from the anomaly detector trainer. Fresh raw data is initially processed with the trained log parser and then forwarded to the anomaly detector, which retrieves the trained model from the model registry. Anomaly detection can be performed either on demand or scheduled periodically. We have expanded on the concept from the LogBERT method to assign anomaly scores on a per-log basis. For inference, Central Processing Unit (CPUs) are suitable for most use cases. However, GPUs are also supported for inference when there is a need for very high throughput.

The LOMOS dashboards offer interactive visualizations of the extracted log templates and logs with anomaly scores. The dashboard guides administrators through the process of training a model and running inference. Initially, the log parser must be trained, which is accomplished by selecting a period from historical data. Administrators can explore extracted log templates and fine-tune hyperparameters as needed. In the next step, the administrator trains the model, which is the most hardware resource-intensive step. Afterward, the model is ready to be deployed. The administrator can then execute one-time or periodic inference, with the latter being based on a given interval. The administrator can monitor logs and anomaly scores in the dashboard and set rules to receive alerts in case of high anomaly scores.



Funded by Horizon Europe Framework Programme of the European Union

The workflow depicted in Figure 7 is distributed across various components (see LOMOS architecture in Figure 7). Log data storage serves as a log source in the workflow diagram. The log parser is housed within the LOMOS parsing component. LOMOS log anomaly detection is utilized for both the anomaly detection trainer and the anomaly detector. The tasks are carried out by Celery CPU and GPU workers. The outcomes are then showcased in the Dashboard. All the components are dockerized, making them easy to deploy.

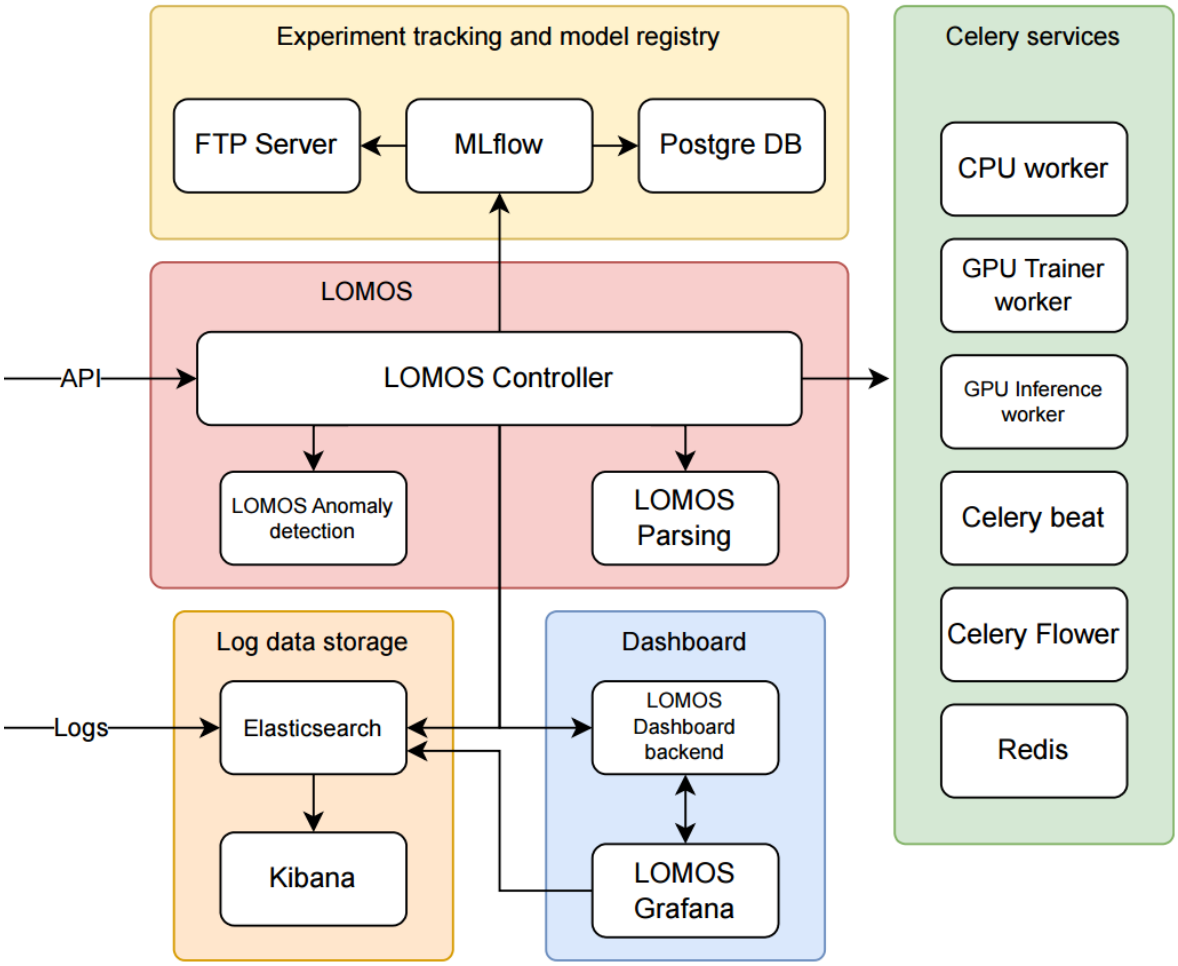


*Figure 7 LOMOS Architecture*

### Licence and download

LOMOS has a proprietary license owned by XLAB released on a case-by-case basis beyond the pilot usages and trials.

The component artifacts (e.g. code, documentation, images...) can be downloaded or pulled from the project's development Github repository: https://github.com/Cylcomed/LOMOS.

LOMOS CLI for starting LOMOS parsing, training and inference tasks can be downloaded from here: https://github.com/Cylcomed/lomos-cli

The repositories can be opened on invitation. To request access to this repository, please contact: Mr. Simone Favrin (s.favrin@mediaclinics.it).



Funded by Horizon Europe Framework Programme of the European Union

### 3.2.2 Pilot Integration Outcomes

LOMOS as a part of the CYLCOMED Toolbox was used in Pilot 1: Cybersecurity in Hospital Equipment for COVID-19 ICU patients (testbench platform) and in Pilot 2: Cybersecurity for Telemedicine Platforms (Mediaclinics Health Platform (MD MHP)). The aim of using LOMOS in both pilots was to monitor and analyse application and system logs to detect an anomaly in which could be a threat or an attack.

In Pilot 1, LOMOS was deployed alongside LADS and Dashboard on the EVIDEN server. LOMOS communicates with the RGB server in the RGB testbench where IAM components are deployed, digesting the logs from the IAM components (Lus4MED and FE4MED), running training and inference and displaying anomaly score on the Dashboard. The decision to deploy LOMOS on the EVIDEN server was influenced by the fact that LOMOS needs GPU power to train the model which RGB server can't provide.

In Pilot 2, LOMOS was configured and deployed on the dedicated machine with GPU (server) that will serve as the testing platform for the Toolbox in the OPBG and FHUNJ hospitals. In this scenario, LOMOS consumes the Mediaclinics Health Platform web proxy logs, trains the model using these logs, runs the inference and displays the anomaly score on the Dashboard.

This scenario was also primarily used to finetune the LOMOS anomaly detection engine using two attack simulations: brute force entry to the platform (multiple HTTP request to a single URL in a short period of time) and low-rate DoS (in the form of multiple HTTP requests that keep the sessions and ports opened longer than expected and thus drain the resources).

Additional adjustments were made to LOMOS deployment configuration:

- CUDA bindings were added to the Pytorch version to enable compatibility with Nvidia GPUs on the host server, alleviating the need to install CUDA drivers.
- LOMOS containers now communicate in the same Docker network (with local IP addresses), resolving the network issue of needing to call the host IP address.

## 3.3 Lessons Learned

### 3.3.1 AI-Behavioural Analysis

Lessons learned for LADS are twofold:

- *Visibility of network traffic* – how to identify relevant virtual network interfaces for the assets subject of protection. It is not a straightforward solution in some cases, especially in more complex docker environments, and some level of support and automation is needed. We note that LADS can monitor multiple network interfaces per single environment, and a default option is to listen on all of them. However, it is not always a desirable solution as it can see traffic for other asserts not in scope.

