Deliverable D2.5

AMOE – v2

Editor(s):	Franz Deimling (FABA)
Responsible Partner:	Fabasoft R&D GmbH
Status-Version:	Final – v1.0
Date:	31.10.2025
Type:	OTHER (SW)
Distribution level:	PU

Project Number:	101120688
Project Title:	EMERALD

Title of Deliverable:	D2.5 AMOE – v2
Due Date of Delivery to the EC	31.10.2025

Workpackage responsible for the Deliverable:	WP2 - Methodology for knowledge extraction
Editor(s):	Franz Deimling (FABA)
Contributor(s):	Angela Fessl (KNOW)
Reviewer(s):	Angela Fessl (KNOW) Cristina Martínez, Juncal Alonso (TECNALIA)
Approved by:	All Partners
Recommended/mandatory readers:	WP1, WP2, WP3, WP4 and WP5

Abstract:	Final evidence extraction from policy documents that can be integrated with the certification graph
Keyword List:	Evidence extraction, policy documents, organisational metrics
Licensing information:	This work is licensed under Creative Commons Attribution-ShareAlike 4.0 International (**CC BY-SA 4.0 DEED** https://creativecommons.org/licenses/by-sa/4.0/)
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. The European Union cannot be held responsible for them.

Document Description

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
v0.1	25.09.2025	First draft version	Franz Deimling (FABA)
v0.2	10.10.2025	Added contents, updated sections and figures	Franz Deimling (FABA)
v0.2	17.10.2025	Quality Assurance Review	Angela Fessl (KNOW)
v0.3	29.10.2025	Addressed internal review feedback	Franz Deimling (FABA)
v0.4	29.10.2025	Final review	Cristina Martínez/ Juncal Alonso (TECNALIA)
v0.5	30.10.2025	Addressed final review feedback	Franz Deimling (FABA)
v1.0	31.10.2025	Final version to be submitted to the European Commission	Cristina Martínez/ Juncal Alonso (TECNALIA)

Table of contents

Terms and abbreviations 6
Executive Summary 7
1 Introduction 8
1.1 About this deliverable 8
1.2 Document structure 8
1.3 Updates from D2.4 9
1.4 Technological advances from the MEDINA project 9
2 Implementation 11
2.1 Functional description 11
2.1.1 Fitting into the overall EMERALD Architecture 15
2.2 Technical description 16
2.2.1 Prototype architecture 16
2.2.2 Technical specifications 18
2.3 Testing and quality management of evidence extraction method 19
2.3.1 Annotation setup 19
2.3.2 Test setup 20
2.4 Limitations and future work 21
3 Delivery and usage 23
3.1 Package information 23
3.2 Installation instructions 23
3.3 Instructions for use 24
3.4 Licensing information 27
3.5 Download 27
4 Conclusions 28
5 References 29

List of tables

Table 1. Overview of deliverable updates with respect to D2.4 9
Table 2. AMOE.01 - Upload PDF document 11
Table 3. AMOE.02 - Provision of extracted evidence to Evidence Store 12
Table 4. AMOE.03 - Refine evidence extraction approach 12
Table 5. AMOE.04 - Compare results from multiple documents 13
Table 6. AMOE.05 - Select metrics per document 13
Table 7. AMOE.06 - Classify document, select respective metrics (optional) 14
Table 8. AMOE.07 - Metric states 14
Table 9. Overview of AMOE's source code package contents 23

List of figures

FIGURE 1. EMERALD COMPONENT OVERVIEW DIAGRAM (D1.2 [10]) ... 16
FIGURE 2. AMOE ARCHITECTURE DIAGRAM ... 17
FIGURE 3. SCREENSHOT OF ANNOTATIONS IN INCEPTION ... 20
FIGURE 4. AMOE EVIDENCE EXTRACTION METHOD/APPROACH COMPARISON ... 21
FIGURE 5. TARGET OF EVALUATION: EXTRACTED EVIDENCE FOR A SET OF METRICS CHOSEN (D4.4 [13]) ... 25
FIGURE 6. TARGET OF EVALUATION: OVERVIEW PAGE OF THE UPLOADED DOCUMENTS (D4.4 [13]) ... 25
FIGURE 7. EVIDENCE DETAIL PAGE (D4.4 [12]) ... 26
FIGURE 8. UPLOAD POLICY DOCUMENT AND SELECT METRICS (D4.4 [12]) ... 26

Terms and abbreviations

AI	Artificial Intelligence
AMOE	Assessment and Management of Organisational Evidence
API	Application Programming Interface
BDR	Business Driven Requirement
CPU	Central Processing Unit
CSP	Cloud Service Provider
DB	Database
DoA	Description of Action
EC	European Commission
GA	Grant Agreement to the project
GPU	Graphical Processing Unit
GUI	Graphical User Interface
ID	Identifier
HTML	Hypertext Markup Language
IaaS	Infrastructure as a Service
KPI	Key Performance Indicator
KR	Key Result
MEDINA	Predecessor project of EMERALD
NLP	Natural Language Processing
nDCG	normalized Discounted Cumulative Gain
PaaS	Platform as a Service
PDF	Portable Document Format
QA	Question Answering
RCM	Repository of Controls and Metrics
SD	Sequence Diagram
SW	Software
ToE	Target of Evaluation
TRL	Technology Readiness Level
UI	User Interface
WP	Work Package

Executive Summary

This deliverable presents the final design, architecture, and implementation state of the *Assessment and Management of Organisational Evidence* (*AMOE) component*, an evidence extractor for policy documents. The main contributions are related to the key result KR1-EXTRACT of EMERALD, a framework to continuously extract knowledge from different layers of a target of evaluation and prepare suitable evidence based on them.

The policy document evidence extractor (*AMOE*), developed in Task 2.3 "Extraction of evidence out of policy documents" and described in this deliverable, aims at identifying relevant text segments related to security related features, as defined in the respective EMERALD metrics based on specific controls and security requirements of various security schemes. The extracted evidence is stored in the EMERALD *Evidence Store*. Other related deliverables in WP2, all due at project month 24 (October 2025), provide functional and technical details on further evidence extractors from different sources, i.e., D2.3 [1] on source code evidence extraction in Task 2.2, D2.7 [2] on security and privacy preserving evidence extraction in Task 2.4, D2.9 [3] on runtime data extraction in Task 2.5. All these details contributed to D2.1 [4] on the overall information model of the certification graph in Task 2.1.

This document starts by illustrating how the policy document evidence extractor (*AMOE*) fits into the overall EMERALD architecture. The main part provides functional and technical descriptions of the evidence extractor *AMOE*, including its purpose and scope, the (current and planned) coverage of the EMERALD requirements, the components' internal architecture and their subcomponents. These descriptions are complemented by information on delivery and usage, as well as on limitations and future work. Finally, the document concludes with a short summary.

Based on the work described in this deliverable, the policy document evidence extractor is integrated into the EMERALD framework. This is the second and final iteration of the deliverable coming from Task 2.3. The previous version of this deliverable (D2.4 [5]) was delivered in month 12 (October 2024).

1 Introduction

EMERALD aims to offer a suite of tools and techniques for evidence extraction, leveraging a knowledge graph-based approach. KR1-EXTRACT facilitates a unified, tool-supported methodology for continuously extracting knowledge across various layers of a target of evaluation (ToE) - such as infrastructure, platform, runtime data, policy documents, software, and AI models.

The goal of WP2 is to develop a cohesive view of the target of evaluation being certified by extracting and enriching knowledge from these layers and generating relevant evidence for security metrics. A key focus of this work package is the research and design of tools and techniques to extract knowledge from diverse sources. Central to this is the *Evidence Store*, utilizing a graph-based model that acts as a common structure, populated by all evidence extraction tools with evidence[1].

1.1 About this deliverable

The goal of this deliverable is to present the EMERALD evidence extractor tool *AMOE* and how it is integrated into the EMERALD framework. This report reflects the current prototype of *AMOE*, which was originally launched in MEDINA[2]. In EMERALD, it should be advanced to a higher TRL and improved to verify that the functionality is adapted to the needs of the EMERALD pilot use cases.

EMERALD follows a central evidence management approach to provide a unified view of the ToE under certification at different layers of the service. The different evidence extraction tools are ranging from the infrastructure layer (e.g., virtual resources), to the business layer (e.g., policies and procedures), to the implementation layer (e.g., source code files), and the data layer (e.g., increasingly used AI models) in cloud applications. *AMOE* focuses on providing evidence based on policy documents which shall be included into the whole automated certification process. This deliverable gives insights into the technical and functional approach that *AMOE* uses to support the key results of the project (e.g. the use cases demonstrated by the pilots as well as the technical integration via the Certification graph and the evidence extraction workflows of EMERALD).

