



MEDINA

Deliverable D5.5

MEDINA integrated solution-v3

Editor(s):	Debora Benedetto, Claudio Caimi, Ahmed Ibrahim, Claudia Zago
Responsible Partner:	Hewlett Packard Italiana, SRL (HPE)
Status-Version:	Final – v1.0
Date:	03.08.2023
Distribution level (CO, PU):	PU

Project Number:	952633
Project Title:	MEDINA

Title of Deliverable:	MEDINA integrated solution-v3
Due Date of Delivery to the EC	31.07.2023

Work package responsible for the Deliverable:	WP5 - MEDINA Framework Integration
Editor(s):	Debora Benedetto, Claudio Caimi, Ahmed Ibrahim, Claudia Zago (HPE)
Contributor(s):	TECNALIA, Bosch, CNR, Fabasoft, FhG, XLAB
Reviewer(s):	Juncal Alonso (TECNALIA) Cristina Martinez (TECNALIA)
Approved by:	All Partners
Recommended/mandatory readers:	WP2, WP3, WP4, WP6

Abstract:	This deliverable integrates all the components developed by the other technical WPs in the MEDINA framework. Different versions of the solution have been provided following an incremental approach. The first version was an initial prototype with the core functionalities implemented (at M15); the second version (at M27) augmented these functionalities taking into consideration the feedback coming for the use cases; and the final version (M33) includes corrections and feedback coming from the implementation of the use cases. The software is accompanied by a Technical Specification Report. This set of deliverables is the result of Task 5.3.
Keyword List:	Architecture, Workflows, Components Integration, CI/CD, Integrated UI
Licensing information:	This work is licensed under Creative Commons Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0) http://creativecommons.org/licenses/by-sa/3.0/
Disclaimer	This document reflects only the author's views and neither Agency nor the Commission are responsible for any use that may be made of the information contained therein

Document Description

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
v0.1	18.05.2023	Initial TOC	HPE, HPE CDS
v0.2	09.06.2023	Added contributions in Section 1.2, Section 2.1, Section 5	HPE CDS
v0.3	22.06.2023	Partner contributions in Section 4	Fabasoft, XLAB, TECNALIA, CNR
v0.4	03.07.2023	Contributions to Section 3 and Appendix E	TECNALIA, Bosch
v0.5	27.07.2023	FhG contributions in Section 4	FhG
v0.6	27.07.2023	Internal Review	TECNALIA
v0.7	30.07.2023	Addressed all comments from internal review	HPE, HPE CDS
v1.0	31.07.2023	Ready for submission	TECNALIA

Table of contents

Terms and Abbreviations	10
Executive Summary	12
1 Introduction	13
1.1 About this deliverable	13
1.2 Document structure	14
1.3 Updates from D5.4	14
2 MEDINA Test Bed and Secure DevOps Infrastructure	16
2.1 Test Bed environment	16
2.1.1 Component Integration Methodology	16
2.2 Implementation of the CI/CD solution	20
2.2.1 Operating Environment.....	20
2.2.2 Pipelines	22
3 Generic Architectural Workflows.....	25
3.1 Generic MEDINA Workflows	25
3.2 Roles	26
3.3 Authorization Model	27
3.3.1 WF1 – Preparation of Target of Certification (ToC)	27
3.3.2 WF2 – Preparation of MEDINA Components	28
3.3.3 WF3 – EUCS deployment on ToC	29
3.3.4 WF4 – EUCS Preparedness – ToC Self-Assessment	30
3.3.5 WF5 – EUCS Compliance Assessment	31
3.3.6 WF6 – EUCS – Maintenance of ToC certificate	32
3.3.7 WF7 – EUCS –Report on ToC Certificate	33
3.3.8 WF8 – Auditor - Verifiable credentials for certificates (NEW)	34
3.3.9 WF9 – Auditor - Integrity verification (NEW)	34
3.4 Authorization Model for the MEDINA Integrated UI	35
3.4.1 Role diagrams.....	38
4 MEDINA Framework Components and Integration	41
4.1 Catalogue (block #1).....	43
4.1.1 Catalogue of Controls and Metrics.....	43
4.2 Certification Metrics and Language (block #2).....	51
4.2.1 NL2CNL Translator.....	51
4.2.2 CNL Editor	52
4.2.3 DSL Mapper	55
4.3 Risk Assessment and Optimisation Framework (block #3)	55
4.3.1 Risk Assessment and Optimisation Framework (RAOF).....	55

4.4	Continuous Evaluation and Certification Life-Cycle (block #4)	60
4.4.1	Continuous Certification Evaluation	60
4.4.2	Automated Certificate Life Cycle Manager	63
4.4.3	Automated Self-Sovereign Identity-based certificates management	64
4.5	Organizational Evidence Gathering and Processing (block #5)	67
4.5.1	Assessment and Management of Organizational Evidence	67
4.6	Orchestrator and Databases (block #6)	70
4.6.1	Orchestrator and Databases	70
4.6.2	Trustworthiness System	73
4.7	Evidence Collection and Security Assessment (block #7)	78
4.7.1	Evidence Collection	78
4.8	Company Compliance Dashboard (block #8)	82
4.8.1	Implementation and Integration Status	82
4.8.2	Graphical User Interface	83
4.8.3	TRL	84
5	MEDINA Integrated User Interface (block #8)	85
5.1	Implementation	85
5.1.1	Functional description	85
5.1.2	Technical description	88
5.1.3	Delivery and usage	92
6	Conclusions	94
7	References	95
8	APPENDIX A: Operating Environment	98
8.1	Kubernetes Installation and Configuration	98
8.2	Kubernetes Dashboard	100
8.3	Hardware Infrastructure	101
9	APPENDIX B: Webinars	103
9.1	Docker and Kubernetes Webinar with Sample Component Integration example	103
9.2	Keycloak Webinar	104
9.3	Authorization and Filtering Webinar	104
9.4	CI/CD Webinar	105
9.5	Codyze Webinar	105
10	APPENDIX C: Component Integration Rounds	106
10.1	First Round - First integration workshop	106
10.2	Second Round – Continuous Integration	108
11	APPENDIX D: Pipelines	113
12	APPENDIX E: Generic Architectural Workflows	118
12.1	WF1 - Preparation of Target of Certification (ToC)	118

12.1.1 Related Architectural Components	118
12.1.2 Workflow	118
12.2 WF2 - Preparation of MEDINA Components	118
12.2.1 Related Architectural Components	119
12.2.2 Workflow	119
12.3 WF3 - EUCS deployment on ToC	120
12.3.1 Related Architectural Components	120
12.3.2 Workflow	120
12.4 WF4 - EUCS Preparedness – ToC Self-Assessment.....	121
12.4.1 Related Architectural Components	121
12.4.2 Workflow	122
12.5 WF5 - EUCS Compliance Assessment	123
12.5.1 Related Architectural Components	123
12.5.2 Workflow	124
12.6 WF6 - EUCS – Maintenance of ToC certificate	125
12.6.1 Related Architectural Components	126
12.6.2 Workflow	126
12.7 WF7 - EUCS –Report on ToC Certificate	127
12.7.1 Related Architectural Components	127
12.7.2 Workflow	128
12.8 WF8 – Auditor- Verifiable credentials for certificates (NEW)	128
12.8.1 Related Architectural Components	129
12.8.2 Workflow	129
12.9 WF9 - Auditor- Integrity verification (NEW)	130
12.9.1 Related Architectural Components	131
12.9.2 Workflow	131
13 APPENDIX F: Published APIs.....	134
13.1 Component: Catalogue of Controls and Metrics.....	134
13.2 Component: NL2CNL Translator and DSL Mapper	138
13.3 Component: CNL Editor	138
13.4 Component: Risk Assessment and Optimisation Framework	138
13.5 Component: Continuous Certification Evaluation.....	140
13.6 Component: Life Cycle Manager	140
13.7 Component: Automated Self-Sovereign Identity-based certificates management (SSI)	141
13.8 Component: Assessment and Management of Organizational Evidence – AMOE.....	142
13.9 Component: Orchestrator	143
13.10 Component: Trustworthiness System	144

13.11	Component: Evidence Collection (Cloud Discovery).....	145
13.12	Component: Security Assessment (Clouditor)	146
14	APPENDIX G: User Manuals	147