- *Training duration* – how to identify the duration of training for a pilot environment. It is also not a straightforward solution given different environments may have specific behaviour occurring only during some specific hours per day or even per week. Important for the LADS training is to cover all time periods where representative traffic occurs to make sure it learns (sees) all legitimate traffic. Not only that, but it also needs minimum necessary volume of such traffic behaviour in order to train well the models.

Both lessons are important to be addressed in future releases of the LADS, and have already formed part of the roadmap of the asset.



Funded by Horizon Europe Framework Programme of the European Union

### 3.3.2 CMD Log Monitoring

Lessons learned for LOMOS are related to the configuration and deployment as well as to the training and inferencing.

Medical environments where LOMOS is deployed are heterogeneous and with different limitations regarding hardware available (GPU present or not), GPU acceleration, virtualisation and hostOS technologies used as well as accessing LOMOS in hospital pilots. Thus, it was necessary to reconfigure LOMOS engine and adapt deployment scripts to have LOMOS successfully deployed and running in these environments.

Regarding model training, LOMOS uses the LogBERT methodology. Testing of this methodology in several other projects beside CYLCOMED (SUNRISE (HE-101073821) and ICOS ((HE-101070177))) has shown that the (self-supervised) training process is sensitive to the proper structure and content of logs. From this it follows that the logs must be human-readable and include precise timestamp, i.e., must not be pre-processed in any way. Consequently, not all logs could be used directly, and thus only appropriate system logs (Nginx webproxy for Pilot 2) and application logs (Lus4MED and Fed4MED) were used.

Regarding content, logs should come from a system with mixed normal behaviour and anomalies (cybersecurity incidents) as by using the self-supervised training process, LOMOS's model will be able to properly distinguish between normal behaviour and cybersecurity incidents. However, if the anomalies are too frequent (e.g. logs capture only the time period of a cyberattack), the model trained will consider them as normal behaviour and will not be able to detect similar activities during inferencing. Thus, a proper combination of representative system or application activities is very important for model training and inferencing.



Funded by Horizon Europe Framework Programme of the European Union

# 4 Device Integrity, Security and Service Management Layer

## 4.1 Device Integrity Check

As is well known, certification of a medical device is a complicated, expensive and time-consuming process.

For this purpose, the strategy has been to evaluate the technological state of art. We have also carefully addressed the necessary standards for regulatory adherence, focusing primarily on interoperability of medical devices and the applicable tools for safeguarding highly confidential medical information. Central to this effort is the implementation of robust encryption methods, which serve as the cornerstone for data protection. In this scenario, the deployment has involved integrating Cylcomed's security tools into a multi-parameter monitoring device manufactured by RGB Medical Devices. This device is an NMT Infusion controller and is tailored for automated neuromuscular relaxation regulation, where it continuously assesses relaxation levels, computes the appropriate drug dosage to maintain target levels, and autonomously manages an infusion pump to deliver the medication. In practical healthcare environments, the exchange of sensitive data often involves transmission to external cloud-based systems such as Hospital Information Systems (HIS). Consequently, the device operates within a highly interconnected ecosystem, making the security tools developed through the Cylcomed project especially pertinent. These tools are designed with versatility in mind, capable of adapting to other medical devices beyond the current application. To ensure successful integration, it has been vital to understand the procedural steps and technological advancements required.In summary, integrating Cylcomed's security solutions into legacy medical devices entails deploying external intermediary hardware, developing device-specific applications for data encryption, and adapting server-side systems for secure data management, all while ensuring compliance with regulatory standards and maintaining interoperability across diverse healthcare environments.

### 4.1.1 Key Features and Innovations

The innovation has been to find a strategy that minimizes the certification requirements for legacy devices. Incorporating new functionalities in medical devices such as in the Use Case 1 of Cylcomed is an inherently complex task, with significant considerations around safety and security protocols and regulatory compliance. Many devices lack the necessary computational resources or do not run on operating systems compatible with the Cylcomed tools. To address this challenge, the chosen strategy has involved deploying an external intermediary device that interfaces directly with the monitor, hosting all security functionalities, and acting as a bridge between the device and external networks. For this purpose, a Raspberry Pi 4 mini-computer was selected, offering sufficient processing capacity to run the Cylcomed applications and providing essential communication ports.

We have incorporated the security features that result from CYLCOMED in a standard external Linux-based board, with enough CPU power so that performance of essential functionalities is not compromised, connected via a serial channel to the medical monitoring device. The medical monitoring functionalities have been preserved from cybersecurity hazards and constitute the legacy device. The external module interoperates with the user interface SW component of the medical monitoring device, e.g. to report the status of the SW integrity. Besides, the infusion pump controller functionality is also embedded in the external board. It communicates with other medical devices such as the infusion pump tree, and the HIS (Hospital Information System) via ethernet, to report on the vital signs' evolution of the patient. These gateway functionalities are performed following the security recommendations



Funded by Horizon Europe Framework Programme of the European Union

mandated by regulatory norms. This strategy has been followed using a legacy device (Vision Air Equipment), and the focus has been to achieve an implementation that can be used as a module for legacy devices in the Intensive Care scenario. Of course, the procedure to upgrade the functionality of another device requires an integration toolkit with a protocol that includes the cybersecurity alarms generated. As manufacturer of the legacy device, RGB has adapted the external board with an open protocol generated in CYLCOMED. A User´s Guide for the implementation of the External Board connection to the Vision Air Multiparameter monitor has been written, including the communication protocol and the specific test actions to verify the proper integration and a guide to perform the implemented functionalities.

A critical component of this integration has been the FE4MED application, developed by Eviden, which manages data encryption. This application guarantees that data transmitted from the device remains encrypted throughout its journey across networks, with decryption restricted to authorized entities. Each device-specific implementation involves developing a tailored application for the Raspberry Pi, responsible for capturing monitor data, preparing it for encryption, and managing policy requests for data access. The project allows flexibility regarding the development of technologies and programming languages, provided that applications are containerized as Docker images and interact appropriately with the Cylcomed toolbox.

XLAB provides LOMOS, a tool that analyzes logs with artificial intelligence techniques to draw conclusions from the logs. To use these tools, a specific application is developed for each device and deployed on the Raspberry Pi board. The Cylcomed project does not impose restrictions on the development of these applications, neither in technologies nor programming languages, beyond being deployable as Docker images and those applied by the interaction with the toolbox applications.
Beyond device-side considerations, server infrastructure must also be configured to support secure data handling. Two main scenarios are envisaged:

1. Utilising a server environment that already hosts Cylcomed tools: Here, the application running on Raspberry Pi must format data and generate or request decryption policies compatible with the server's specifications. This approach enables multiple types of monitors to send data to a unified platform.
2. Integrating Cylcomed functionalities into existing server applications: This involves embedding the security and decryption tools developed by Eviden and Martel into the server-side software, enabling encrypted data storage and user authentication. Modifications include updating user registration systems and adjusting database operations to ensure data remains encrypted at rest, accessible only to authorized users.

### 4.1.2 Pilot Integration Outcomes

The infusion controller functionality as a medical device has been fully achieved. We have reached a TRL level of 6, and the reason why it is not higher is because we need a bit more robustness before performing clinical trials in a real environment.

From the point of view of performance, we have done the performance testing in a successful way, doing the work at laboratory level as was initially programmed. For this purpose, we have developed a Digital-twin solution operating as a test bench able to mimic the behavior of the patient to relaxant drug infusion. For this purpose, we have used pharmacokinetic and pharmacodynamic (PK/PD) patient modelling.