1.2 Document structure

The document is structured as follows. In Section 2, the functional and technical descriptions of *AMOE* are described. This covers the requirements for *AMOE* in the EMERALD project as well as the *AMOE* architecture description and how it fits into the whole EMERALD architecture. Furthermore, an overview of the testing and quality management for evidence extraction is described alongside the annotation setup. Additionally, this section includes limitations and future work to be commenced regarding *AMOE*.

Section 3 focuses on the delivery and usage of *AMOE*. First the package and its contents are described, followed by installation and deployment instructions. Second, this section also provides some instructions of use, licensing information, and where to download the current public version.

The deliverable is concluded in Section 4, followed by some references in Section 5.

[1] For details consult the *AMOE* and the *Evidence Store* data model presented in deliverable D1.2 [9]

[2] https://medina-project.eu/

1.3 Updates from D2.4

This deliverable evolves from D2.4 AMOE-v1 [5]. Making the document self-contained and easier to follow, parts of the content comes from D2.4 [5] since they have not changed, and other parts are new. To simplify tracking progress and updates from the previous version, Table 1 shows a summary of the changes and additions related to each section of the document.

Table 1. Overview of deliverable updates with respect to D2.4

Section	Changes
1 Introduction	Added new subsections “Updates from D2.4”and “Technological advances from the MEDINA project”.
2.1 Functional description	The requirements and their status were updated. The architecture overview was updated in section 2.1.1.
2.2.1.1 Sub-components description	Updated details of the sub-components.
2.3.2 Test setup	The test setup and evaluation results were updated.
2.4 Limitations and future work	The future work and limitations were updated to the current state of the project.
3.2 Installation instructions	An environment variable was added.
3.3 Instructions for use	The figures and respective descriptions were updated.
4 Conclusions	Updated conclusion.
5 References	Updated references.

1.4 Technological advances from the MEDINA project

EMERALD leverages the results of the MEDINA project: *Security framework to achieve a continuous audit-based certification in compliance with the EU-wide cloud security certification scheme* (GA 952633) [6]. The goal is to advance the existing systems and increase the TRL to make the tools more approachable for different user groups, including nonexperts. The modular system shall become more integrated and accessible via the joint workflows in the EMERALD user interface – based on the set of different tool APIs.

The focus in EMERALD is to improve the functionality of *AMOE* and provide all features via an API. The following points describe the most prominent additions / updates since the previous achievements in MEDINA [7].

Achievements in MEDINA:

- Basic webservice / UI (no API, only backend)
- Upload documents, process all available metrics for the document (time and resource demanding)
- Document pre-processing optimized for MEDINA use case data
- Evidence extraction and assessment directly via the UI, submission of results manually
- No multi-pod[3] support (no parallel processing/handling of requests)
- Local file storage (files not available from other instances of *AMOE*)

[3] https://kubernetes.io/docs/concepts/workloads/pods/

New achievements in EMERALD:

- Main usage of *AMOE* is now via an external UI utilizing the updated API endpoints
- The process of extracting metrics has been changed – now only a subset of metrics will be extracted on demand (via API), instead of the full set when uploading a file
- The document pre-processing pipeline was improved and adjusted to the pilot documents
- The evidence extraction methods were also improved and tested with new data from the pilots.
- The integration into the *EMERALD UI* and the workflows defined by WP4 has been implemented.
- Upcoming: the integration to the updated *Evidence Store* using the metrics from the collaborative metric repository (see D3.4 [8]) and updated evidence handling in EMERALD (that differs from MEDINA) can be implemented as soon as the dependencies are fulfilled.
- *AMOE* evidence states (e.g. successfully extracted, failed, ready for audit, submitted to *Evidence Store*, …) have been added to the tool.

2 Implementation

The following subsections provide functional and technical descriptions of *AMOE*.

2.1 Functional description

Overall purpose. *AMOE* is based on a prototype developed in a previous project called MEDINA[2]. It is designed to extract evidence based on metrics, which target specific parts of policy documents. After the extraction process, the evidence can be inspected in a GUI (Graphical User Interface) that comes with *AMOE* or retrieved via the API. Once the evidence results have been reviewed by a user, they can be forwarded to the EMERALD framework.

Context, scope and motivation. *AMOE* allows to transform the organisational process of checking policy documents for their content into a technical process. Text passages can be checked against predefined goals and target values. As policy documents are rather static, compared to other evidence gathered (e.g. log files, runtime information), the evidence gathering is done once per document for a specific set of metrics and target values. In case updates are required, the new document is processed, and additional evidence is produced. The policy evidence results are integrated into the EMERALD audit process via submission to the *Evidence Store* component and subsequent processing in the *Assessment* component.

Requirements. The relevant requirements from D1.4 [9] with their respective implementation progress (degree of implementation - *partially / fully / not implemented*) and a brief description of how they are / will be implemented are given in Table 2 to Table 8.

Table 2. AMOE.01 - Upload PDF document

Field	Description
Requirement ID	AMOE.01
Short title	Upload PDF document.
Description	The component shall be able to receive a PDF document via API and process its contents regarding the defined metrics. The PDF shall receive a unique ID so that it can be retrieved and deleted later.
Status	Implemented
Priority	Must
Component	*EMERALD UI, AMOE*
Source	Component
Type	Technical
Related KR	KR1_EXTRACT, KR2_CERTGRAPH, KR8_PILOTS
Related KPI	KPI 1.1
Validation acceptance criteria	The user can upload a document via API. The user shall be able to retrieve document meta data by using the unique id that is returned on successful upload. The process shall finish in reasonable time.
Progress	100% - Fully implemented
Milestone	MS2: Components V1 (M12)

AMOE provides the functionality to upload a PDF document via its API. The uploaded file receives a unique id that can be used to retrieve the metadata and evidence linked to the file.

Table 3. AMOE.02 - Provision of extracted evidence to Evidence Store

Field	**Description**
Requirement ID	AMOE.02
Short title	Provision of extracted evidence to the *Evidence Store*
Description	The evidence extraction component needs to be able to forward the extracted evidence to the EMERALD *Evidence Store*, so it can be used for assessment and further audit processes.
Status	Work in Progress
Priority	Must
Component	*AMOE, Clouditor-Evidence Store*
Source	Component
Type	Technical
Related KR	KR1_EXTRACT, KR2_CERTGRAPH, KR8_PILOTS
Related KPI	KPI 1.1
Validation acceptance criteria	A user with permissions to forward evidence shall be able to use the API to submit the extracted evidence to the *Evidence Store*. The process shall finish in reasonable time.
Progress	50% - Partially implemented (testing with EMERALD deployment is open, and might also require some additional changes)
Milestone	MS5: Components V2 (M24)

The functionality was tested in the MEDINA[2] framework. The adjustment to the EMERALD data model and *Evidence Store* API remains to be fully implemented. It has been tested on a subset of metrics – however, for this to be finished the metrics need to be available in the *RCM* and *Evidence Store*. Also, the process must be tested within the EMERALD deployment, before it can be considered fully implemented.

Table 4. AMOE.03 - Refine evidence extraction approach

Field	**Description**
Requirement ID	AMOE.03
Short title	Refine evidence extraction approach
Description	The evidence extraction approach should be refined to the needs of the pilots, so that the tool is able to provide relevant evidence for the metric assessments.
Status	Work in Progress
Priority	Should
Component	*AMOE*
Source	Component
Type	Technical
Related KR	KR1_EXTRACT, KR2_CERTGRAPH, KR8_PILOTS
Related KPI	KPI 1.1
Validation acceptance criteria	Users of *AMOE* should be able to view a documentation text or diagram showing the performance or results of the extraction approach. The performance shall be tuned to metrics and policy documents to be provided by the pilots - including information/annotations of targets in the documents.
Progress	70% - Partially implemented

Milestone	MS5: Components V2 (M24)

The refinement of the extraction approach has been implemented. The extraction processes need to be fine-tuned to data and metrics related to and provided by the pilots. The quality assessment and annotation processes have been set up and can be used to progress and evaluate the evidence extraction approach. This feature is ready to be tested by the pilots via the *EMERALD UI*. The feedback received can then be incorporated.

Table 5. AMOE.04 - Compare results from multiple documents

Field	**Description**
Requirement ID	AMOE.04
Short title	Compare results from multiple documents
Description	Results from multiple policy documents shall be comparable using *AMOE*. A metric can be used to extract evidence from different policy documents. *AMOE* shall provide the results via API for a metric and given cloud service.
Status	Work in Progress
Priority	Should
Component	*AMOE*
Source	Component
Type	Technical
Related KR	KR1_EXTRACT, KR8_PILOTS
Related KPI	KPI 1.1
Validation acceptance criteria	The user can retrieve the extracted evidence based on a metric via API for different uploaded policy files by supplying the metric id as well as a cloud service id in the request.
Progress	90% - Partially implemented (testing with updated metrics from collaborative metric repository required is required and they need to be retrieved by the *RCM*)
Milestone	MS2: Components V1 (M12)

AMOE provides an API to retrieve evidence results for a target of evaluation (cloud service) and metric id. The testing in the EMERALD environment still must be conducted before this can be considered fully implemented.