List of tables

TABLE 1.	OVERVIEW OF DELIVERABLE UPDATES WITH RESPECT TO D5.4	15
TABLE 2.	STATUS OF POINT-TO-POINT CONNECTIONS DURING THE THIRD ROUND	18
TABLE 3.	GENERIC MEDINA WORKFLOWS	25
TABLE 4.	MEDINA ROLES AND LEVELS OF VISIBILITY	26
TABLE 5.	MAPPING LOCAL MEDINA USERS AND CLOUD SERVICES	27
TABLE 6.	WORKFLOW 1.....	28
TABLE 7.	RBAC MODEL FOR WORKFLOW 1	28
TABLE 8.	WORKFLOW 2.....	28
TABLE 9.	RBAC MODEL FOR WORKFLOW 2	28
TABLE 10.	RBAC MODEL FOR CATALOGUE UI IN WORKFLOW 2	29
TABLE 11.	WORKFLOW 3.....	29
TABLE 12.	RBAC MODEL FOR ORCHESTRATOR UI IN WORKFLOW 3	29
TABLE 13.	RBAC MODEL FOR CNL EDITOR UI IN WORKFLOW 3.....	30
TABLE 14.	RBAC MODEL FOR AMOE UI IN WORKFLOW 3.....	30
TABLE 15.	WORKFLOW 4.....	30
TABLE 16.	RBAC MODEL FOR SATRA IN WORKFLOW 4	31
TABLE 17.	RBAC MODEL FOR CATALOGUE QUESTIONNAIRE IN WORKFLOW 4	31
TABLE 18.	WORKFLOW 5.....	32
TABLE 19.	RBAC MODEL FOR ORCHESTRATOR IN WORKFLOW 5	32
TABLE 20.	RBAC MODEL FOR AMOE UI IN WORKFLOW 5	32
TABLE 21.	WORKFLOW 7.....	33
TABLE 22.	RBAC MODEL FOR CCE UI IN WORKFLOW 7	33
TABLE 23.	RBAC MODEL FOR ORCHESTRATOR UI IN WORKFLOW 7	33
TABLE 24.	WORKFLOW 8.....	34
TABLE 25.	RBAC MODEL FOR SSI UI IN WORKFLOW 8	34
TABLE 26.	WORKFLOW 9.....	34
TABLE 27.	RBAC MODEL FOR DLT UI IN WORKFLOW 9	34
TABLE 28.	RBAC MODEL FOR INTEGRATED UI (APPLIES TO ALL WFs)	37
TABLE 29.	IUI COMPONENT CARD	86
TABLE 30.	IUI.01 REQUIREMENT	87
TABLE 31.	IUI.02 REQUIREMENT	87
TABLE 32.	IUI.05 REQUIREMENT	87
TABLE 33.	IUI.06 REQUIREMENT	87
TABLE 34.	ROLES AND USERNAMES IMPLEMENTED IN KEYCLOAK	90
TABLE 35.	LIST OF ALL COMPONENTS INTEGRATED IN THE MEDINA IUI	90
TABLE 36.	PACKAGE STRUCTURE	93
TABLE 37.	STATUS OF POINT-TO-POINT CONNECTIONS DURING THE FIRST ROUND	108
TABLE 38.	STATUS OF POINT-TO-POINT CONNECTIONS DURING THE SECOND ROUND	111
TABLE 39.	WF1 DESCRIPTION.....	118
TABLE 40.	WF2 DESCRIPTION.....	119
TABLE 41.	WF3 DESCRIPTION.....	121
TABLE 42.	WF4 DESCRIPTION.....	122
TABLE 43.	WF5 DESCRIPTION.....	124

TABLE 44. WF6 DESCRIPTION	126
TABLE 45. WF7 DESCRIPTION	128
TABLE 46. WF9 DESCRIPTION	129
TABLE 47. WF8 DESCRIPTION	131

List of figures

FIGURE 1. TECHNICAL WEBINARS	17
FIGURE 2. PUBLIC GITLAB – LICENSE	20
FIGURE 3. PRIVATE GITLAB REPOSITORY	21
FIGURE 4. CI/CD TOOLS	22
FIGURE 5. JENKINS DEPLOYMENT DASHBOARD	23
FIGURE 6. CODYZE SECURITY PIPELINE STEP	23
FIGURE 7. DEFECTDOJO SAMPLE CODYZE RESULTS	24
FIGURE 8. HOME PAGE FOR THE NON-AUTHENTICATED CUSTOMER	35
FIGURE 9. HOME PAGE FOR THE PRODUCT AND SERVICE OWNER USER ROLE	36
FIGURE 10. WORKFLOW DIAGRAM FOR THE PRODUCT (SECURITY) ENGINEER USER ROLE	38
FIGURE 11. WORKFLOW DIAGRAM FOR THE IT SECURITY GOVERNANCE USER ROLE	39
FIGURE 12. WORKFLOW DIAGRAM FOR THE PRODUCT AND SERVICE OWNER USER ROLE	39
FIGURE 13. WORKFLOW DIAGRAM FOR THE SECURITY ANALYST USER ROLE.....	39
FIGURE 14. WORKFLOW DIAGRAM FOR THE AUDITOR ROLE.....	40
FIGURE 15. WORKFLOW DIAGRAM FOR A NON-AUTHENTICATED USER.....	40
FIGURE 16. MEDINA ARCHITECTURE AND DATA FLOW	42
FIGURE 17. CONNECTION CATALOGUE-SATRA FOR QUESTIONNAIRE MANAGEMENT	45
FIGURE 18. CATALOGUE GUI	46
FIGURE 19. CATALOGUE - CONTROLS OF THE “ORGANISATION OF INFORMATION SECURITY” CATEGORY	46
FIGURE 20. CATALOGUE - LIST OF REQUIREMENTS	47
FIGURE 21. CATALOGUE - FILTER TO SEARCH FOR “EUCS & HIGH” REQUIREMENTS	47
FIGURE 22. CATALOGUE - FILTER TO SEARCH FOR “CKM” CONTROLS	48
FIGURE 23. CATALOGUE - METRICS IMPLEMENTED FOR THE “OPS-05.3H” REQUIREMENT	48
FIGURE 24. CATALOGUE - DETAILS OF A THE “MALWAREPROTECTIONENABLED” METRIC	49
FIGURE 25. CATALOGUE - IMPLEMENTATION GUIDELINE FOR THE ISP-03.5 REQUIREMENT.....	50
FIGURE 26. CATALOGUE - QUESTIONNAIRE - QUESTIONS FOR THE OIS-01 SECURITY CONTROL, BASIC LEVEL..	50
FIGURE 27. CNL EDITOR – REOS VISUALIZATION.....	54
FIGURE 28. CNL EDITOR – SHOWING A SPECIFIC REO	54
FIGURE 29. RAOF GUI	57
FIGURE 30. RAOF - SELECT TOE	58
FIGURE 31. RAOF - SETUP OF TARGETS OF EVALUATION	58
FIGURE 32. RAOF - LIST OF RESOURCES	58
FIGURE 33. RAOF - REQUIREMENTS TO BE FULFILLED.....	59
FIGURE 34. RAOF - RESULTS OF THE STATIC RISK ANALYSIS	60
FIGURE 35. CCE GUI - EVALUATION TREE	62
FIGURE 36. AN EXAMPLE OF A CERTIFICATE AS DISPLAYED IN THE CERTIFICATES VIEW IN THE ORCHESTRATOR GUI	64
FIGURE 37. SSI FRAMEWORK GUI FOR CSPs	66
FIGURE 38. SSI FRAMEWORK GUI FOR THE CAB	66
FIGURE 39. SSI FRAMEWORK GUI FOR THE CSP CUSTOMERS	67
FIGURE 40. AMOE LANDING PAGE.....	69
FIGURE 41. AMOE FILE OVERVIEW	69
FIGURE 42. AMOE COMPLIANCE STATUS VIEW	70
FIGURE 43. ORCHESTRATOR GUI	71