Funded by Horizon Europe Framework Programme of the European Union

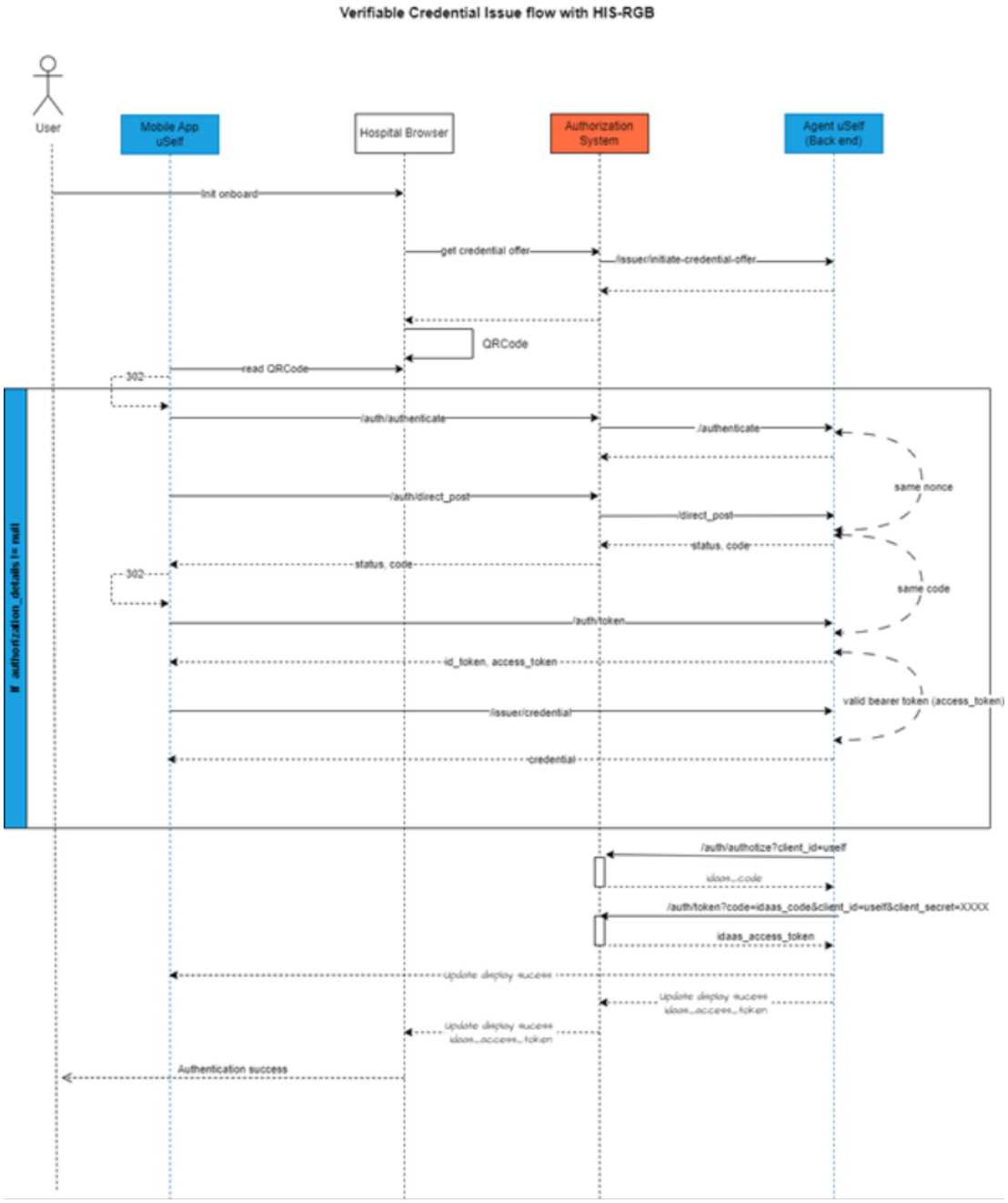


*Figure 8 User's onboarding process flow*

Given the diversity of medical monitors and the variety of data transmission protocols available, direct integration of the Cylcomed toolbox without customisation is impractical. However, the medical devices interoperability will be extremely simplified in the future, thanks to the final deployment (after nearly 50 years of development) of the standard IEEE 11073. This is commercially called SDC protocol (stands for Service Oriented Device Communication). This means that customisation will require much less effort in the future.

To learn that ISO IEEE 11073 will be finally deployed as the SDC protocol (universal language) is a great achievement that will open a new vision of medical devices and greatly enhance current performances. For legacy devices that want to be interoperable with SDC, there are multiple "paths" that a device manufacturer can take, depending on what the current device hardware can support, how regulations affect the requirements and how the hospitals using their legacy device are operating



Funded by Horizon Europe Framework Programme of the European Union

## 4.2 CMD Security Maintenance

### 4.2.1 Key Features and Innovations

The security functionalities of Cylcomed are delivered via Docker containerization, ensuring standardised deployment and simplified inter-component communication. All essential modules are packaged into Docker images, facilitating seamless system operation. For management, configuration, and updates, the project adopts the Mender platform, following the protocols established by Martel within this initiative.

Given the importance of keeping connected devices constantly up to date with the latest security patches, to minimise potential attack avenues and reduce the risk of security exploits, we deploy Mender as the main tool to provide **Over the Air** (OTA) updates.

3 components are part of this setup:

- **The fleet of connected medical devices**: these are the devices Mender will establish communication with, and where the updates will be deployed. Each device will have a Mender “client” installed to permit communication with the Mender server.
- **The Mender Server:** The main service, or more accurately a combination of services, that deals with communication with devices and delivering the updates (packaged in the form of artefacts) to them. This also includes authentication for users and devices, and storage for the updates to be delivered.

**The Workstation:** This component handles checking the device updates and packaging them into the artefacts that will then be provided to the Mender Server and delivered to the connected devices..

### 4.2.2 Pilot Integration Outcomes

Mender meanwhile was deployed and used to successfully deploy packaged artifacts towards connected devices, both simulated in the form of a virtual machine environment and physical in the form of Raspberry PI based devices in Pilot 1 alongside RGB.

OTA management via Mender was adopted for secure device and service updates. It has rollback capabilities in controlled environments. Critically, the architecture shifted from cloud-first to hybrid for Pilot 1 addressed regulatory and ethical constraints per D6.1 [1] and initial feedback in D6.2 [2], ensuring compliance with GDPR/MDR while maintaining scalabilityAs stated before, the RGB External Board for Device Integrity Check is included in the Device Integrity, Security & Service Management layer. This Linux module connects to legacy devices like Vision Air without changing their firmware. It sends Integrity Reports using its API. CMD Security Maintenance uses OTA and IaC for updates and deployment with Mender. It exposes endpoints through Mender Management APIs in test setups. Lessons Learned In perspective, the approach taken in Cylcomed to incorporate new cybersecurity tools in legacy devices seems to have been the correct one.

## 4.3 Lessons Learned

The approach has been to have a separate piece of small connector hardware (a small micro-pc the size of a raspberry-pi), directly attached to the legacy device. This solution is valid for all types (brands) of medical devices: The connector then talks to the legacy device using its regular interface, which usually is a serial port, and on the other end outputs encrypted/secured messages over an ethernet/Wi-Fi port. The main advantage here is that this is a minimum



Funded by Horizon Europe Framework Programme of the European Union

effort retrofit for most hospitals, as no updates or external connection of the device is required, and no native ethernet interface on the legacy device is required. However, the main advantage is that, according to the MDR, a component that merely translates signals from one format to another is not categorized as “Medical Device” and therefore its development does not fall under ISO 13485.

A variation of the previous approach: Instead of having one connector device per legacy device, it is also possible to provide one “server scale” connector device that does the securing for multiple devices at once. This is especially relevant for smaller devices or cases like Use Case 1 of Cylcomed, where multiple devices are working together in the same rack (i.e. infusion pump racks).

In the future, a more robust, but also most effort and costly will be to roll out a full update of the native firmware of the device to incorporate a cybersecurity library that natively talks over a network interface. The requirements would of course be to have enough powerful hardware on the device, the ability to roll out updates, and network access to the device. Also, as this affects the software of a medical device, such an update would be required under MDR development & life cycle management according to ISO 13485 under the applying safety class of the device.

A second most robust approach would be similar with the same downsides apart from maybe a faster development time: The device would have a separate piece of software running and doing the security role to the native proprietary signals used. Depending on the device, this could fall into a different safety class which might be relevant.