Table 6. AMOE.05 - Select metrics per document.

Field	**Description**
Requirement ID	AMOE.05
Short title	Select metrics per document
Description	*AMOE* should offer the possibility to select some metrics before they are extracted for a document. This speeds up the processing time as metrics that are not contained in the document do not need to be checked. Also, it should be more convenient for the user, as the results are more precise, and less irrelevant results need to be discarded.
Status	Work in Progress
Priority	Should
Component	*AMOE, EmeraldUI*
Source	Component

Type	Technical
Related KR	KR1_EXTRACT, KR8_PILOTS
Related KPI	KPI 1.1
Validation acceptance criteria	The user can send a set of metric ids in the upload request or before the extraction process is started to make sure only those metrics are being processed for the uploaded file.
Progress	90% - Partially implemented
Milestone	MS5: Components V2 (M24)

This feature is implemented and can be used via the API. The original *AMOE* GUI has not been updated to include this, as it is discontinued and will only be kept for existing implementations. This feature has been integrated into the *EMERALD UI* and is ready to be tested by the pilot partners. Once the validation of the pilot partners has been done, this requirement will be fully implemented.

Table 7. AMOE.06 - Classify document, select respective metrics (optional)

Field	**Description**
Requirement ID	AMOE.06
Short title	Classify document, select respective metrics (optional)
Description	*AMOE* could use document classification to pre-select some metrics based on the category, text, requirements or other features that would be of use. This could potentially, reduce the manual workload and help to provide only results for metrics that target the specific document.
Status	Discarded
Priority	Could
Component	*AMOE*
Source	Component
Type	Technical
Related KR	KR1_EXTRACT, KR2_CERTGRAPH, KR8_PILOTS
Related KPI	KPI 1.1
Validation acceptance criteria	The component owner can configure this option in *AMOE* if it is implemented. For the different categories, different metrics need to be defined. The validation steps need to be defined more clearly, once the requirement is implemented.
Progress	0% - Not implemented
Milestone	MS8: Integrated audit suite V3 (M34)

The implementation of this requirement has been discarded because it goes beyond the scope of the EMERALD project The feature was considered optional from the start, and after re-evaluating the UI and pilot requirements, it can be safely discarded. Moreover, given the limited data available to train a model for this functionality, there is a high risk of overfitting to the pilot scenarios. The feature would only be valuable if it could generalize across a wide range of documents, which is currently not the case.

Table 8. AMOE.07 - Metric states

Field	**Description**
Requirement ID	AMOE.07
Short title	Metric states

Description	*AMOE* could add some internal states to the metrics. This should help to visualize the current process for every metric and role. Here is a list of metric flags that could be used: • new: the metric has been successfully extracted • extraction-failed: evidence could not be extracted • internal-started: internal auditor/compliance manager started inspecting the metric • ready-for-audit: internal auditor/compliance manager has finished with the metric, and marked it ready for auditor • revise-policy: auditor sets the metric to be revised • audit-finished: auditor is ok with the metric • result-outdated: automatic or manual triggered check if result is outdated
Status	Work in Progress
Priority	Could
Component	*AMOE, EmeraldUI*
Source	Component
Type	Technical
Related KR	KR1_EXTRACT, KR8_PILOTS
Related KPI	KPI 1.1
Validation acceptance criteria	The user can retrieve the metric state / evidence state information via API. The state should change given the defined strategy - this change could be obtained depending on the actual state on different times using the API.
Progress	90% - Partially implemented Metrics or to be more precise *AmoeEvidence* can have the following states: • **undefined**: if the evidence was created in an outdated version of *AMOE* • **extraction-complete**: the evidence has been successfully extracted • **extraction-complete-no-result-found**: *AMOE* could not find an answer in the document, but there was also no error during extraction • **extraction-failed**: evidence could not be extracted • **internal-started**: internal auditor/compliance manager started by setting a compliance state and or comment for the evidence • **ready-for-audit**: internal auditor/compliance manager has submitted the evidence to the *EvidenceStore*
Milestone	MS5: Components V2 (M24)

Evidence states were introduced to the *AMOE* internal objects to allow users to gain more information about the current state of extraction. The states are available via the API and can be shown in the *EMERALD UI*.

2.1.1 Fitting into the overall EMERALD Architecture

Figure 1 depicts *AMOE's* connections in the overall EMERALD architecture. *AMOE* provides the functionality to add assessment results of organisational requirements/metrics to the EMERALD framework. It works with metrics from the *RCM (Repository of Controls and Metrics)* and accesses the target values from the *Orchestrator* API (metric configuration, if defined). Alternatively, the metrics can be read from a metric configuration file. Once an uploaded policy document is processed and the evidence is extracted and confirmed by a user, it can be forwarded to the *Evidence Store* and handled further by other EMERALD components according to the evidence/assessment pipeline defined. *AMOE* provides its functionalities via an API to the *EMERALD UI*.

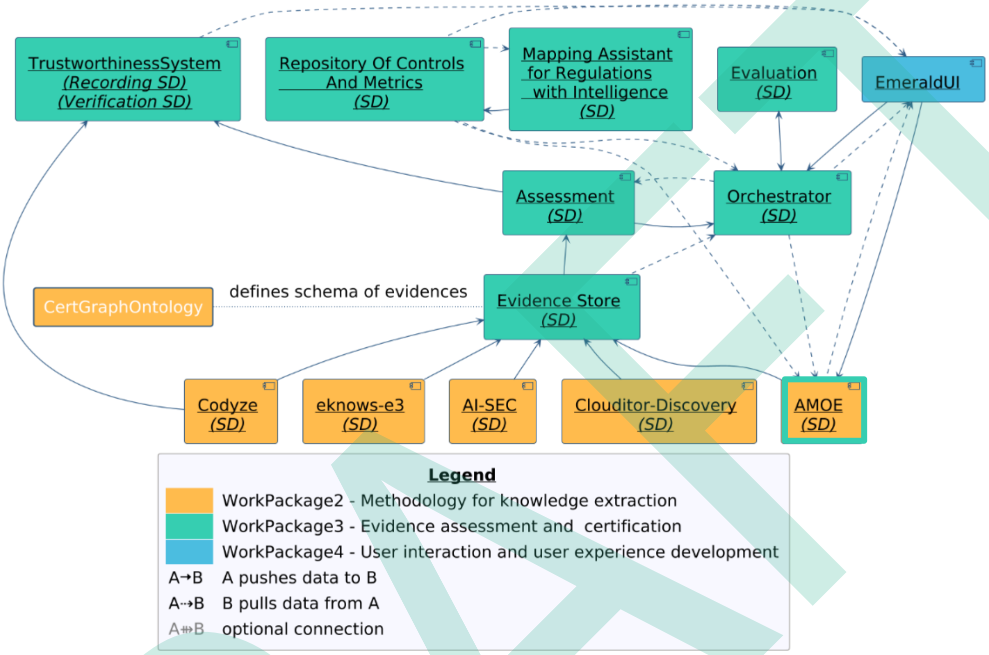


Figure 1. EMERALD component overview diagram (D1.2 [10])

2.2 Technical description

This section describes the technical details of the implemented software.

2.2.1 Prototype architecture

Figure 2 depicts the *AMOE* architecture. The main subcomponent is the webservice built on Quart[4]. The UI and API are the main parts hosted by the webservice. The component is connected to a Keycloak[5] instance for authentication and authorization of the users. To access data of the EMERALD framework, *AMOE* utilizes Python clients generated for the different component APIs based on their respective OpenAPI[6] files (*RCM*, *Orchestrator*, *Evidence Store*).

The MongoDB[7] is connected via the *db utils* wrapper. It consists of three s, one for storing the user action log data, one for the file metadata and one for the extracted evidence. The Redis[8] instance is storing the session data required for the authentication libraries.

[4] https://pypi.org/project/Quart/
[5] https://www.keycloak.org/
[6] https://www.openapis.org/
[7] https://www.mongodb.com/
[8] https://redis.io/

The solid connections in Figure 2 show the components used when deploying *AMOE*. The dashed connections (*qa quality checks*) reflect offline parts that are run on demand. A previous version of *AMOE*, developed in the MEDINA[9] project, has been described in [7].