FIGURE 44. THE VIEW OF A SINGLE CLOUD SERVICE: IT SHOWS THE SERVICE'S NAME, ID, DESCRIPTION, AND OTHER INFORMATION. THE TABS AT THE TOP ALLOW TO CONFIGURE THE CLOUD SERVICE, REVIEW ITS DISCOVERED RESOURCES, REVIEW ITS METRICS, AND ITS ASSESSMENT RESULTS.	72
FIGURE 45. THE ASSESSMENT VIEW OF A CLOUD SERVICE: IT ALLOWS TO FILTER EXISTING ASSESSMENT RESULTS FOR THE SELECTED CLOUD SERVICE FOR DIFFERENT PARAMETERS LIKE RESOURCE TYPE AND TIMEFRAME. ALSO, THE "SHOW MORE INFO" BUTTON REVEALS DETAILS ABOUT AN ASSESSMENT RESULT'S INFORMATION	72
FIGURE 46. THE METRICS VIEW: IN THIS VIEW, CONFIGURED METRICS CAN BE REVIEWED, INCLUDING THEIR REGO CODE THAT POSSIBLY HAS BEEN GENERATED BY THE DSL MAPPER	72
FIGURE 47. MEDINA EVIDENCE TRUSTWORTHINESS MANAGEMENT SYSTEM GUI	75
FIGURE 48. MEDINA EVIDENCE TRUSTWORTHINESS MANAGEMENT SYSTEM - AUTOMATIC VERIFICATION SERVICE FILTERS FOR RECORDED EVIDENCE	76
FIGURE 49. MEDINA EVIDENCE TRUSTWORTHINESS MANAGEMENT SYSTEM - AUTOMATIC VERIFICATION SERVICE RESULTS FOR RECORDED EVIDENCE	76
FIGURE 50. MEDINA EVIDENCE TRUSTWORTHINESS MANAGEMENT SYSTEM - AUTOMATIC VERIFICATION SERVICE FILTER FOR SPECIFIC EVIDENCE	77
FIGURE 51. MEDINA EVIDENCE TRUSTWORTHINESS MANAGEMENT SYSTEM - AUTOMATIC VERIFICATION SERVICE RESULT FOR SPECIFIC EVIDENCE	77
FIGURE 52. CCD MAIN DASHBOARD VIEW	83
FIGURE 53. CCD - IMPORT OF EUCS	83
FIGURE 54. CCD - VIEW ON THE WORKFLOW VERIFICATION INFORMATION	84
FIGURE 55. MEDINA UI ARCHITECTURE.....	88
FIGURE 56. MEDINA JWT FIELDS	89
FIGURE 57. MEDINA LOGIN PAGE	89
FIGURE 58. UI OF COMPONENTS - COMMON STYLES.....	91
FIGURE 59. MEDINA IUI CORNICE	92
FIGURE 60. IUI - ABOUT PAGE FOR THE ADMIN USER ROLE	92
FIGURE 61. KUBERNETES CLUSTER INSTALLATION WITH RKE	98
FIGURE 62. EXCERPT OF MEDINA'S DOCKER REGISTRY	99
FIGURE 63. URL NAMING CONVENTION FOR DEV/TEST ENVIRONMENTS	100
FIGURE 64. SERVICE ACCOUNT TYPE USED FOR THE KUBERNETES DASHBOARD.....	100
FIGURE 65. KUBERNETES DASHBOARD	101
FIGURE 66. KUBERNETES CLUSTER ON THE MEDINA INFRASTRUCTURE	102
FIGURE 67. SPRING SWAGGER TEMPLATE ON GITLAB.....	103
FIGURE 68. SAMPLE PROJECT DEPLOYMENT STEPS.....	103
FIGURE 69. DEMO PROJECT IN THE TEST ENVIRONMENT	104
FIGURE 70. K8S DASHBOARD: COMPONENTS DEPLOYED IN DEV ENVIRONMENT	106
FIGURE 71. STATUS OF THE FIRST INTEGRATION OF THE MEDINA COMPONENTS	107
FIGURE 72. STATUS OF THE SECOND INTEGRATION OF THE MEDINA COMPONENTS	110
FIGURE 73. JENKINS SEED JOB	114
FIGURE 74. PIPELINES	115
FIGURE 75. BUILD PIPELINE	115
FIGURE 76. DEPLOY PIPELINE.....	116
FIGURE 77. DEPLOY PIPELINE WITH AVAILABLE ENV	116
FIGURE 78. SECURITY PIPELINE.....	116
FIGURE 79. CLEAN-CLUSTER PIPELINE.....	117
FIGURE 80. CERTIFICATE MAINTENANCE (SOURCE: EUCS [11]).....	126

Terms and Abbreviations

AMOE	Assessment and management of organizational evidence
API	Application Programming Interface
CAB	Conformity Assessment Body
CCE	Continuous Certification Evaluation
CCD	Company Compliance Dashboard
CCE	Continuous Certification Evaluation
CI/CD	Continuous Integration / Continuous Deployment
CISO	Chief Information Security Officer
CNL	Controlled Natural Language
CSA or EU CSA	Coordination and Support Action
CSP	Cloud Service Provider
CSS	Cascading Style Sheets
DLT	Distributed Ledger Technologies
DoA	Description of Action
DSL	Domain Specific Language
EC	European Commission
EUCS	European Cybersecurity Certification Scheme for Cloud Services
GA	Grant Agreement to the project
gRPC	Google Remote Procedure Call
GUI	Graphical User Interface
HTTPS	Hypertext Transfer Protocol Secure
IaaS	Infrastructure As A Service
IT	Information Technologies
JSON	JavaScript Object Notation
JWT	JSON Web Token
IP	Internet Protocol
IUI	Integrated User Interface
KPI	Key Performance Indicator
KR	Key Result
LCM	Life Cycle Manager
NCCA	National Cybersecurity Certification Authority
NL	Natural Language
NL2CNL	Natural Language To Controlled Natural Language
NLP	Natural Language Processing
OPA	Open Policy Agent
OWASP	Open Web Application Security Project
PaaS	Platform As A Service
RAM	Random Access Memory
RAOF	Risk Assessment and Optimisation Framework
RBAC	Role Based Access Control
RDF	Resource Description Framework
REO	Requirements and Obligations
REST	Representational State Transfer
RKE	Rancher Kubernetes Engine
SaaS	Software-as-a-Service
SARIF	Static Analysis Results Interchange Format

SATRA	Self-Assessment Tool for Risk Analysis
SCA	Software Composition Analysis
SPDX	Software Package Data Exchange
SQL	Structured Query Language
SSH	Secure Shell
SSI	Self-Sovereign Identity
SSL	Secure Sockets Layer
SSO	Single Sign-On
SW	Software
ToE	Target of Evaluation
ToC	Target of Certification
TOM	Technical and Organizational Measure
TRL	Technology Readiness Level
UC	Use Case
UI	User Interface
URL	Uniform Resource Locator
VAT	Vulnerability Assessment Tools
WF	Workflow
VM	Virtual Machine

Executive Summary

This document is the third version of the D5.3 [1], that points out the result of the task 5.3 in M33 (July 2023). The goal of this third version is to have a final stable environment and automated solution for the MEDINA solution including corrections and feedback from the implementation of the use cases.

In this deliverable we present the third version of the MEDINA integrated solution with increased functionalities compared to the initial prototype in M15 and the second prototype in M27, and also taking into consideration the feedback coming from the evaluation in the two MEDINA use cases. The document shows how some of the main objectives of the work package 5 are achieved in relation to the maintenance of the SecDevOps infrastructure for MEDINA and the support of the continuous integration with dedicated meetings, workshops and webinars.

The document reports the same structure of its previous versions, D5.3 [1] and D5.4 [2], highlighting updates or changes in each section, and placing the unchanged parts in the Appendices. First, it recapitulates the current state of the Test Bed environment with hardware and operating details, and the methodology adopted throughout the integration phase of the components in the MEDINA integrated solution exploiting webinars and demos. An overview of the entire integrated environment including the Kubernetes cluster and the CI/CD infrastructure is provided. The document then goes deep into the description of MEDINA CI/CD implemented solution, how it supports the automation of the processes with the pipelines and their stages with a focus on security aspects. Compared to the previous version, two new workflows (WF8 and WF9) have been added to the seven workflows that were presented in D5.4; these workflows provided a new view based on the user roles in MEDINA that has been agreed by the consortium for every component involved in each workflow. For each of the eight building blocks that compose the MEDINA architecture their current status and their published APIs are also reported. The last part of the document is dedicated to the *MEDINA Integrated User Interface*, with updates on its technical implementation and usage.

1 Introduction

This section includes an overview of the context of the deliverable, how it is structured and the updates respect to the previous deliverable D5.4 [2].