One of the most time-consuming tasks regarding customization of solutions that require several medical devices to obtain a new functionality is the fact that until now, every device would “speak its own language”. However, there is some important news in relation to the deployment of the IEEE11073 standard.

This standard started in the 80s, but big companies like Dräger, GE and Philips have started implementing it under the acronym SDC. The significance of this is huge, and this is an important message for cybersecurity tools developers considering the exploitation strategy.

However, the toolbox of Cylcomed still makes a lot of sense, since SW cybersecurity tools could /should be implemented in an external board in a future SDC implementation (outside of Cylcomed scope) as a short-term solution to avoid complete re-certification of legacy devices. Then it is a matter of seeing whether they fit within the expectations of the cybersecurity tools to be implemented in SDC.

We have reached as an outcome of Cylcomed the conclusion that after a thorough look at applicable standards, we have finally found the one that is most relevant of all: IEEE 11073. And we come at the right time to exploit our results (outside of Cylcomed time frame).



Funded by Horizon Europe Framework Programme of the European Union

# 5 Identity, Access Management, and Data Protection Layer

The final architecture of Identity and Access Management (IAM) and Data Protection layer is depicted in Figure 9. This picture shows how the IAM and Data Protection tools (LuS4MED and C-OPA, and FE4MED, respectively) interact each other, for protecting the privacy of the patient's data (from connected medical devices and smart phones) and the access control. Also, shows the relation with other CYLCOMED toolbox tools, such as AI-Log Monitoring (LOMOS). In a nutshell, the health data collected by the CMDs are protected by encrypting the patient's health data with the FE4MED (the Attribute-Based Encryption data protection solution). These protected data are stored in the hospital system. The medical staff proceed to authenticate by presenting Verifiable Credentials (VC), containing user role and attributes, stored in the LuS4MED mobile app wallet. The role and attributes included in the VC are provided to the authorisation solution (C-OPA in pilot 1 and legacy Keycloak in pilot 2) which grants the users to access the allowed services. Based on the user role and attributes the FE4MED decrypts those requested data the user is allowed to see. The use of the data protection layer is monitored by the AI-Log monitoring tool (the LOMOS solution) for detecting anomalies during the encryption/decryption processes [2].

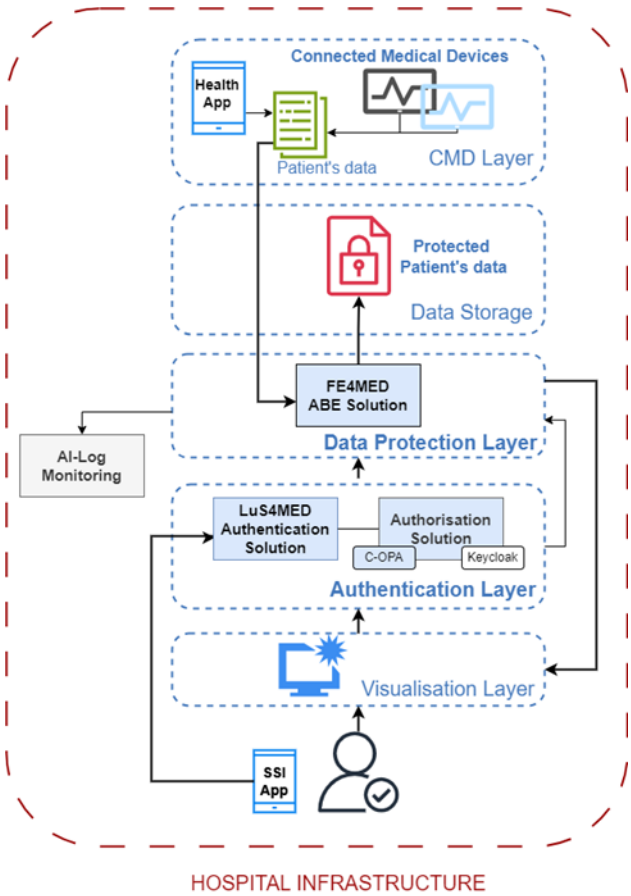


*Figure 9 Interaction between the IAM & Data Protection tools for protecting data from CMDs*



Funded by Horizon Europe Framework Programme of the European Union

## 5.1 SSI solution

The SSI solution named LuS4MED (uSelf for Medical Data) comprises two elements, the uSelf Agent for issuing and verifying the VC, and the uSelf mobile app for storing and present the VCs.

### 5.1.1 Key Features and Innovations

During the period of development of Task 5.3 “Identity & Access Management and Data Protection for Connected Medical Devices” the LuS4MED tool has been implemented and integrated with the Hospital system in order to give access to the data stored in the hospital system. This access is allowed by using a VC tailored to the CYLCOMED scenarios. Figure 10 depicts how the two components of the LuS4MED act:

- The LuS4MED App stores in the Wallet the VC issued by the uSelf Agent.
- The uSelf Agent issues and validates the VC in the hospital system domain

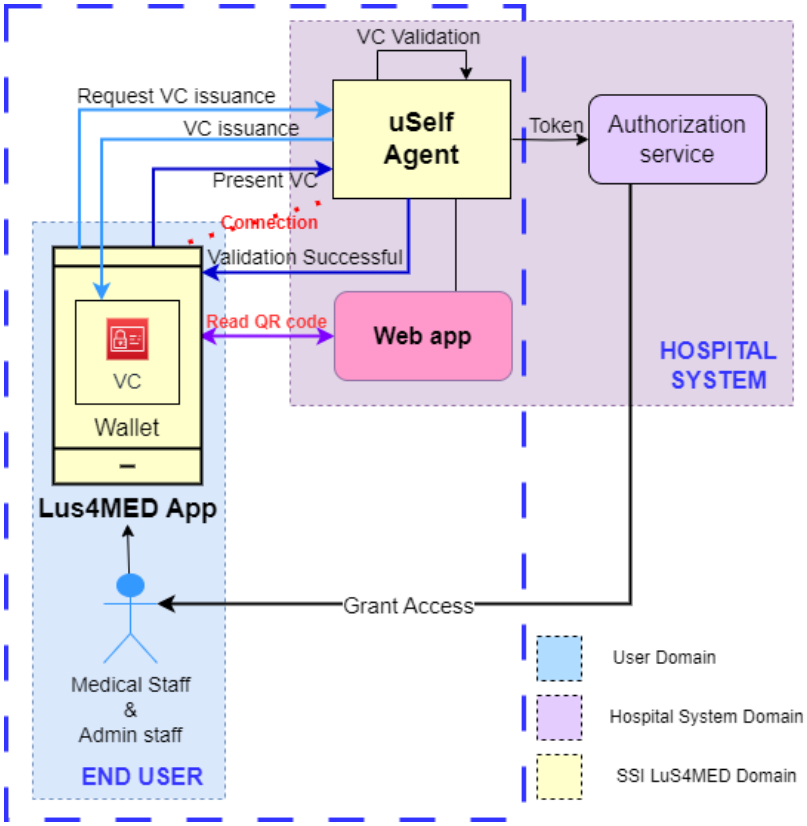


*Figure 10 Access to Hospital System using VC*

Namely, the uSelf Agent plays the role of VC issuer in behalf the hospital, and also validates the VC presented by the user when the medical staff authenticates against the hospital system. The VC issued by the Agent contains the role and users’ attributes in two-fold, the authorisation solution uses the role for granting user access to the allowed services, and the FE4MED decrypts the patient’s health data based on the attributes provided. In this manner both, the authentication solution (LuS4MED) and the protection solution (FE4MED) are integrated for **protecting the access to the user’s data** and **preserve the privacy of these sensitive data**.