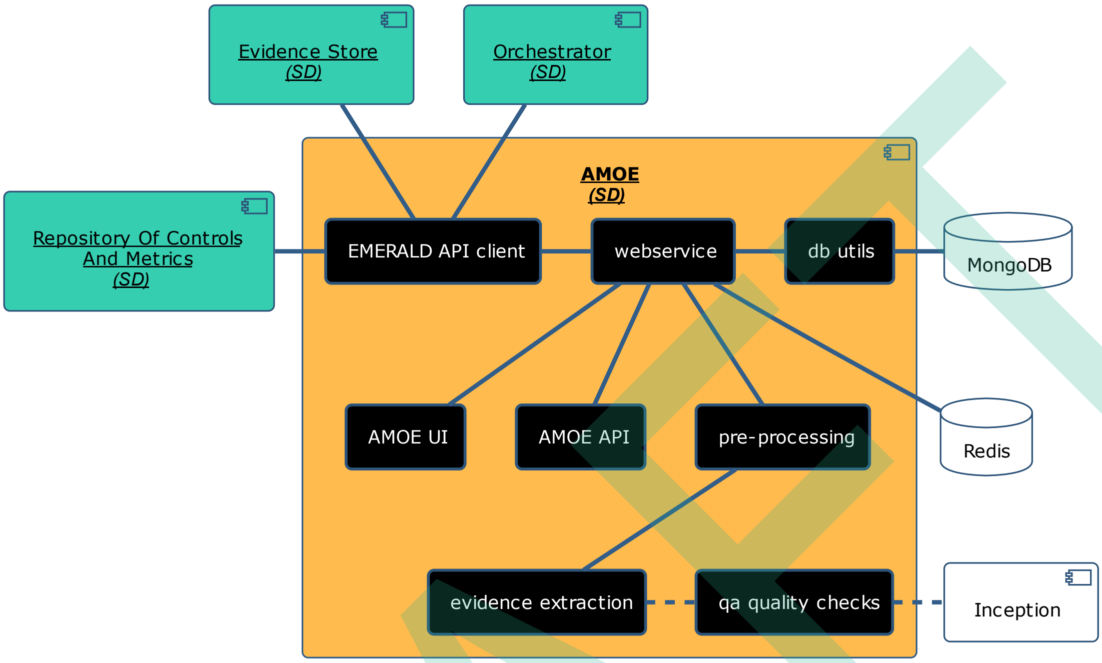


Figure 2. AMOE architecture diagram

2.2.1.1 Sub-components description

This section describes the sub-components of *AMOE*. Figure 2 depicts how the sub-components are related to each other, and the lines reflect how they are related / how data flows between the sub-components.

Webservice

This is the core component redirecting the data flow to the relevant subcomponents. It serves the *AMOE* UI and API and verifies authentication via the Keycloak[5] instance from EMERALD.

EMERALD API client

This subcomponent is used to access the different components of the EMERALD framework. It is used to retrieve requirements and metrics from the *Repository of Controls and Metrics* and the metric configuration from the *Orchestrator*. Furthermore, it is used for submitting the assessment results and extracted evidence to the *Evidence Store*.

DB utils

This subcomponent is used to store and access evidence results as well as local assessment results. It is also used to log relevant information such as by whom and when a document has been uploaded, or an assessment result has been changed.

[9] https://medina-project.eu

AMOE UI

The graphical user interface serves to upload documents as well as to access the processed evidence. It enables the user to search, filter, and manage the organisational evidence. This UI was developed mainly in MEDINA and will not be further developed in EMERALD. Only small changes have been introduced to provide additional support for debugging the tool. *AMOE* will be used mainly via its API through the *EMERALD UI*.

AMOE API

The API enables data access for other applications such as the *EMERALD UI*. It can be used to perform the most essential functions of *AMOE*. These include uploading a document, retrieving the processed evidence, setting assessment results, and submitting the evidence results to the *Evidence Store*.

Pre-processing

This subcomponent is triggered in a background process once a document has been uploaded. It performs the necessary transformations (PDF to HTML conversion, removal of header/footer, stop word removal, …) to enable the evidence extraction.

Evidence extraction

This subcomponent is triggered after the pre-processing pipeline is done. In MEDINA[10] different approaches for evidence extraction in *AMOE* were tested. The keyword-based approach is currently active due to showing historically the best performance in the test results. It uses a set of predefined keywords linked to specific organizational metrics to find relevant sections of documents during a cloud audit. For each metric (e.g., password policy), some keywords are identified (e.g., “password”, “age”, “maximum”), and the tool scans policy documents to extract matching sections by using the section headings. After this initial retrieval, the section text serves as input for a question answering (QA) system alongside the metric question. QA is a natural language processing (NLP) task - a trained model of a QA system can provide answers to a question, given a question and text as input. The top answer of the QA is provided by the system. The keyword-based approach might be replaced by the “WHOLE_DOC” approach as a default setting for *AMOE*, given the findings of the test results (see Section 2.3.2). Like this approach, but using an alternative pre-trained model is “WHOLE_DOC_MDBERTA”. More details and alternative approaches tested (e.g. using cosine-similarity), are described in the paper [11] and in the *AMOE* source code.

QA quality checks

The question answering (QA) quality check subcomponent enables the user to compare the extracted evidence (using e.g. the keyword-based approach) with the annotations exported from the INCEpTION tool. See Section 2.3 for details on the quality management process.

2.2.2 Technical specifications

The *AMOE* tool is written in Python >=3.12. It uses various Python libraries as well as the *pdftohtml* functionality from *poppler utils*[11]. The webservice is built on *Quart*[12], the evidence

[10] https://medina-project.eu/

[11] https://poppler.freedesktop.org/

[12] https://pypi.org/project/quart/

extraction is based on *transformers*[13], *PyTorch*[14] and the *roberta-base-squad2*[15] model from *Hugging Face*.

The component is using *MongoDB*[16] and *Redis*[17] to store the data. Evidence and logs are stored in the *MongoDB*. *Redis* is used in parallel with the *quart-session* library.

2.3 Testing and quality management of evidence extraction method

To test the evidence extraction method used in *AMOE*, the extracted data is compared to some predefined target values. This allows us to compute some scores that can be used to adjust the settings of the approach. The quality management is thus split into two parts: 1) the annotation setup and 2) the execution of tests, evaluation, and analysis of the results.

2.3.1 Annotation setup

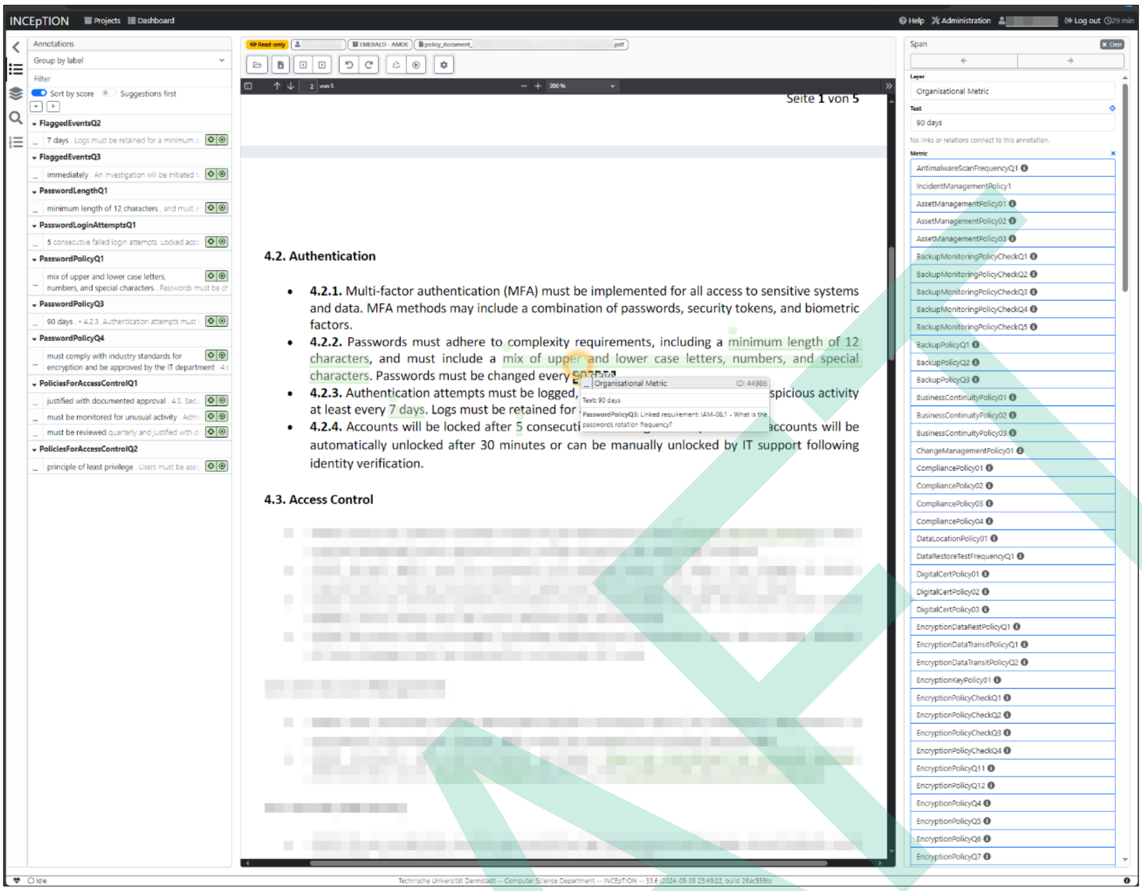
AMOE's evidence extraction targets relevant texts of policy documents based on metrics. To test the evidence extraction pipeline of *AMOE*, one needs to gather data. The data in this case consist of a set of documents and text passages that represent the evidence. The evidence text passages can be marked/defined by using specific annotation tools. The resulting annotations can then be compared against the results of *AMOE*. The annotation represents a link of the metric id to a start and end position of the full document text.