1.1 About this deliverable

As stated in the “Introduction” section of D5.2 [3], WP5 “MEDINA framework Integration” has as outcome five deliverables that can be divided in two parallel series:

- Those that define the MEDINA integrated solution in detail (D5.1 [4] and D5.2 [3])
- Those that describe the developed solution (D5.3 [1], D5.4 [2] and D5.5).

This deliverable is the final version of the three deliverables of WP5 dedicated to the *developed* “MEDINA integrated solution” (aka MEDINA framework). It reports about the current status and the advancements achieved on the integration of the MEDINA components and is the result of task T5.3 “System Continuous Integration and Optimization”.

Since this is a self-contained document, the description of the integration strategy and implementation adopted during the whole task can be found here, although the improvements introduced in the last six months, from M27 to M33, have been highlighted. Further details about the updates with respect to D5.4 can be found in section 1.3.

The document starts by describing the details of the hardware infrastructure provided to set up the Test Bed environment and how this environment is implemented and used. The Test Bed environment hosts the MEDINA components, further details about its installation and configuration can be found in the *APPENDIX A: Operating Environment*. Once the Test Bed environment has been set up, partners can release their components and the following sections describe the methodology adopted to achieve this integration. Finally, the current status of the MEDINA framework release and the integration of its component is detailed.

Secondly, the document describes the overall design of the CI/CD solution that has been put in place to support the development and integration activities of the MEDINA framework. This solution foresees three pipelines of build, deploy and security to perform the automation of the integration component.

Thirdly, the document presents the workflows used by the Use Cases to test the correct behaviour of the MEDINA framework. The workflows, including the new WF8 and WF9, are described in detail in the *APPENDIX E: Generic Architectural Workflows*. In this period, partners have focused on the introduction of the user role point of view, implementing the authorization and filtering strategies in the components.

Fourthly, the document presents an overview of the implementation status of each component, explaining the interaction with the other components and providing brief details on the component user interface (if any). In addition, the user manuals of those MEDINA framework components that have a GUI are included in *APPENDIX G: User Manuals*¹.

Finally, the document includes the description of the two MEDINA User Interfaces. On the one hand the *MEDINA Integrated User Interface* (IUI), which is the entry point to access to the MEDINA framework in Use Case 1, led by Bosch. And on the other hand, the *Company Compliance Dashboard* (CCD), that has been implemented to support Use Case 2, led by Fabasoft, with the purpose of demonstrating how MEDINA achieves a high level of modularity

¹ The user manual of each tool is also available in the MEDINA IUI by clicking on the “Help” menu option.

through its components and several core APIs, such that potential customers are able to integrate MEDINA seamlessly into their own ecosystem.

1.2 Document structure

The rest of the document is structured as follows:

Section 2 presents the Test Bed Environment, describing its configuration and the hardware infrastructure provided, the description of the methodology adopted for the component integration through the “Keycloak”, “Authorization and Filtering” and “CI/CD” webinars, and the current status of the integration of components. It then describes the implementation and strategy adopted for the CI/CD solution.

Section 3 describes the generic workflows based on nine example scenarios with related architectural components. These workflows are described from the point of view of authorization and filtering and are presented from perspective of the user’s role and permissions.

Section 4 presents the MEDINA framework components. There is a sub section for each block describing all components that belong to it. Each component is presented with an overview of its scope in MEDINA, its implementation status, and its integration with the other MEDINA components, and its TRL. If available, its graphical user interface is also described.

Section 5 is dedicated to the *MEDINA Integrated User Interface* component, which is the component implemented in Work Package 5.

Finally, Section 6 reports the conclusions.

The Appendices sections are dedicated to topics that have not changed much from D5.4 [2] or are too extensive to be included in the main sections of the document. They are structured as follows:

- *APPENDIX A: Operating Environment*, describes the installation and configuration of the Kubernetes cluster into the Test Bed environment and the final results achieved.
- *APPENDIX B: Webinars*, describes the webinar organized for the explanation of the main aspects and operations of Docker and Kubernetes and the demonstration through a demo example on how manually release the components into the Test Bed environment. Webinars were also held on the operation of Keycloak and the Jenkins pipeline. Finally, the last webinar is about the use of *Codyze* and its integration into the CI/CD pipeline.
- *APPENDIX C: Component Integration Rounds*, describes the workshops held to complete the first and the second releases of the MEDINA framework in the “dev” and “test” environments and the status of component integration achieved.
- *APPENDIX E: Generic Architectural Workflows*, describes the workflows in detail, going step-by-step through the interactions between architectural components and the generic role(s) being involved.
- *APPENDIX F: Published APIs*, describes the REST API exposed by the MEDINA components, including a section dedicated to each of them.
- *APPENDIX G: User Manuals* contains the user manuals of all the MEDINA component that have a GUI.

1.3 Updates from D5.4

This deliverable evolves from D5.4 [2], so much of its content is common to that included in the previous document, with the ultimate goal of providing a self-contained deliverable that

facilitates the reader’s understanding. To simplify the tracking of progress and updates with respect to the previous version of the deliverable (D5.4), Table 1 shows a brief summary of the changes and additions made in each of the sections.

Table 1. Overview of deliverable updates with respect to D5.4

Section	Change
2	The integration methodology was completed, and the continuous deployment and validation strategy has been adopted during this third round. The point-to-point connections have been finalized and a new webinar for the MEDINA Codyze adoption has been released. Finally, the security CI/CD pipeline has been extended to include the MEDINA Codyze analysis, and a new Dashboard for tracking the status of the component released is now available.
3	This section has been updated by introducing two new workflows (WF8 and WF9).
4	This section contains an update of the technical description of the MEDINA components. An overview of the CCD component has been introduced.
5	The functional description chapter of the MEDINA IUI has been extended to report the component card and the status of the IUI requirements. The technical description now contains an overview of the microservices architecture pattern, and the new authorization strategy implemented. Moreover, the Integration of components section now describes the style rules agreed with all component owners to achieve a harmonized look and feel of the MEDINA framework. The Technical specifications section now also contains a description of the look and feel of the MEDINA IUI.
Appendix A	The hardware infrastructure used for the Test Bed environment is described here and remains unchanged from the previous release.
Appendix B	This appendix has been reorganized to host the webinars. Now, it includes the “Keycloak”, “Authorization and Filtering”, “CI/CD” and “Codyze” webinars.
Appendix C	This appendix has been reorganized to accommodate the first and second component integration rounds. The first round was described in the previous release. The second round has been moved to this appendix and its description is unchanged from the previous release.
Appendix D	This appendix contains the description of the pipeline schema put in place to support the implementation of the CI/CD solution in MEDINA.
Appendix E	This appendix revisits and updates the details related to the generic architectural workflows and introduces two new workflows (WF8 and WF9).
Appendix F	This appendix describes the REST API exposed by each component.
Appendix G	This appendix contains the User Manual for each component.

2 MEDINA Test Bed and Secure DevOps Infrastructure

This section presents the current status of the Test Bed environment and the hardware infrastructure used for its installation, which are not changed from the previous deliverable D5.4 [2].

It also describes the methodology followed to achieve the third release of the MEDINA framework, giving details on the new webinars held to help partners during this process and the situation of the status of component integration and point-to-point connections in month 33. This third round, in particular, focused on finalizing component functionalities based on feedback from the WP6 validation, and on improving the graphical user interfaces of all components.

2.1 Test Bed environment

The Test Bed environment is the environment in which the MEDINA framework is delivered to test and verify all functionalities. It is unchanged from the previous description in D5.4 [2], but for completeness of understanding we report here its configuration.

As described in *APPENDIX A: Operating Environment*, the Test Bed environment was installed and configured from scratch and consists of a three nodes Kubernetes [5] cluster with two different, independent and isolated virtual environments:

- **Development:** is used by developers for testing their modules without fear of bugs or errors. This environment does not affect the end users and is used to improve the code of the MEDINA micro-services before deploying them to the Test environment.
- **Test:** the main purpose here is to ensure that all the updates made on the different modules work as expected. This environment, which is more stable than the development environment, is used by developers for integration testing and by Use Case owners for the validation and quality assessment of the MEDINA components.

All the micro-services in the Test Bed environment are containerized and communicate with each other via a RESTful API over a secure HTTPS protocol.

The hardware equipment to setup the Kubernetes environments is described in the *APPENDIX A: Operating Environment - Hardware Infrastructure*.