The Table 1 CYLCOMED project repositories for LuS4MED tool includes the CYLCOMED project repositories for the LuS4MED components:



Funded by Horizon Europe Framework Programme of the European Union

*Table 1 CYLCOMED project repositories for LuS4MED tool*

| Component | Link[3] | Content |
|---|---|---|
| **LuS4MED** | Cylcomed/LuS4MED-SSI | LuS4MED description, deployment and examples of use |
| **uSelf Agent** | LuS4MED-SSI/uSelf-Agent at main · Cylcomed/LuS4MED-SSI | uSelf description and docker image |
| **uSelf Mobile App** | LuS4MED-SSI/uSelf-App at main · Cylcomed/LuS4MED-SSI | uSelf mobile app description and aar artifact. |

### 5.1.2 Pilot Integration Outcomes

Considering the LuS4MED tool, this solution **simplifies the complexity of integration of the SSI technology** into the hospital systems and **facilitates the management of the VCs**.

The LuS4MED has been deployed into both pilots' infrastructure and integrated with the authorisation solution in a smooth way. In the case of Pilot 1 is integrated with the C-OPA authorisation tool (described in section 5.2) implemented for CYLCOMED project. In the case of Pilot 2 is integrated with the Keyclock legacy system managed by MCI.

## 5.2 C-OPA Authorization solution

The purpose of the C-OPA component is to protect services from unauthorized access. It allows the definition of security access policies for each component it protects and authorizes or denies access based on policy compliance.

### 5.2.1 Key Features and Innovations

C-OPA is made of 3 main components:

- **The Envoy Proxy**: sitting in between the client and the protected service, intercepting a request and forwarding to the Policy Agent (OPA).
- **The Open Policy Agent (OPA)**: processing the security policies alongside the request, permitting or declining access based on what is written in the policies (including checking and validating security tokens).
- **The Policy Store**: the storage component where all security policies are bundles and stored.

The way services are protected thanks to C-OPA can be summarized as follows: the envoy proxy sits in between an entity and a protected service, and intercepts requests made by the entity towards the protected service. The request is then forwarded to OPA, which evaluates

[3] The repositories can be opened on invitation. To request access to this repository, please contact: Mr. Simone Favrin (s.favrin@mediaclinics.it).



Funded by Horizon Europe Framework Programme of the European Union

the request using the policies from the Policy Store. Compliant requests and let through, non-compliant requests are rejected.

This setup comes with some key advantages over implementing this kind of authorization per service:

- The C-OPA proxy can be configured to freely protect any component, including more granular configuration for endpoints, routes, HTTP methods, ...
- The language the policies are written in (Rego) offers a high degree of flexibility, allowing fine evaluation of requests, including the option to alter them on the fly.
- By storing the policies in a dedicated store, they can be accessed and edited with ease, and OPA can be configured to pull them when changes are detected, ensuring that changes in policies are continuously reflected in the component.

### 5.2.2 Pilot Integration Outcomes

C-OPA has been successfully deployed in Pilot 1, protecting the FE4MED component.

In this setup, C-OPA sits in front of the FE4MED component, transparently intercepting requests made to it. Upon receiving a request, C-OPA will verify that the request is compliant with the security policy we have defined for the FE4MED component (Figure 11). In this case, the policy checks for two things:

- Does the request come with a valid (JWT) security token as provided by Lus4Med? Valid meaning a non-expired token with a valid signature.
- If a request is made towards a certain endpoint, was the token issued to a user with the correct role?

```
# Allow admins to access the policy endpoint
allow if {
    valid_jwt
    path == "/policy"
    role == "admin"
}

# Allow doctor1 to access the decryptData endpoint
allow if {
    valid_jwt
    path == "/decryptData"
    role == "doctor1"
}

# Allow doctor2 to access the decryptData endpoint
allow if {
    valid_jwt
    path == "/decryptData"
    role == "doctor2"
}
```

*Figure 11 Excerpt from the security policy where roles are checked for the FE4MED endpoints*

This policy enforces that the **/policy** endpoint can only be accessed by an admin, and the **/decryptData** endpoint can only be accessed by doctor1 or doctor2. If these conditions are not met, the request is denied with an authorization error.



Funded by Horizon Europe Framework Programme of the European Union

## 5.3 ABE Solution

The FE4MED tool has followed a modular design and comprises 3 main modules: i) an encryptor module (provided in two flavours, as mobile aar library for being integrated into a mobile phone, and a docker image for being deployed on a RBPI); ii) a FE4MED server including the decryption module and the key generator; iii) a storing module for policies and a key secure storage.

### 5.3.1 Key Features and Innovations

Regarding the **FE4MED** tool, the **modular design** of this Attribute-Based Encryption (ABE) solution facilitates the use of this data protection tool in different scenarios. Figure 12 FE4MED tool architecture shows the different FE4MED layers.

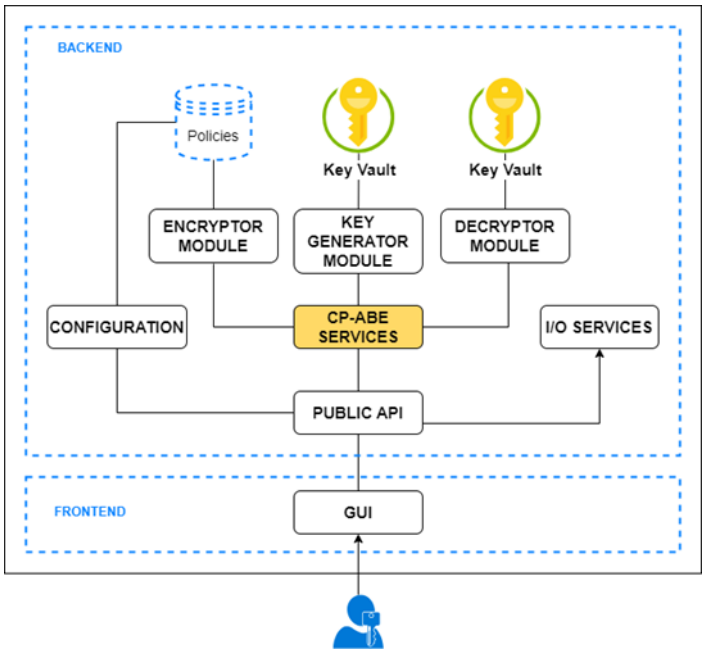


*Figure 12 FE4MED tool architecture [1] [6]*

- CP-ABE services layer, in charge of encrypting and decrypting data and the key authority for generating keys.
- Storage layer, which includes secure elements for storing keys and databases for saving the flexible attribute-based policies.
- Public API for accessing the configuration and I/O services, and the ABE services.
- A GUI for accessing the public API, if needed by admin staff or end users.

The encryptor module has been developed in two flavours: i) Encryptor module for being deployed and integrated in a hospital server infrastructure or in a more constrained environment such as a Raspberry PI; ii) Encryptor module for Android application for being integrated in a smartphone application. With this approach, FE4MED covers the needs of both pilots. On top of that, it provides a **fine-grained access control to data**.

The possibility of using different modules and flavours facilitates the integration of this tool in several scenarios and environments, even in constrained devices. Additionally, the deployment as Docker images facilitates the scalability

Table 2 includes the CYLCOMED project repositories for the FE4MED components:



Funded by Horizon Europe Framework Programme of the European Union

*Table 2 CYLCOMED project repositories for FE4MED tool*

| Component | Link[4] | Content |
|---|---|---|
| **FE4MED** | Cylcomed/FE4MED-ABE | FE4MED overall description of the ABE data protection solution |
| **FE4MED main module** | FE4MED-ABE/abe-go-toolbox at main · Cylcomed/FE4MED-ABE | FE4MED description and docker image |
| **FE4MED mobile encryptor** | FE4MED-ABE/abe-mobile-encryptor at main · Cylcomed/FE4MED-ABE | Encryptor module for being integrated into an Android smart phone |
| **FE4MED server** | FE4MED-ABE/fe4med-server at main · Cylcomed/FE4MED-ABE | Contains the configuration module and the decryption functionalities for being deployed on the hospital infrastructure side |

### 5.3.2 Pilot Integration Outcomes

FE4MED tool has been integrated in both pilots in different way:

- **Pilot 1:** As the CMDs (providing the patient's health data) are connected to a Raspberry PI the encryptor module is deployed in this element.
- **Pilot 2:** The use of smart phones for retrieve the patient's data implies the use of the FE4MED mobile encryptor. The first iteration of the integration will use a remote endpoint for this step to ease the development of the MVP.