The data annotation process and the preparational steps consists of three parts. First, the policy documents (PDFs) need to be gathered. Second, the annotation software INCEpTION[18] needs to be set up and running. Third, an annotation project can be set up by adding the list of metrics to the tag set and configuring the layers to be annotated.

The tag set for the metrics can be generated using an excel list/csv containing the metric ids with the support of the *python* program supplied in the *AMOE* source code "`src/extract_metric_tag_list.py`".

After the INCEpTION project is set up, the policy documents can be uploaded, and users can annotate the files. To annotate, the text must be selected and afterwards a metric can be assigned. If the metric or annotation is hovered, more information is displayed like for example the description that has been configured in the tag set. Once the document has been annotated, the curation process can be applied to ensure high quality data. In the curation step, multiple annotations by different users are combined into a single source of truth, which can be exported and used for the quality assessment of the evidence extraction approach. Figure 3 depicts a screenshot of the annotation view in INCEpTION. In the specific case shown the annotated text samples are highlighted in green.

[13] https://github.com/huggingface/transformers
[14] https://pytorch.org/
[15] https://huggingface.co/deepset/roberta-base-squad2
[16] https://www.mongodb.com/
[17] https://redis.io/
[18] https://inception-project.github.io/



FlaggedEventsQ2
- 7 days . Logs must be retained for a minimum o

FlaggedEventsQ3
- immediately . An investigation will be initiated t

PasswordLengthQ1
- minimum length of 12 characters , and must i

PasswordLoginAttemptsQ1
- 5 consecutive failed login attempts. Locked acc

PasswordPolicyQ1
- mix of upper and lower case letters, numbers, and special characters . Passwords must be ch

PasswordPolicyQ3
- 90 days . • 4.2.3. Authentication attempts must

PasswordPolicyQ4
- must comply with industry standards for encryption and be approved by the IT department . 4.

PoliciesForAccessControlQ1
- justified with documented approval . 4.5. Sec
- must be monitored for unusual activity . Adm
- must be reviewed quarterly and justified with d

PoliciesForAccessControlQ2
- principle of least privilege . Users must be ass

4.2. Authentication

- **4.2.1.** Multi-factor authentication (MFA) must be implemented for all access to sensitive systems and data. MFA methods may include a combination of passwords, security tokens, and biometric factors.
- **4.2.2.** Passwords must adhere to complexity requirements, including a minimum length of 12 characters, and must include a mix of upper and lower case letters, numbers, and special characters. Passwords must be changed every 90 days
- **4.2.3.** Authentication attempts must be logged, spicious activity at least every 7 days. Logs must be retained for
- **4.2.4.** Accounts will be locked after 5 consecuti accounts will be automatically unlocked after 30 minutes or can be manually unlocked by IT support following identity verification.

Organisational Metric ID: 44986
Text: 90 days
PasswordPolicyQ3: Linked requirement: IAM-08.1 - What is the passwords rotation frequency?

4.3. Access Control

[redacted]

Layer: Organisational Metric
Text: 90 days
No links or relations connect to this annotation.
Metric:
- AntimalwareScanFrequencyQ1
- IncidentManagementPolicy1
- AssetManagementPolicy01
- AssetManagementPolicy02
- AssetManagementPolicy03
- BackupMonitoringPolicyCheckQ1
- BackupMonitoringPolicyCheckQ2
- BackupMonitoringPolicyCheckQ3
- BackupMonitoringPolicyCheckQ4
- BackupMonitoringPolicyCheckQ5
- BackupPolicyQ1
- BackupPolicyQ2
- BackupPolicyQ3
- BusinessContinuityPolicy01
- BusinessContinuityPolicy02
- BusinessContinuityPolicy03
- ChangeManagementPolicy01
- CompliancePolicy01
- CompliancePolicy02
- CompliancePolicy03
- CompliancePolicy04
- DataLocationPolicy01
- DataRestoreTestFrequencyQ1
- DigitalCertPolicy01
- DigitalCertPolicy02
- DigitalCertPolicy03
- EncryptionDataRestPolicyQ1
- EncryptionDataTransitPolicyQ1
- EncryptionDataTransitPolicyQ2
- EncryptionKeyPolicy01
- EncryptionPolicyCheckQ1
- EncryptionPolicyCheckQ2
- EncryptionPolicyCheckQ3
- EncryptionPolicyCheckQ4
- EncryptionPolicyQ11
- EncryptionPolicyQ12
- EncryptionPolicyQ4
- EncryptionPolicyQ5
- EncryptionPolicyQ6
- EncryptionPolicyQ7

Figure 3. Screenshot of annotations in INCEpTION

2.3.2 Test setup

Several tests can be conducted in the project to check the performance of the evidence extraction approach and tune it to the metrics/documents of the EMERALD project. The exported annotations (ground truth) can be compared to the results of *AMOE*. The most basic score that can be computed is based on the number of matches vs the total number of metrics annotated. The evidence extraction approach shall be evaluated by computing this score per pilot. For example, if for a document 28 metrics have been annotated in INCEpTION (#of annotated evidence) and *AMOE* retrieves the correct answer for 19 metrics (#correctly retrieved evidence), the resulting ratio would be ca. 0.68.

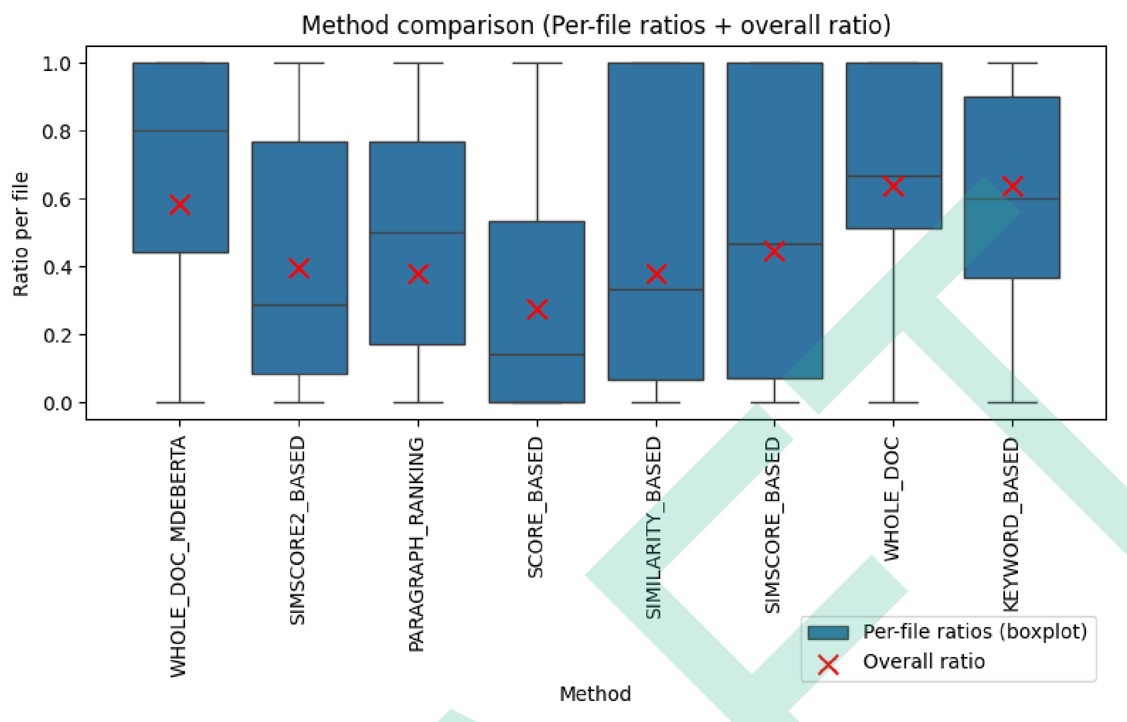


Figure 4. AMOE evidence extraction method/approach comparison

For pilot 3 (FABA), 11 files have been annotated for which 37 out of 58 metrics (ratio=0.64) could be retrieved for the WHOLE_DOC and WHOLE_DOC_MDBERTA approaches (see 2.2.1.1 evidence extraction). The other approaches had lower scores, as you can see in Figure 4. The figure shows box plots of the ratios per file that were annotated for every evidence extraction approach. The red "x" shows the overall ratio for each method based on the total number of correctly retrieved evidence text (matching the annotation) vs. the number of annotated samples. The approaches using the whole document performed better than the others (e.g. keyword based). This is probably due to the documents being shorter in general as used in the previous experiments (and outdated approaches) described in the *AMOE* paper [11]. More tests are required to further improve the performance.