Since the two MEDINA Use Cases, Bosch and Fabasoft, are validating the components released in the “Test” environment, they are also hosting a “Validation” environment. During the validation activities, feedback has been regularly reported in WP5 meetings and updates and improvements have been regularly introduced accordingly.

2.1.1 Component Integration Methodology

Once the Test Bed environment has been properly configured and all the necessary installations have been performed, the next step is to deploy all the component in the cluster and make the framework stable and reachable by external people.

This section recalls the methodology adopted throughout the project to perform the integration of components and focuses on the progress achieved during the third round. In particular, the third version of the MEDINA integrated solution has been released and continuous updates have been made to meet the feedback from the Use Cases.

In order to better organize the integration work, we adopted the following methodology which presents the actions to be taken until the complete release of the MEDINA framework:

1. Each component must be available on the internal private GitLab repository
2. Each component must be containerized into a docker image, the docker image must be available on the internal private docker registry Artifactory
3. Deployment of each component into the Development environment in the MEDINA Kubernetes cluster named “dev”
4. Standalone tests to check each component has been correctly deployed in the Development environment
5. Point to point tests for the communication in pairs of the components in the Development environment
6. Point to point tests in the Development environment verifying that the workflows described in section 3 have been correctly implemented
7. Deployment of the stable version of each component in the Test environment in the MEDINA Kubernetes cluster named “test”
8. Standalone tests to check each component has been correctly deployed in the Test environment
9. Point to point tests for the communication in pairs of the components in the Test environment
10. Point to point tests in the Test environment verifying that the workflows described in section 3 have been correctly implemented.

This methodology has been implemented through two instruments: workshops and webinars. The overall integration consists of three rounds at M15, M27 and M33. The webinars were recorded and shared with all partners in the Fabasoft cloud, in a folder named “TECHNICAL WEBINARS” (see Figure 1). This has allowed partners to re-watch them whenever needed.

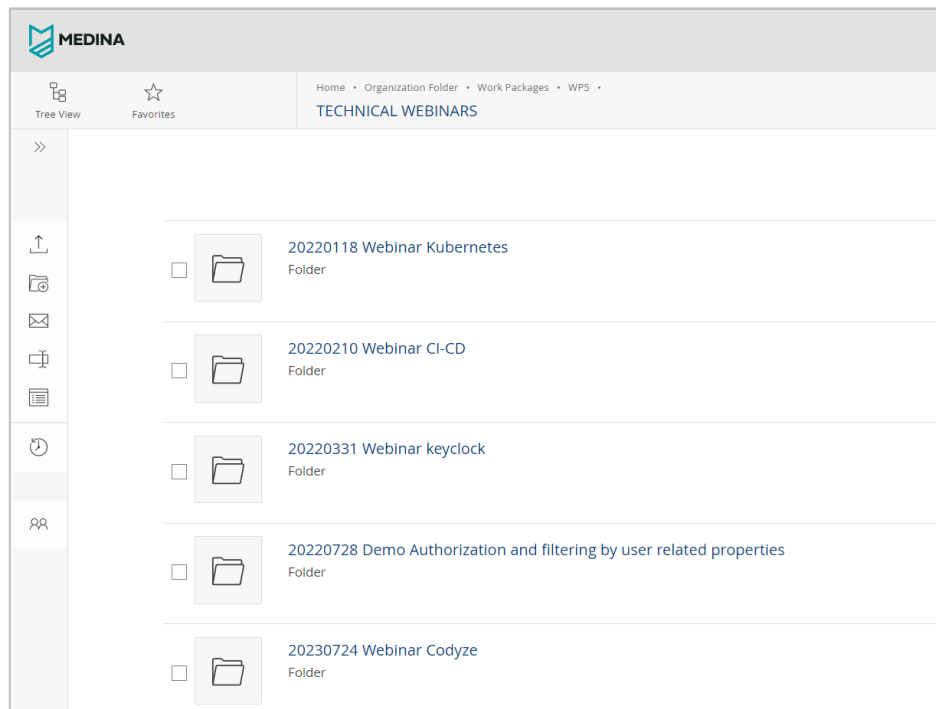


Figure 1. Technical Webinars

During the first integration round (M15), which is described in Appendix C, *First Round - First integration workshop*, HPE coordinated the integration of components, which was carried out manually by each partner. To support this, a webinar and a workshop were organised. During the webinar the main concepts and functionalities of Docker and Kubernetes were illustrated (see *Docker and Kubernetes Webinar with Sample Component Integration example* in APPENDIX

B: Webinars. During the workshop the partners were supported by T5.3 in the implementation of the first five actions of the methodology described above: integration in GitLab, build and push of the docker images into Artifactory, and deployment and tests in the Development environment of the MEDINA Kubernetes cluster.

During the second round (M27), which is described in Appendix C, *Second Round – Continuous Integration*, the Continuous Integration and Continuous Delivery (CI/CD) approach was finalized and completely adopted thanks to the implementation of the pipelines. The first stable version of the MEDINA framework was released and most of the point-to-point connections were implemented. Two webinars were delivered to illustrate to the partners the authorization and authentication concepts (see *Authorization and Filtering Webinar* in the Appendix B) and how to configure them using Keycloak (see *Keycloak Webinar* in the Appendix B), and a third webinar was dedicated to the implementation of the CI/CD pipelines (see *CI/CD Webinar* in Appendix B). The release of the first version of the MEDINA framework allowed the Use Cases to perform their validation of the workflows described in section 3. During this round, the last four actions foreseen by the defined methodology were successfully completed by all partners.

This final round (M33) focused on the continuous integration and changes of the components based on the feedbacks and requirements of the Use Cases. T5.3 collaborated with Work Package 6 and monitored all the component activities to help partners to achieve the final status of the applications. The continuous integration of component updates will continue until the end of the MEDINA project to further improve the framework. The other focus during this round was the improvement of the MEDINA framework graphical user interface, so that the user has a pleasant experience and enjoys the new look and feel of the platform.

2.1.1.1 Third Round – Integration of the validation feedback

During the second integration round all the components involved in the MEDINA framework were successfully implemented and released into both “dev” and “test” Kubernetes environments. The partners configured their pipelines, and the continuous integration and deployment strategy was full adopted.

During the third integration round all the point-to-point connections were established and tested. The final result is reported in Table 2 as follows:

- Light green: the connection was implemented during the first or second rounds
- Dark green: the connection has been successfully implemented during the third round
- Grey: the connection is no longer needed

Comparing the contents of Table 2 with the previous status shown in Table 38 (see *Second Round – Continuous Integration* in Appendix C), it can be seen that: one more connection has been introduced between the *MEDINA Integrated User Interface* and the *MEDINA Evidence Trustworthiness System* component, and all connections that were in progress have been completed.

Table 2. Status of point-to-point connections during the third round

Component Name A	Component Name B	Status
Orchestrator	Continuous Certification Evaluation	CONNECTED
Orchestrator	Trustworthiness System	CONNECTED
Orchestrator	Security Assessment	CONNECTED
Orchestrator	Catalogue of Controls & Metrics	CONNECTED
Orchestrator	NL2CNL Translator	CONNECTED

Component Name A	Component Name B	Status
Codyze	Orchestrator	CONNECTED
Cloud Evidence Collector	Security Assessment	CONNECTED
Security Assessment	Evidence Collection from VAT	CONNECTED
Security Assessment	Evidence Collection from Wazuh	CONNECTED
DSL Mapper	Orchestrator	CONNECTED
DSL Mapper	Catalogue of Controls and Metrics	DISCARDED
NL2CNL Translator	Catalogue of Controls and Metrics	CONNECTED
NL2CNL Translator	CNL Editor	CONNECTED
CNL Editor	DSL Mapper	CONNECTED
CNL Editor	Catalogue of Controls and Metrics	DISCARDED
AMOE	Catalogue of Controls and Metrics	CONNECTED
AMOE	Orchestrator	CONNECTED
Catalogue of Controls and Metrics	Static Risk Assessment and Optimisation Framework	CONNECTED
Continuous Certification Evaluation	Catalogue of Controls and Metrics	CONNECTED
Continuous Certification Evaluation	Dynamic Risk Assessment and Optimisation Framework	CONNECTED
Continuous Certification Evaluation	Life Cycle Manager	CONNECTED
Dynamic Risk Assessment and Optimisation Framework	Life Cycle Manager	CONNECTED
AMOE	Orchestrator	CONNECTED
SSI Framework	Life Cycle Manager	CONNECTED
Integrated UI	Catalogue of Controls and Metrics	CONNECTED
Integrated UI	Orchestrator	CONNECTED
Integrated UI	CNL Editor	CONNECTED
Integrated UI	Static Risk Assessment and Optimization Framework	CONNECTED
Integrated UI	Continuous Certification Evaluation	CONNECTED
Integrated UI	AMOE	CONNECTED
Integrated UI	NL2CNL Translator	DISCARDED
Integrated UI	SSI Framework	CONNECTED
Integrated UI	MEDINA Evidence Trustworthiness System	CONNECTED

Once all the point-to-point connections were established and the whole framework was available in both “dev” and “test” environments, the next steps were to finalize the components functionalities and validate the whole functions.