In both pilots the FE4MED server is deployed as a docker image, facilitating the scalability of the process.

## 5.4 Lessons Learned

From the technical perspective, the best solution to address the integration of LuS4MED and FE4MED with other components (e.g., C-OPA), monitoring systems (RBPI) and systems (e.g., hospital backend) was to follow a modular design of both components in order to facilitate a smooth integration and improve scalability. With the aim to facilitate the encryption process in the mobile devices (pilot 2), an Android library of the encryptor module has been implemented.

A key point for smoothing the integration process into the hospital system was to maintain a fluent and permanent relationship with the members of the other teams and pilots.

Also, it is important to mention that the integration process has been scheduled in two iterations to ease the piloting.

[4] The repositories can be opened on invitation. To request access to this repository, please contact: Mr. Simone Favrin (s.favrin@mediaclinics.it).



Funded by Horizon Europe Framework Programme of the European Union

# 6 Cybersecurity Dashboard

The CYLCOMED Security Dashboard is the common window into the toolbox: it collects signals from the AI detectors, converts them into clear timelines and views, and helps hospital staff understand what is happening across connected medical devices without digging through raw logs. The final version keeps the same simple idea as in D5.2. [2] (one place to see network and system anomalies), but focuses on on-premise operation suitable for hospitals. At its core, the dashboard ingests events from the two analytical engines—LADS for network behaviour and LOMOS for log behaviour—and renders them as real-time streams and historical trends. Logic groups related detections so users can see patterns rather than isolated alerts (for example, a suspicious network spike from LADS paired with unusual service restarts from LOMOS).

The dashboard has been packaged for repeatable deployment alongside LADS and LOMOS. Hospitals can provision the stack with standard automation so development, testing, and production instances remain consistent. This keeps maintenance simple and ensures updates to the detectors flow cleanly into the monitoring views without manual work.

## 6.1 Key Features and Innovations

The dashboard provides a comprehensive overview of network traffic data and anomaly detection results. Each section serves a specific purpose to help inspect, visualize, and interpret data. Charts and tables support operational monitoring and investigation.

The time series dot-line chart shows network flow predictions grouped in 30-second intervals. It uses date-time data and counts flows as "Normal" or "Anomaly". This helps spot when anomalies happen and compare with normal traffic. The pie chart shows the distribution of flows by RMSE value, counting "Anomaly" and "Normal" based on the threshold. It gives a quick view of anomalous vs. normal traffic percentages. As shown in Figure 13, the LADS dashboard time series and pie chart provide these visualisations side by side.

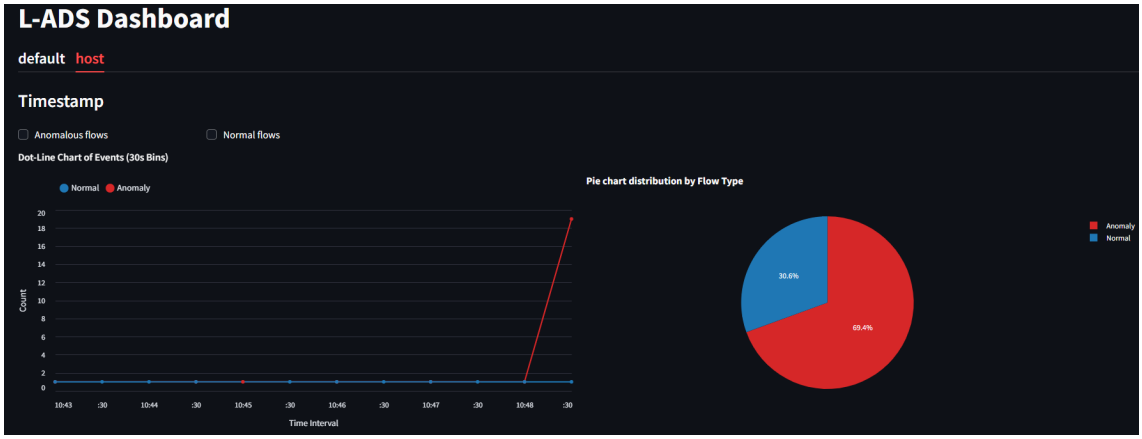


*Figure 13 Dashboard time series and pie chart*

The interactive table lists details of anomalous flows. Users can select flows to inspect. It includes network features like addresses, ports, and protocols. As shown in Figure 14, the LADS dashboard interactive table displays this data.



Funded by Horizon Europe Framework Programme of the European Union

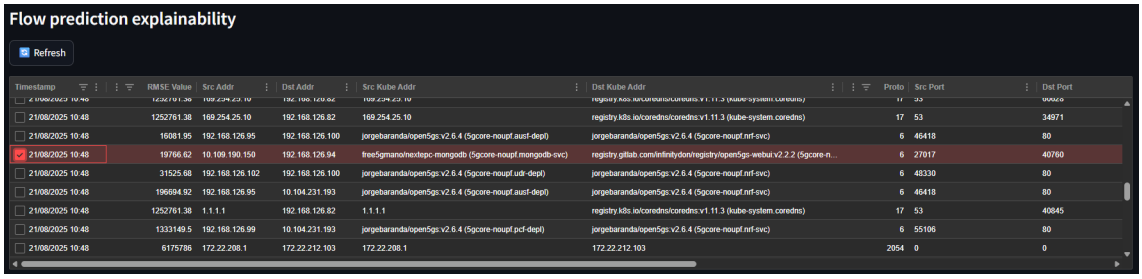


*Figure 14 Dashboard interactive table*

When a flow is selected, bar charts show critical features for predictions and non-conformity features. Critical columns give percentage impact on classification. Non-conformity shows how values exceed thresholds. This explains why a flow is anomalous. As shown in Figure 15, the LADS dashboard flow explainability uses these bar charts.

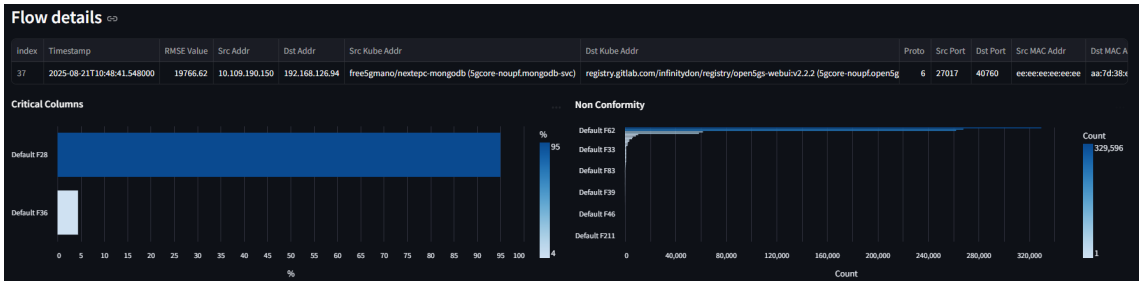


*Figure 15 Dashboard flow explainability*

To further improve understanding of network behavior, the dashboard also provides stacked bar charts that show the distribution of source and destination addresses, kube addresses, MAC addresses, protocols and instance IDs. The data type is the count of flows for each value and flow type ("Normal" or "Anomaly"). In the figures below is shown how the charts look like and how they help users identify hosts addresses, nodes or protocols most involved in each type of flow.