Based on the feedback of the pilots, different strategies will be applied to further adapt the best working approach. The approach is configurable, and so it is adjustable to different domains. This means, that a user can adapt *AMOE* to use an approach that works better on longer documents or shorter if that is required.

2.4 Limitations and future work

AMOE is quite useful to check for policy document details, however, there remain some issues that can be solved in future updates with extended data sets and test pipelines. One key challenge is the limited availability of data, which restricts the system's ability to provide comprehensive answers across diverse domains. Additionally, language-related limitations arise, as the model may struggle to extract relevant responses in languages it has not been trained on. Privacy concerns also play a significant role, particularly since the tool avoids using pre-trained publicly available AI models due to data protection requirements, further constraining the variety and quality of data inputs. The models are selected based on their license as well as local operability – to make sure the data does not have to leave the premise. The prototype operates under limited GPU and CPU resources, which hinders the processing speed and scalability of the system, affecting its overall efficiency and performance when handling complex queries or large datasets. The processing time of the current evidence extraction approach is dependent on the input size – longer documents potentially take longer to process. However, this is somewhat mitigated by applying keywords to reduce the search space (see also [11]). Furthermore, empirical tests via the *EMERALD UI* have been shown that, especially with multiple deployed pods[19] for *AMOE,* the processing time is reasonably low (however, this was not assessed by any benchmark tests). The whole tool was adjusted to allow for parallel processing of different files, using multiple deployments in Kubernetes. This addresses the BDRP4.02 [12] – in order the *AMOE* parts of the platform/*EMERALD UI* run fluidly for the end-users.

AMOE is designed to support the assessment and management of policy documents, but not to fully automate the assessment process at all. With the current design, no guarantee can be given that the AI models would always retrieve the correct evidence. Given the limited data set provided to the project by the different partners, results might be biased for specific use cases/metrics that are only relevant to some of the partners. The performance will be evaluated given the provided resources, since the focus of the project is on innovation rather than research.

At the current state of the project, not all *AMOE* requirements have been fully implemented. AMOE.07 lists a few possible metric states that have been added to improve the traceability and usability of metrics. The remaining (partially implemented) *AMOE* requirements will be fully implemented until the end of the EMERALD project. This will be reflected in a future version of this deliverable. Furthermore, at the current state of *AMOE* it is impossible to determine for sure whether the target of a metric is contained in a document. To limit the processing resources and time waiting for results and, most of all, to extract more precise results, the AMOE.05 was implemented – this allows a user to select a set of metrics per file to avoid processing non-sensical data.

The largest remaining requirement is AMOE.03 – the improvement of the evidence extraction approach. For the remainder of the project all open requirements will be worked on, except AMOE.06 as it has been discarded. The *AMOE* GUI will not be updated further as it will be integrated into the *EMERALD UI*. More tests via the *EMERALD UI*, especially by the pilot partners are required. The feedback will then be incorporated (if in the scope of the project) before the partially implemented requirements are considered fully implemented.

[19] https://kubernetes.io/docs/concepts/workloads/pods/

3 Delivery and usage

The following sections give a short overview of the delivery and usage of the *AMOE* component.

3.1 Package information

AMOE can be deployed as a Docker container. Table 9 shows an overview of the repository folders and files.

Table 9. Overview of AMOE's source code package contents

Folder	Description
`/`	The root folder contains some helper scripts and configurations needed to build and run *AMOE* (e.g. Dockerfile).
`/kubernetes/`	Contains the Kubernetes[20] files for the deployment of *AMOE*.
`/metric_data/`	Contains the local version of the metrics.
`/src`	Contains the source code of *AMOE.*
`/src/paragraph_extraction/`	Contains the code for the pre-processing pipeline.
`/src/qa/`	Contains the code for evidence extraction using the question answering model as well as code to compute quality scores.
`/src/static/`	Contains the stylesheets and images for the webservice.
`/src/templates/`	Contains the HTML templates for the webservice.
`/src/utils/`	Contains code for utility functions of the webservice such as use of other EMERALD component's API, evidence extraction and database management.
`/tests/`	Contains the tests for the *AMOE* API
`/clouditor-evidence-client/`	Contains the repo for the generated Python client for the *Evidence Store* API based on their OpenAPI file.
`/orchestrator-client/`	Contains the repo for the generated Python client for the *Orchestrator* API based on their OpenAPI file.
`/rcm-client/`	Contains the repo for the generated Python client for the *Repository of Controls and Metrics* API based on their OpenAPI file.

3.2 Installation instructions

Clone the *AMOE* repository using git or use any other means to download the source code. Set up a MongoDB and a Redis instance (see Kubernetes files in the repository). For Kubernetes deployment the instructions can be found on the official web page (https://kubernetes.io/docs/concepts/workloads/controllers/deployment/). For building the docker container the instructions on the docker web page (https://docs.docker.com/build/) can be followed.

Set the following environment variables or variables directly in the `config.py`:

- `MONGODB_URL`
- `MONGODB_PORT`

[20] https://kubernetes.io/docs/concepts/workloads/controllers/deployment/

- MONGODB_USER
- MONGODB_PASSWORD
- REDIS_SERVICE
- REDIS_PASS
- REDIS_PORT
- KEYCLOAK_URL
- KEYCLOAK_REALM
- KEYCLOAK_CLIENT_ID
- KEYCLOAK_CLIENT_SECRET
- KEYCLOAK_USER
- KEYCLOAK_PASSWORD

Optionally set (needed to deploy in production with EMERALD components):

- CATALOGUE_API_URL
- ORCHESTRATOR_API_URL
- EVIDENCE_STORE_API_URL
- ALLOWED_ORIGINS

Run the command

```
python3 -m hypercorn -b 0.0.0.0 "src.app:create_app()"
```

to deploy the service locally or deploy it with Kubernetes.