These steps were monitored during the bi-weekly WP5 meetings using a dedicated spreadsheet: all the component activities were reviewed there, and their advancements were discussed with the partners involved. When a new functionality was implemented, the Use Cases validated it and produce feedbacks, stating if it was okay or if more improvements were needed.

The result of this final integration round is the successful integration of all the components and the MEDINA framework working as expected.

The continuous integration and the stabilization of the “dev” and “test” environments will continue until the end of the project.

2.2 Implementation of the CI/CD solution

This section provides updates on the status of the implementation of the CI/CD strategy supported by CI/CD tools in month 33. First, it provides an overview of the operating environment that involves all CI/CD components and the Kubernetes cluster and how they work together in our automated solution designed for MEDINA for software release, which has been achieved through the use of pipelines. Secondly, more details are provided on the four standardized pipelines and their stages, and how they are setup through the Jenkins Seed Job.

2.2.1 Operating Environment

This section describes the overview of the MEDINA Operating Environment proposed to support the CI/CD implementation.

The MEDINA framework is made up by the collaboration of multiple components developed by the partners and published over the Internet. Each component corresponds to one or more microservices and the code is stored in the TECNALIA GitLab version control system, which provides repositories both for private² and open-source³ projects.

All open-source projects are published in TECNALIA's public GitLab, organized with a folder per component where every microservice reports its license, as shown in Figure 2.

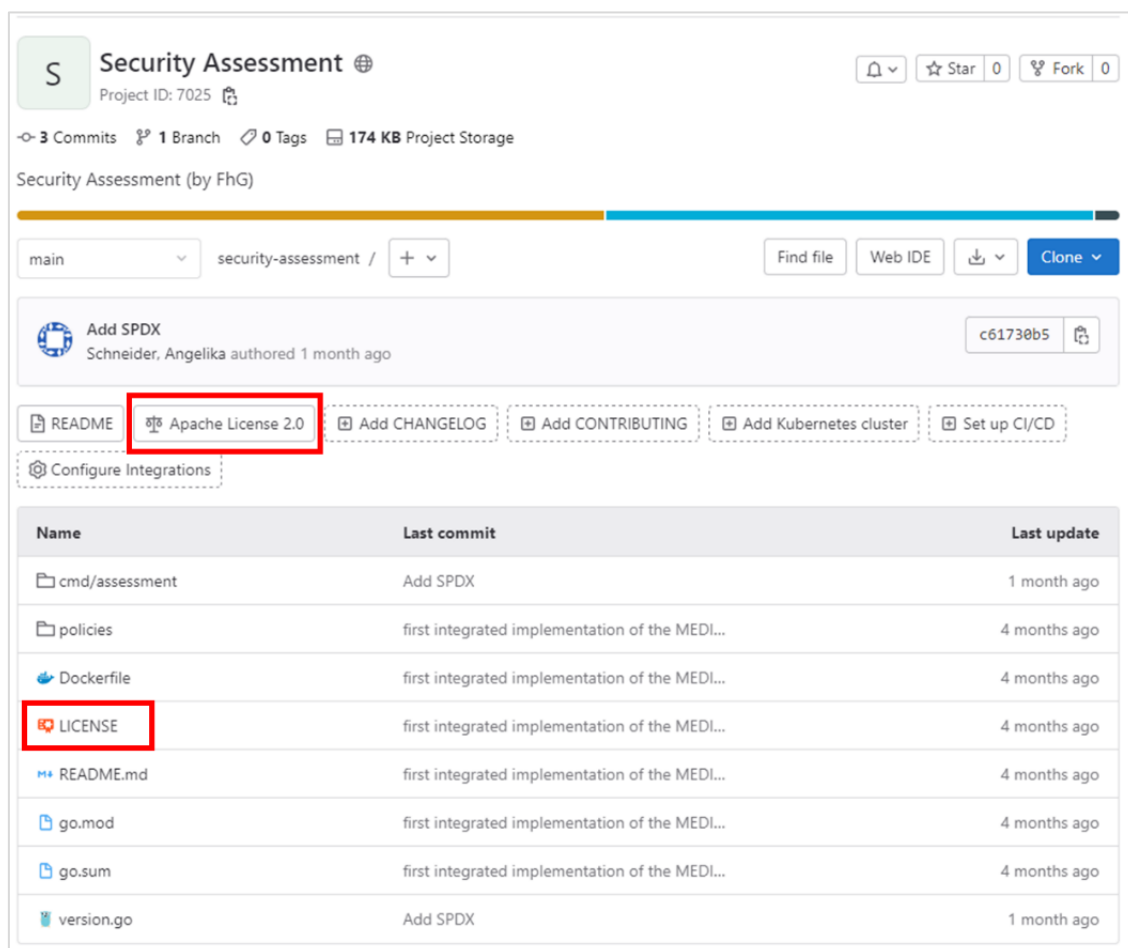


Figure 2. Public GitLab – license

² <https://git.code.tecnalia.com/medina> - [authentication required]

³ <https://git.code.tecnalia.com/medina/public>

In addition, the license is also provided using the SPDX [6] standard. Thus, in each source code file of the open-source projects there is a header indicating the licence details, which is for all components the Apache 2.0 license.

On the other hand, the TECNALIA's private GitLab repository is organized in folders that support work packages and tasks, so that each partner can use a dedicated path for its components. For example, the *CNL Editor* component belongs to the work package 2, Task 2.4 and that is the folder where it is stored, as shown in Figure 3.

During the regular WP5 meetings, it was coordinated and checked that all the components followed the conventions explained above.

W	WP5	Maintainer	MEDINA Framework Integration (HPE)	3	0	2
T	Task_5.3		System Continuous Integration and Optimization (HPE)	0	5	1
T	Task_5.2		Framework CI/CD strategy definition (Leader: HPE)	0	1	1
T	Task_5.1		Requirements, architecture and Infrastructure Specifications (Leader: Tecnalia)	0	0	1
W	WP4		Assessment Methods and Life-Cycle of Continuous Cloud Security Certification (FhG)	4	0	3
T	Task_4.4		Risk-Based Assessment and Security Controls Reconfiguration (Leader: CNR)	0	0	1
T	Task_4.3		Automation of the Cloud Security Certification Life-Cycle (Leader: FhG))	0	1	1
T	Task_4.2		Establishment of a digital audit trail for Cloud Security Certification (Leader: TECNALIA)	0	0	1
T	Task_4.1		Task 4.1 Continuous Evaluation of Cloud Security Certification (Leader: FhG)	0	2	1
W	WP3		Tools to gather evidences for high-assurance cybersecurity certification (TEC)	5	0	1
T	Task_3.5		Managing the trustworthiness of evidence with blockchain and DLT (Leader: TECNALIA)	0	1	2
T	Task_3.4		"Assessment" (Collecting evidences) of organizational measures using Natural Language Processing (Leader: H...)	0	1	2
T	Task_3.3		Analysis of information and data flows in Cloud applications (Leader: FhG)	0	2	1
T	Task_3.2		Continuous "Assessment" of security performance configuration of Cloud workloads (; Leader: XLAB)	0	6	1
T	Task_3.1		Collecting trustworthy evidence to support Cloud Service Certification (Leader: TECNALIA)	0	1	1
W	WP2		Certification Metrics and Specification Languages (CNR)	6	0	4
T	Task_2.6		Risk-based techniques for Certification Assurance Levels (Leader: CNR)	1	1	1
T	Task_2.5		Domain Specific Language Mapper (Leader: CNR)	0	1	1
T	Task_2.4		Controlled Natural Language Editor (Leader: HPE)	1	0	1
T	Task_2.3		Language Specification for Cloud Security Certification (Leader: CNR)	0	1	1
T	Task_2.2		Security Metrics for Continuous Cloud Certification (Leader: TECNALIA)	0	3	1
T	Task_2.1		Elicitation of Security Controls (Leader: TECNALIA)	0	0	1

Figure 3. Private GitLab repository

A microservice has to be containerized into a Docker image in order to be deployed. For this reason, a private Docker registry hosted at TECNALIA, the Jfrog Artifactory⁴ [7], was provided to store the Docker images.