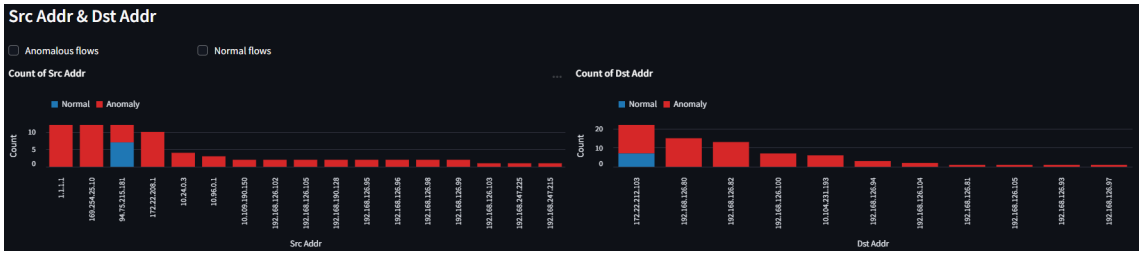


*Figure 16 Dashboard ip addresses bar charts*

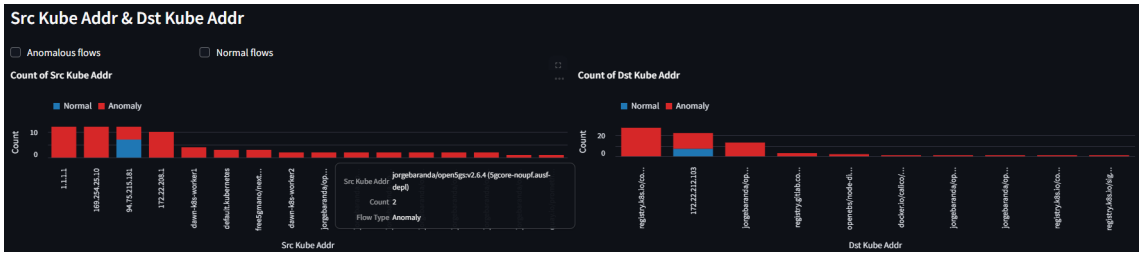


*Figure 17 Dashboard kube addresses bar charts*



Funded by Horizon Europe Framework Programme of the European Union

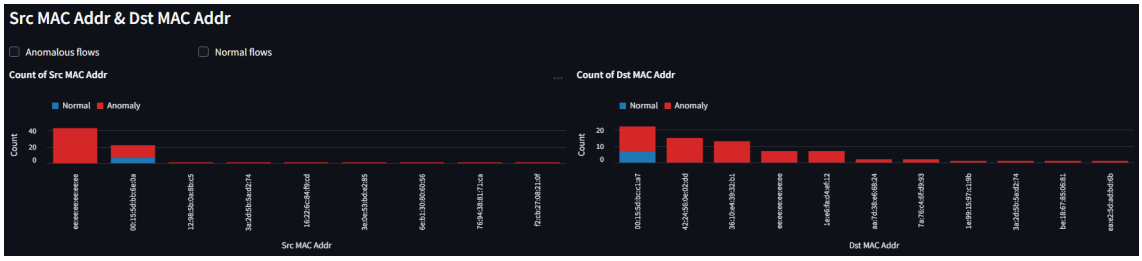


*Figure 18 Dashboard MAC addresses bar charts*

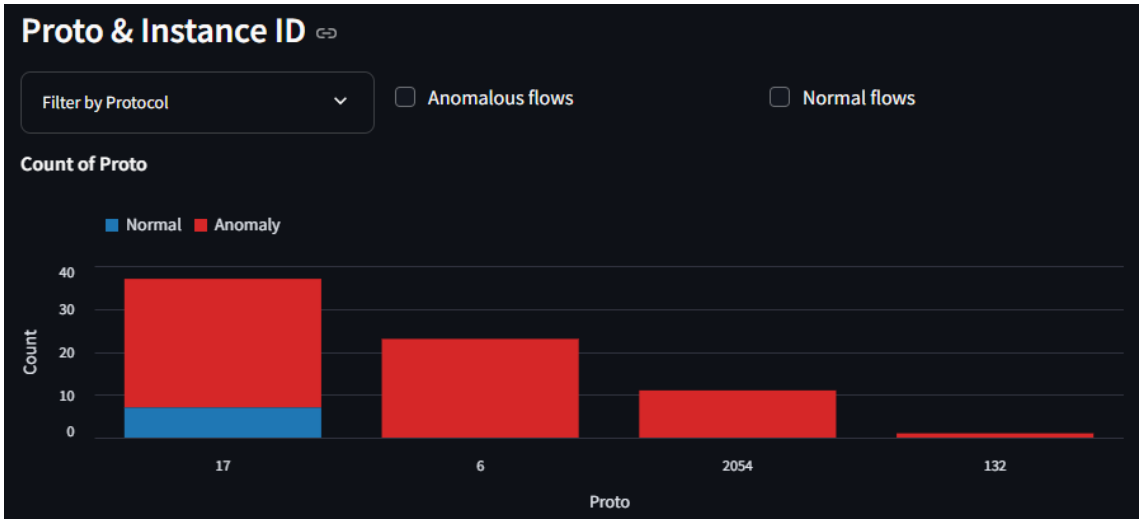


*Figure 19 Dashboard protocol bar chart*

There are also histograms that display the distribution of source and destination ports as well as RMSE values. The data type is the frequency of each port or RMSE range, separated once again by normal and anomalous traffic. In Figure 20, an example of how the distribution of source and destination ports is presented. Similarly, Figure 21 shows the RMSE value distribution with further insights into the classification of the flows.

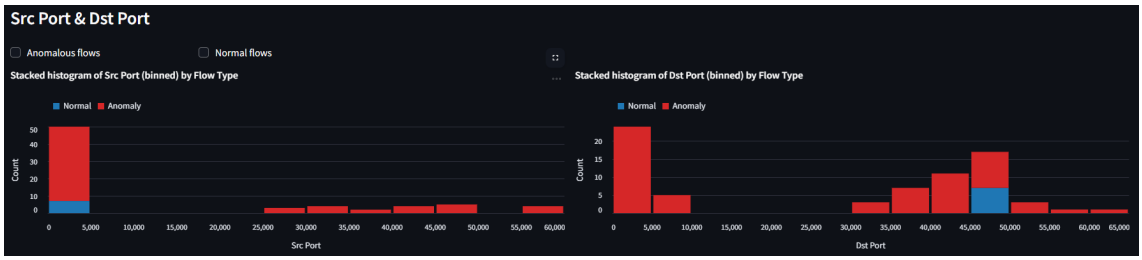


*Figure 20 Dashboard ports histogram*

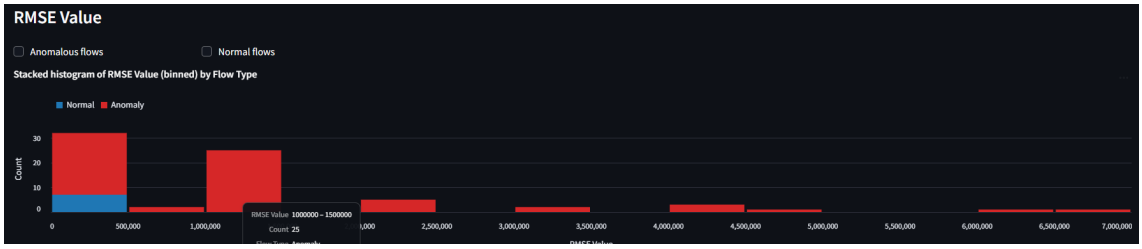


*Figure 21 Dashboard RMSE histogram*

The last part of the dashboard shows Sankey diagrams used to visualize anomalous traffic flows between source and destination addresses, as well as between kube addresses. The data type is the relationship and count of flows between source and destination nodes, as



Funded by Horizon Europe Framework Programme of the European Union

shown in Figure 22. This visualization allows users to see how anomalous traffic flows through the network and helps to identify critical communications.