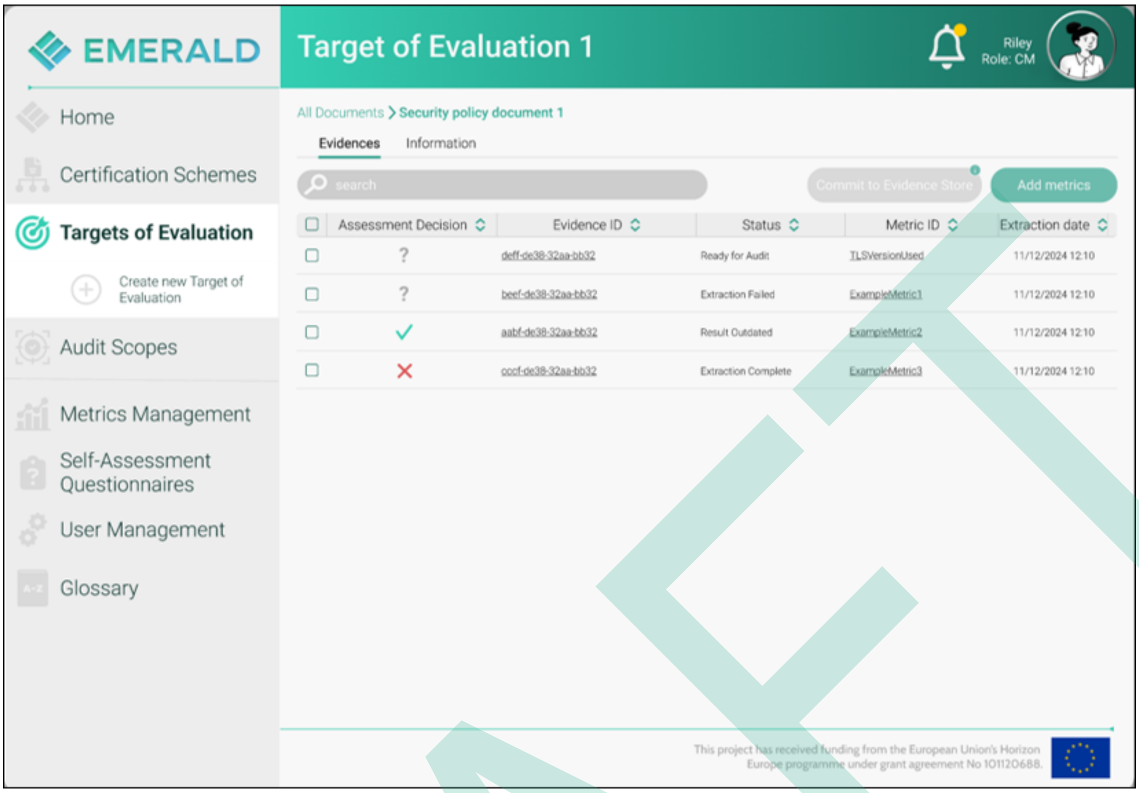
3.3 Instructions for use

Instructions for use are included in the *AMOE*'s public GitLab repository[21]. The OpenAPI documentation can be found in the code repository in GitLab as well as retrieved for every deployment by accessing `<amoe-server-url>/docs`.

Based on the mock-ups developed in WP4, the *AMOE* pages have been implemented in the *EMERALD UI*. Figure 5 depicts the extracted evidence for a set of metrics chosen by a user. Figure 6 shows the overview page of uploaded documents for a target of evaluation. Figure 7 depicts the evidence detail page for users that are allowed to read this data but not change it. The upload dialog and the option to add more metrics to be processed for a file is shown in Figure 8.

The normal procedure to use *AMOE* via the *EMERALD UI* is to use the upload functionality by selecting a policy PDF and a set of metrics. After this, the system starts to extract evidence for the given set of metrics. The results can be viewed for the files, and a compliance status and comment can be set in a dedicated view.

[21] https://git.code.tecnalia.dev/emerald/public/components/amoe-assessment-and-management-of-organizational-evidence



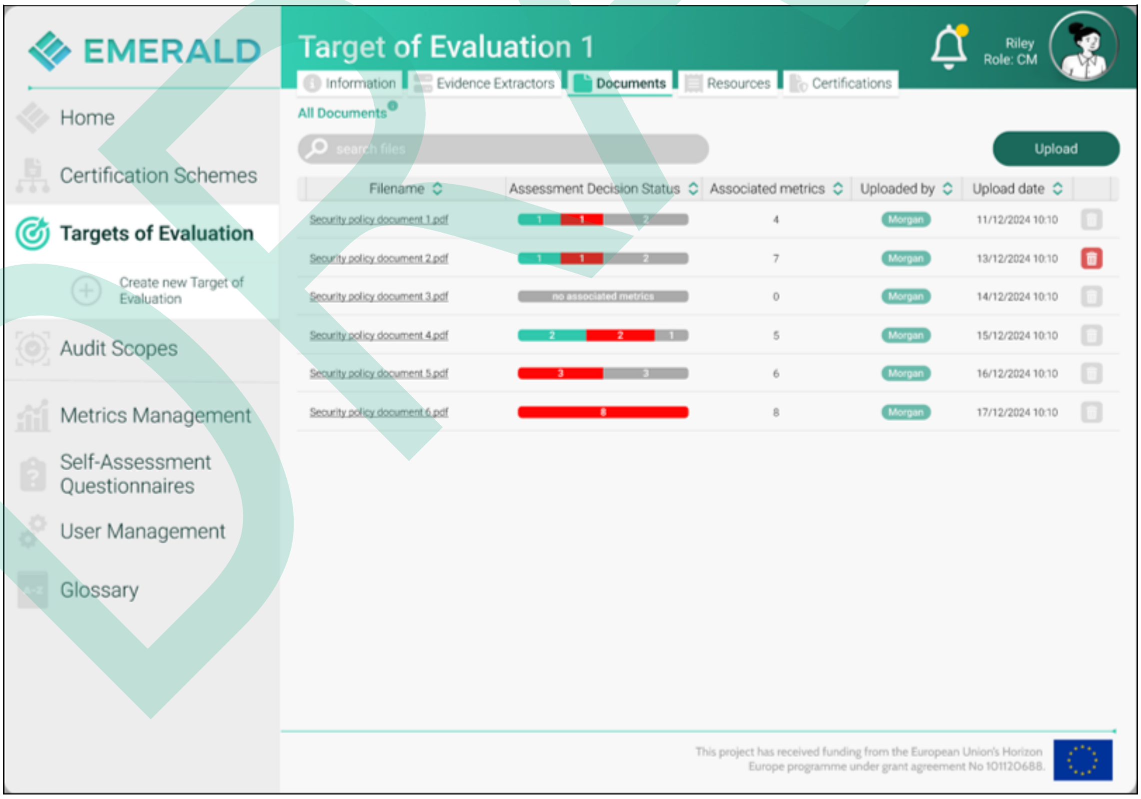
Target of Evaluation 1

All Documents > Security policy document 1

□	Assessment Decision	Evidence ID	Status	Metric ID	Extraction date
□	?	deff-de38-32aa-bb32	Ready for Audit	TLSVersionUsed	11/12/2024 12:10
□	?	beef-de38-32aa-bb32	Extraction Failed	ExampleMetric1	11/12/2024 12:10
□	✓	aabf-de38-32aa-bb32	Result Outdated	ExampleMetric2	11/12/2024 12:10
□	✗	cccf-de38-32aa-bb32	Extraction Complete	ExampleMetric3	11/12/2024 12:10

This project has received funding from the European Union's Horizon Europe programme under grant agreement No 101120688.

Figure 5. Target of Evaluation: Extracted Evidence for a set of metrics chosen (D4.4 [13])



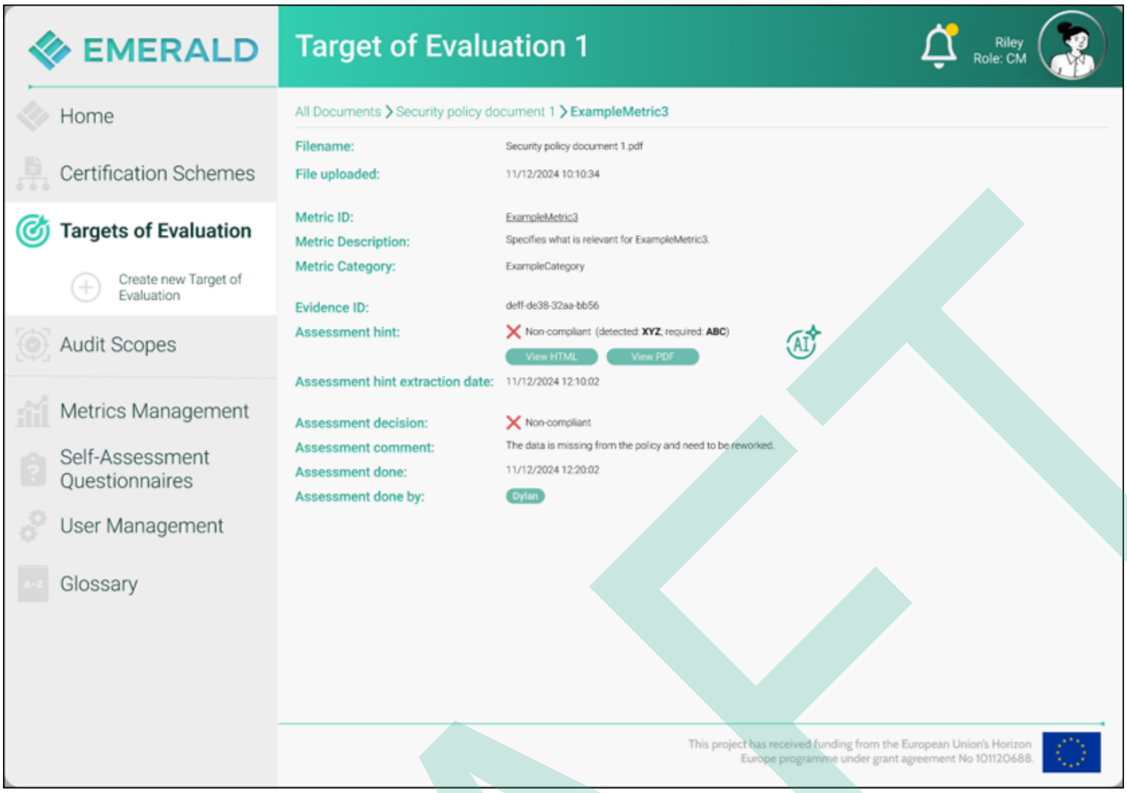
Target of Evaluation 1

All Documents

Filename	Assessment Decision Status	Associated metrics	Uploaded by	Upload date
Security policy document 1.pdf	🟩 1 🟥 1 ▨ 2	4	Morgan	11/12/2024 10:10
Security policy document 2.pdf	🟩 1 🟥 1 ▨ 2	7	Morgan	13/12/2024 10:10
Security policy document 3.pdf	no associated metrics	0	Morgan	14/12/2024 10:10
Security policy document 4.pdf	🟩 2 🟥 2 ▨ 1	5	Morgan	15/12/2024 10:10
Security policy document 5.pdf	🟥 3 ▨ 3	6	Morgan	16/12/2024 10:10
Security policy document 6.pdf	🟥 8	8	Morgan	17/12/2024 10:10

This project has received funding from the European Union's Horizon Europe programme under grant agreement No 101120688.