Finally, the Docker images are deployed to the Kubernetes cluster and exposed over the Internet. The Jenkins automation server handles the delivery of each microservices: it fetches the code from GitLab, builds and stores the Docker image and finally releases it into the Kubernetes cluster (see Figure 4).

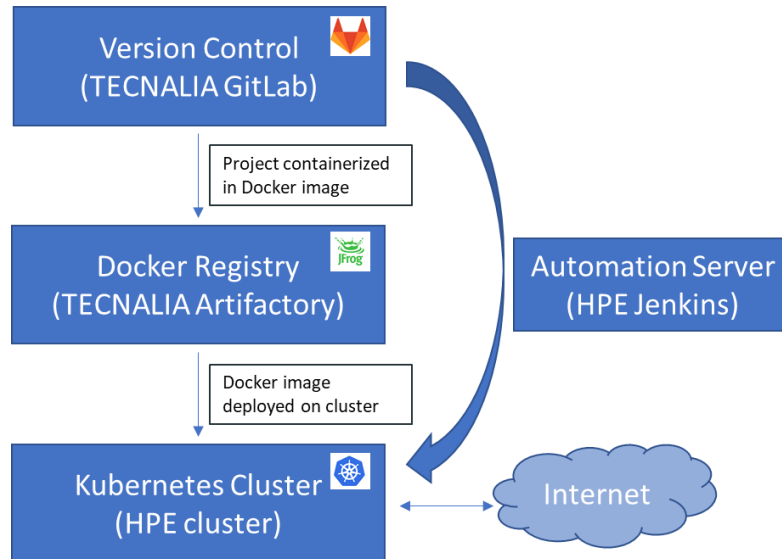


Figure 4. CI/CD tools

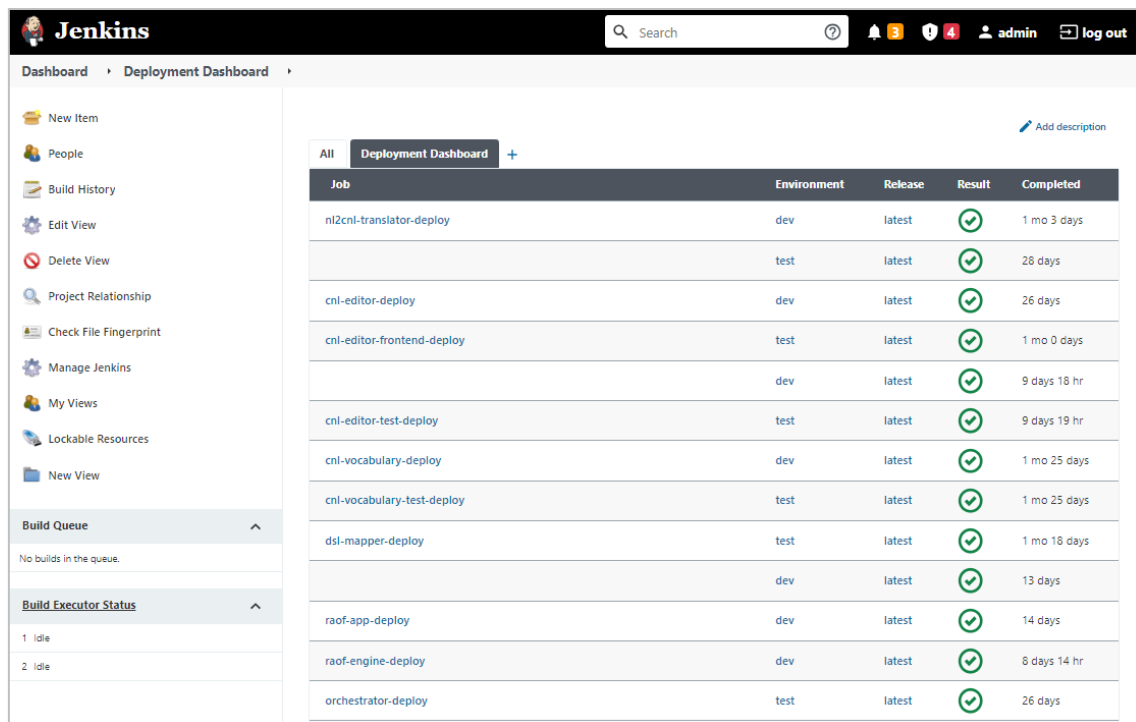
More details about the Jenkins pipelines are explained in the following section.

2.2.2 Pipelines

The implementation of the CI/CD solution that is put in place for supporting the MEDINA framework is based on the pipeline schema (see *APPENDIX D: Pipelines*). As a result, all the components that build the MEDINA framework have been deployed using these Jenkins pipelines and have been released in the two Kubernetes environment “dev” and “test”.

During the third integration round, a Deployment Dashboard has been introduced inside Jenkins (see Figure 5) to optimize the use of the CI/CD implemented solution.

⁴ <https://artifact.tecnalia.com/ui> - [authentication required]



The screenshot shows the Jenkins Deployment Dashboard. On the left is a sidebar with navigation links: New Item, People, Build History, Edit View, Delete View, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views, Lockable Resources, and New View. Below these are sections for 'Build Queue' (showing no builds) and 'Build Executor Status' (showing 1 idle and 2 busy executors). The main area displays a table of deployment jobs.

Job	Environment	Release	Result	Completed
nl2cnl-translator-deploy	dev	latest	✓	1 mo 3 days
	test	latest	✓	28 days
cnl-editor-deploy	dev	latest	✓	26 days
cnl-editor-frontend-deploy	test	latest	✓	1 mo 0 days
	dev	latest	✓	9 days 18 hr
cnl-editor-test-deploy	test	latest	✓	9 days 19 hr
cnl-vocabulary-deploy	dev	latest	✓	1 mo 25 days
cnl-vocabulary-test-deploy	test	latest	✓	1 mo 25 days
dsi-mapper-deploy	test	latest	✓	1 mo 18 days
	dev	latest	✓	13 days
raof-app-deploy	dev	latest	✓	14 days
raof-engine-deploy	dev	latest	✓	8 days 14 hr
orchestrator-deploy	test	latest	✓	26 days

Figure 5. Jenkins Deployment Dashboard

This Dashboard is an instrument to keep trace of the releases of the MEDINA components in both “dev” and “test” environments. Its main advantage is that it can be used to check if an error occurs, based on the result of the deploy pipeline, and easily verify which components are involved. This useful tool has been designed to meet the project needs and the use of the Jenkins pipelines made by the partners.

As a final step in the Security pipeline, the MEDINA component *Codyze* [8] has been added. *Codyze* is a static code analysis tool developed by FhG partner (see Section 4.7.1.5). In particular, Figure 6 illustrates the new *Codyze* step in the security pipeline.

Stage View

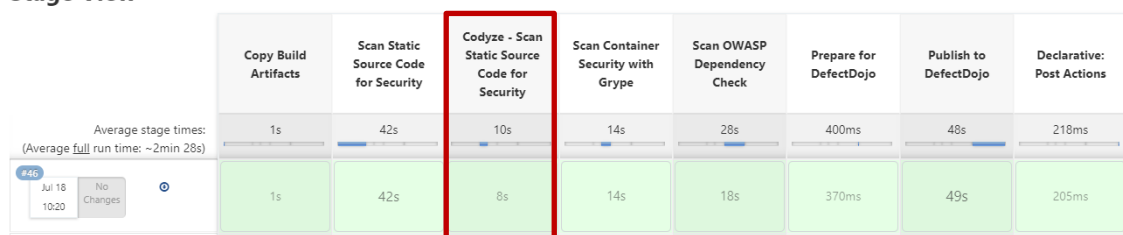


Figure 6. Codyze security pipeline step

Each MEDINA component is evaluated against the rulesets of *Codyze*. Reports are generated as SARIF and automatically processed by DefectDojo⁵. Results can be reviewed in DefectDojo WebUI, and the overall report is attached as artefact to the security pipeline in Jenkins. Figure 7 shows an example of *Codyze* application. This example checks that the TLS Cipher version is set to 1.2; instead in the code is used the version 1.0. This creates a compliance violation reported by DefectDojo as “Critical”.