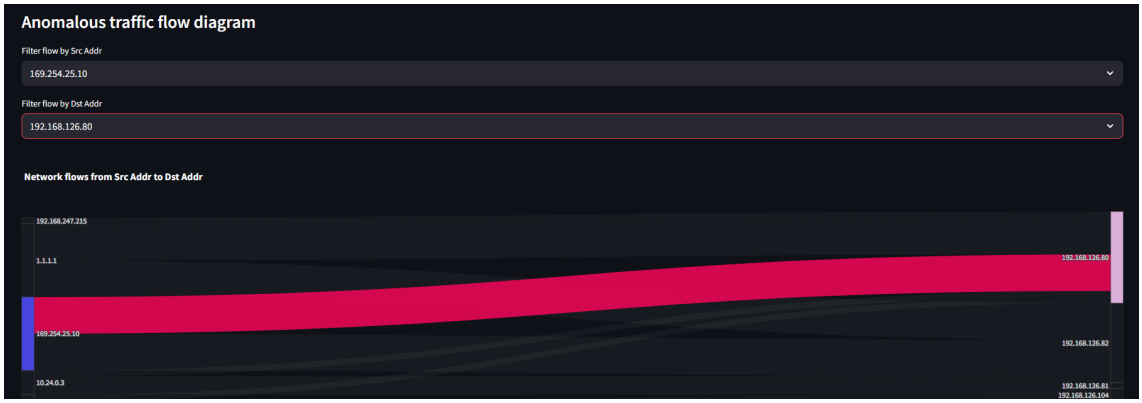


*Figure 22 Dashboard traffic flow Sankey diagram*

## 6.2 Pilot Integration Outcomes

For Pilot 1, the Dashboard was deployed on an EVIDEN/ATOS cloud instance along with LADS and LOMOS, due to resource constraints on RGB's server. However, as explained in section 2, these limitations allowed us to extend the adaptability of our solution so that deployment is not limited to on-prem but also to hybrid environments.

For Pilot 2, the Dashboard was deployed on the physical server installed in the hospital. In this environment, the limitation was access to services to check proper functioning and improvements continuously. However, a solution was found where, through MCI, the necessary assistance could be provided to the system and updated without interfering with the ongoing clinical trials.

**Licence and download**

The Dashboard has a proprietary license owned by ATOS Spain released on a case-by-case basis beyond the pilot usages and trials.

The component artifacts (e.g. code, documentation, images...) can be downloaded or pulled from the project's development Github repository: https://github.com/Cylcomed/Dashboard.

The repositories can be opened on invitation. To request access to this repository, please contact Esteban Armas (esteban.armas@eviden.com).

## 6.3 Lessons Learned

Running on-premises is usually preferred for compliance and data protection, but it needs more setup and care (sizing, backups, patching). A hybrid approach helped development: partners could test new views in a VM and only promote them to the hospital server once approved. When logs might include sensitive fields, we kept processing on-prem and only shared aggregated metrics for remote checks.

Finally, the modular design is key. Treating each CYLCOMED Tool component as a separate, well-defined entity allows us to add features without breaking pilots.



Funded by Horizon Europe Framework Programme of the European Union

# 7 Conclusions and Future Work

WP5 set out to design, build, and integrate a toolbox that healthcare providers can actually run. The final system meets that goal. We now have a coherent set of components—AI detection (LADS, LOMOS), device integrity and service management, identity and authorisation (LuS4MED, C-OPA), data protection (FE4MED), and a single dashboard—that work together on-premise and in hybrid testbeds. The pilots showed that the toolbox can be added to legacy ICU equipment without changing the device firmware and can operate within hospital policies for telemedicine. Using containers and automation reduced site-to-site differences and made roll-backs and updates predictable.

However, two practical limits remain. First, model training needs representative “normal” data ; access to that data can be constrained, and training windows must cover typical behaviour to avoid false alerts. Second, hardware varies across sites; GPU support improves training speed, but the system must run well on standard CPUs.

Based on what we learned, we propose the following for future work:

- **Wider interoperability:** Prepare for IEEE 11073 SDC adoption so the integrity module and OTA tooling can talk to future devices with less custom work.
- **Security features and audits:** Extend audit trails across components (who accessed what and when) and provide export of summaries for compliance teams.
- **Usability and training**: Keep the dashboard simple by default and add “advanced” options only where needed. Develop tailored training materials for hospital staff on toolbox use, focusing on quick anomaly response and basic config, to build awareness without deep tech skills.
- **Tool adaptability:** Improve tools like LOMOS to run in varied environments, e.g., reduce GPU needs for training or handle shorter data windows when access is limited.



Funded by Horizon Europe Framework Programme of the European Union

# 8 Bibliography

[1] CYLCOMED, "D5.1 CYLCOMED Toolbox Prototype," EU Commission, 2024.

[2] CYLCOMED, «D5.2 CYLCOMED Toolbox,» EU Commission, 2025.

[3] CYLCOMED, «D6.1 Pilot planning and evaluation strategy incl. clinical trial submission package,» EU Commission, 2023.

[4] CYLCOMED, «D6.2 Pilots initial phase report incl. midterm recruitment report,» EU Commission, 2024.

[5] CYLCOMED, «D3.1 Baseline analysis, requirements and specifications,» EU Commission, 2023.

[6] CYLCOMED, «D3.2 Requirements and specifications consolidation,» EU Commission, 2024.

[7] CYLCOMED, «D6.3 Pilot final phase report,» EU Commission, 2025.

[8] EU Commission, «REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL,» 2016. [En ligne]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679. [Accès le 25 08 2025].

[9] EU Commission, «REGULATION (EU) 2017/745 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL,» 05 04 2017. [En ligne]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:02017R0745-20250110. [Accès le 25 08 2025].

[10] EU Commission, «REGULATION (EU) 2017/746 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL,» 05 04 2017. [En ligne]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32017R0746. [Accès le 25 08 2025].

[11] ISO, «ISO 14971:2019,» 2019. [En ligne]. Available: https://www.iso.org/obp/ui/#iso:std:iso:14971:ed-3:v1:es:sec:A. [Accès le 25 08 2025].

[12] Streamlit, «Streamlit Documentation,» 2025. [En ligne]. Available: https://docs.streamlit.io. [Accès le 25 08 2025].

[13] J. VanderPlas, B. Granger, J. Heer, D. Moritz, K. Wongsuphasawat, A. Satyanarayan, E. Lees, I. Timofeev, B. Welsh et S. Sievert, «Altair: Interactive Statistical Visualizations for Python,» *Journal of Open Source Software,* vol. 3, n° %132, pp. 1-2, 2017.

[14] Plotly, «Plotly,» 2013. [En ligne]. Available: https://plotly.com/python/. [Accès le 25 08 2025].





Funded by Horizon Europe Framework Programme of the European Union

[15] PyPi, «streamlit-aggrid,» 2023. [En ligne]. Available: https://pypi.org/project/streamlit-aggrid/. [Accès le 25 08 2025].

[16] J. Z. Z. Z. a. M. R. L. P. He, «Drain: An online log parsing approach with fixed depth tree,» *IEEE international conference on web services (ICWS),* vol. 2017, pp. 33-40, 2017.

[17] S. Y. a. X. W. H. Guo, «LogBERT: Log Anomaly Detection via BERT,» *International joint conference on neural networks (IJCNN),* pp. 1-8, 2021.

[18] M. W. C. K. L. a. K. T. J. Devlin, «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,» chez *CoRR*, 2018, p. vol. abs/1810.04805.

[19] Health Sector Cybersecurity Coordination Center, «Healthcare Sector DDoS Guide,» U.S. Department of Health and Human Services, 2024.

[20] W. Zhijun, L. Wenjing, L. Liang et Y. Meng, «Low-Rate DoS Attacks, Detection, Defense, and Challenges: A Survey,» *IEEE Access,* vol. 8, pp. 43920-43943, 2020.

[21] V. Vedula, P. Lama, R. Boppana et L. Trejo, «On the Detection of Low-Rate Denial of Service Attacks at Transport and Application Layers,» *Electronics,* vol. 10, p. 2105, 2021.

[22] OpenID, "OpenID 4 Verifiable Presentations," OpenID, 2024. [Online]. Available: https://openid.net/specs/openid-4-verifiable-presentatios-1_0.html. [Accessed 3 2025].

[23] OpenID, «OpenID for Verifiable Credential Issuance,» 2024. [En ligne]. Available: https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html. [Accès le 3 2025].

[24] CYLCOMED, «D2.1 Analysis of Ethical, Legal and Data Protection Frameworks,» EU Commission, 2023.

[25] CYLCOMED, «D2.2 Examination of Ethical, Legal and Data Protection Requirements,» EU Commission, 2023.

[26] CYLCOMED, «D3.3 Guidance improvement - intermediate report,» EU Commission, 2024.



Funded by Horizon Europe Framework Programme of the European Union