Figure 6. Target of Evaluation: Overview Page of the Uploaded Documents (D4.4 [13])


EMERALD
Target of Evaluation 1
Riley
Role: CM
Home
Certification Schemes
Targets of Evaluation
Create new Target of Evaluation
Audit Scopes
Metrics Management
Self-Assessment Questionnaires
User Management
Glossary
View HTML
View PDF
AI
This project has received funding from the European Union's Horizon Europe programme under grant agreement No 101120688.

All Documents > Security policy document 1 > ExampleMetric3

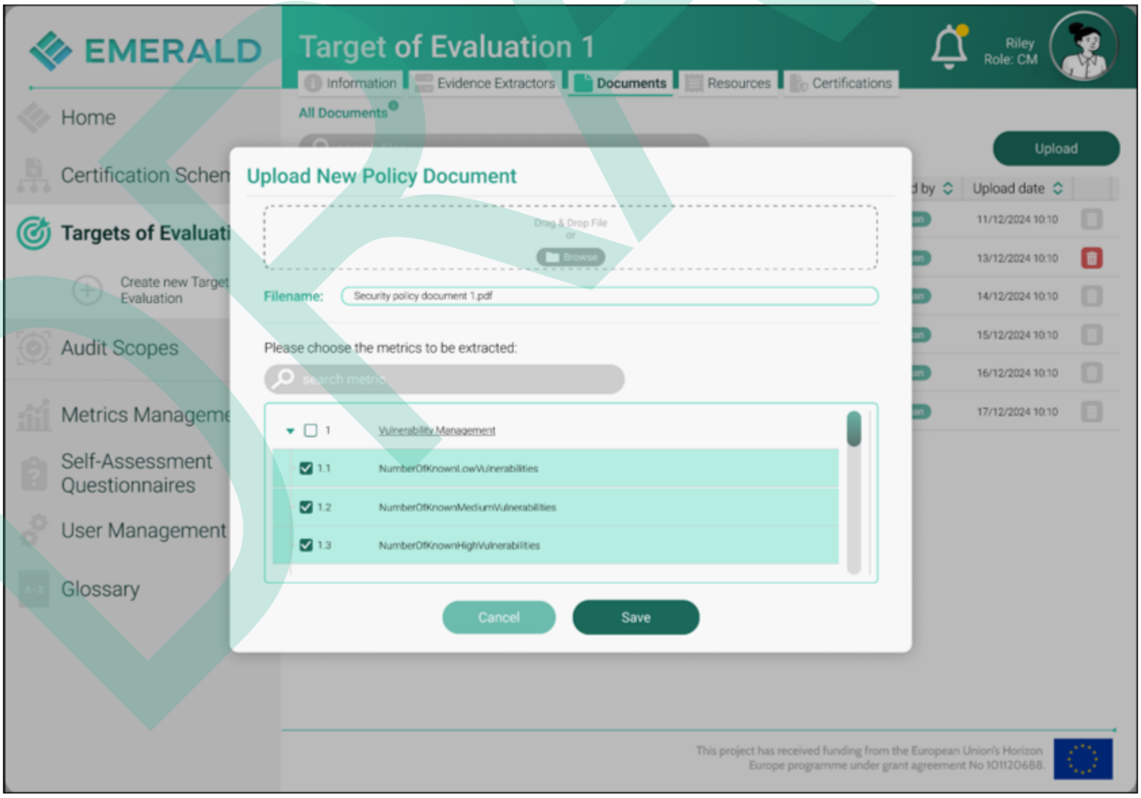
Filename: Security policy document 1.pdf
File uploaded: 11/12/2024 10:10:34

Metric ID: ExampleMetric3
Metric Description: Specifies what is relevant for ExampleMetric3.
Metric Category: ExampleCategory

Evidence ID: deff-de38-32aa-bb56
Assessment hint: ✗ Non-compliant (detected: **XYZ**; required: **ABC**)
Assessment hint extraction date: 11/12/2024 12:10:02

Assessment decision: ✗ Non-compliant
Assessment comment: The data is missing from the policy and need to be reworked.
Assessment done: 11/12/2024 12:20:02
Assessment done by: Dylan

Figure 7. Evidence detail page (D4.4 [12])



Upload New Policy Document

Filename: Security policy document 1.pdf

Please choose the metrics to be extracted:

- □ 1 Vulnerability Management
- ☑ 1.1 NumberOfKnownLowVulnerabilities
- ☑ 1.2 NumberOfKnownMediumVulnerabilities
- ☑ 1.3 NumberOfKnownHighVulnerabilities

Figure 8. Upload policy document and select metrics (D4.4 [12])

3.4 Licensing information

The component is licenced under Apache 2.0.

3.5 Download

The component code can be downloaded from the EMERALD public Gitlab repository[22].

[22] https://git.code.tecnalia.dev/emerald/public/components/amoe-assessment-and-management-of-organizational-evidence

4 Conclusions

AMOE is designed to extract relevant information based on EMERALD metrics from different policy documents provided by the project partners. In this deliverable, the technical report of the EMERALD evidence extraction component *AMOE* is presented. The functional description and how *AMOE* fits into the general EMERALD framework is described. Furthermore, a list of sub-components is given and how they interact.

AMOE is using the APIs of other EMERALD components - the *Repository of Controls and Metrics* to retrieve information of the metrics and security schemes, the *Orchestrator* to retrieve specific target values and metric configurations, and the *Evidence Store* to integrate the extracted results into the EMERALD framework. The clients are generated using the respective OpenAPI files.

AMOE is built using Python and different NLP libraries and pre-trained AI models. The basic requirements are under development, and the functionalities are offered to the *EMERALD UI* via a dedicated *AMOE* API. The partially implemented requirements have progressed quite far and will be finalized until the end of the project. The pilot feedback from the validation activities in WP4 and WP5 will be incorporated into the future adjustments.

5 References

[1] EMERALD Consortium, "D2.3 Source Evidence Extractor – v2," 2025.

[2] EMERALD Consortium, "D2.7 ML model certification – v2," 2025.

[3] EMERALD Consortium, "D2.9 Runtime evidence extractor – v2," 2025.

[4] EMERALD Consortium, "D2.1 Graph Ontology for Evidence Storage: Description of a uniform schema for storing and linking heterogenous data," 2024.

[5] EMERALD Consortium, "D2.4 AMOE–v1," 2024.

[6] MEDINA consortium, "CORDIS - EU research results," 2023. [Online]. Available: https://cordis.europa.eu/project/id/952633. [Accessed 26 09 2025].

[7] MEDINA Consortium, "D3.6 - Tools and techniques for collecting evidence of technical and organisational measures-v3 (https://medina-project.eu/public-deliverables/)," 2023.

[8] EMERALD Consortium, "D3.4 Evidence assessment and Certification–Implementation-v2," 2025.

[9] EMERALD Consortium, "D1.4 EMERALD solution architecture - v2," 2025.

[10] EMERALD Consortium, "D1.2 Data modelling and interaction mechanisms - v2," 2025.

[11] F. Deimling and M. Fazzolari, "AMOE: A Tool to Automatically Extract and Assess Organizational Evidence for Continuous Cloud Audit," In: Atluri, V., Ferrara, A.L. (eds) Data and Applications Security and Privacy XXXVII. DBSec 2023. Lecture Notes in Computer Science, vol 13942. Springer, Cham. https://doi.org/10.1007/978-3-031-37586-6_22, 2023.

[12] EMERALD Consortium, "D5.1 Pilot definition, set-up & validation plan," 2024.

[13] EMERALD Consortium, "D4.4 User interaction and user experience concept–v2," 2025.