⁵ <https://www.defectdojo.org/>

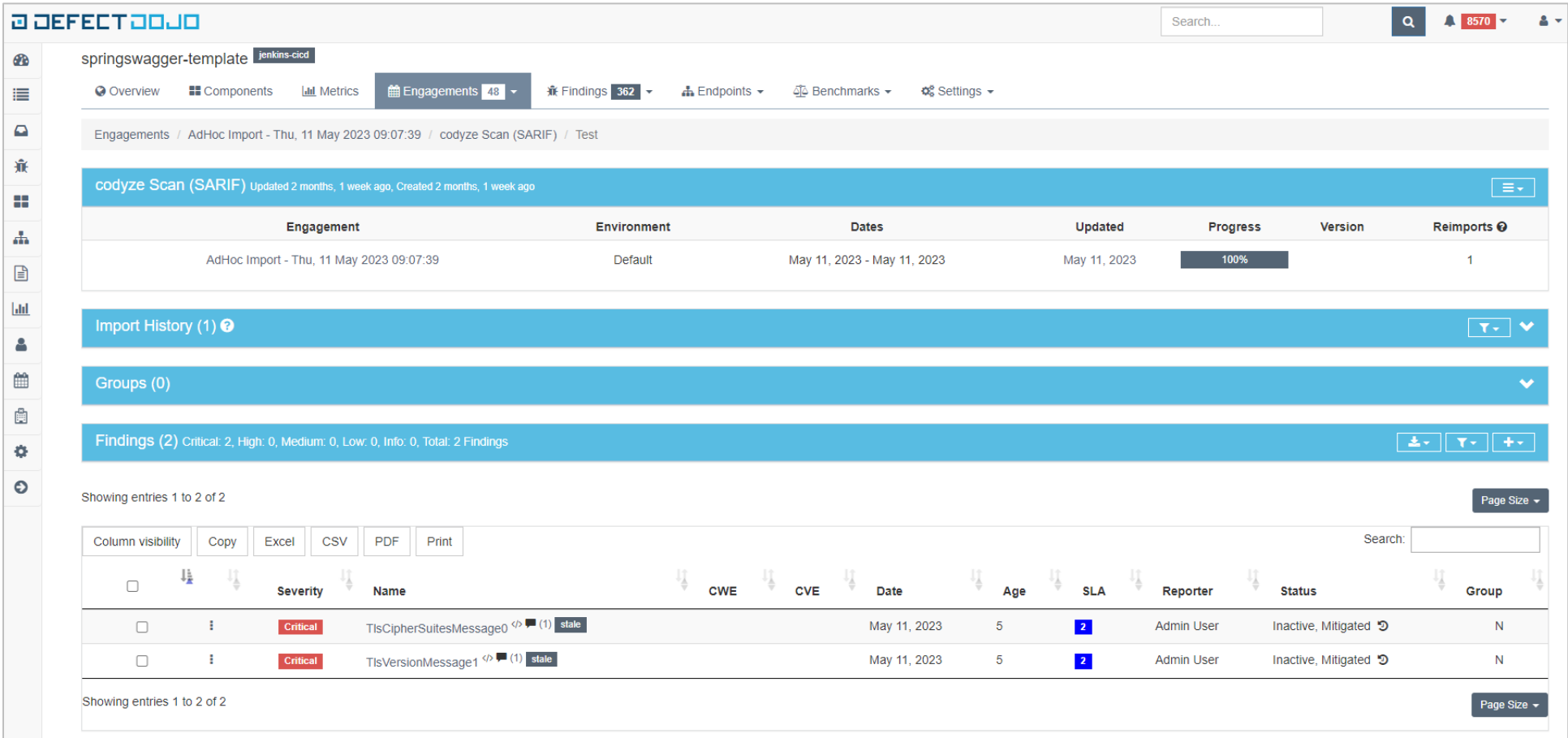


Figure 7. DefectDojo sample Codyze results

3 Generic Architectural Workflows

This section provides the final version of the generic MEDINA workflows (WFs), which were first introduced in D5.3 [1] and further detailed in D6.3 [9] and D5.4 [2]. Furthermore, we present the final “Authorization Concept” comprising the definition of roles and access-levels assigned to the different UI components of the developed framework. The presented concept has been implemented in the validation use cases as documented in D6.4 [10].

For interested readers, the full description of the MEDINA workflows can be found in *APPENDIX E: Generic Architectural Workflows*.

3.1 Generic MEDINA Workflows

This section provides as background the generic workflows which comprise the MEDINA framework, consisting of the different scenarios/interactions shown in Table 3. Please notice that at the time of writing the generic workflows have been updated with those related to the Auditor (CAB, NCCA) namely “Verifiable Credentials for Certificates” (WF8), and “Integrity Verification” (WF9).

Table 3. Generic MEDINA workflows

Workflow	Comment	Other/Dependency
WF1 – Preparation of Target of Certification (ToC)	Setup, configure and deploy the cloud service to certify (ToC) on top of the chosen hyperscaler(s). This process includes configuring the underlying PaaS/IaaS.	Mandatory workflow CSP Responsibility Dependencies: None
WF2 – Preparation of MEDINA components	Setup, configure and deploy the MEDINA components. Only related to those components under the responsibility of the CSP.	Mandatory workflow CSP Responsibility Dependencies: WF1
WF3 – EUCS deployment on ToC	Setup, configure and deploy the corresponding EUCS framework (for the chosen assurance level basic/substantial/high) on the ToC.	Mandatory workflow CSP Responsibility Dependencies: WF1, WF2
WF4 – EUCS Preparedness – ToC Self-Assessment	Self-assess preparedness for EUCS certification based on the chosen assurance level. This is a risk-based approach.	Optional workflow CSP Responsibility Dependencies: WF1, WF2, WF3
WF5 – EUCS – compliance assessment	Performs a point-in-time (discrete) EUCS compliance assessment for the ToC. When such discrete assessment is periodically executed, then we achieve the MEDINA notion of “continuous”.	Mandatory workflow CSP/ CAB Responsibility Dependencies: WF1, WF2, WF3
WF6 – EUCS – maintenance of ToC certificate	Start certificate maintenance life-cycle for the ToC. Based on current EUCS, the maintenance process comprises the following stages: (issuance ⁶), renewal, continuation, update, re-issuance (new certificate), withdrawal, and suspension.	Mandatory workflow Dependencies: WF1, WF2, WF3, WF5
WF7 – EUCS – report on ToC certificate	Reports on EUCS certificate status for a ToC. The report can be obtained by the CSP, in which case the level of provided details might	Optional workflow CSP Responsibility Dependencies: WF1, WF2,

⁶ Despite the initial issuance of certificate is not mentioned in the maintenance process defined by the core EUCS document, for MEDINA purposes this discussion is part of the *Life-cycle Manager* (WP4).

Workflow	Comment	Other/Dependency
	vary.	WF3, WF5
WF8 – Auditor-Verifiable Credentials for Certificates (NEW)	The SSI component is used to issue verifiable credentials to the CSPs and generate verifiable proofs to the cloud customers.	Optional workflow CAB / CSP Responsibility Dependencies: WF1, WF2, WF3, WF5
WF9 – Auditor Integrity Verification (NEW)	Using the MEDINA Evidence Trustworthiness System, the CAB verifies the integrity of the collected evidence.	Optional workflow CAB Responsibility Dependencies: WF1, WF2, WF3, WF5

Based on these generic workflows, the rest of this section focuses on presenting the roles and the authorization concept that has been defined for the MEDINA framework.

3.2 Roles

To present the authorization concept (see section 3.3), first the basic roles in MEDINA (cf. D6.3 [9]) are introduced. Table 4. presents each one of the generic roles associated with the MEDINA framework, along with the access level they have to the Cloud Service Provider's data. The level of access ranges from "Cloud Service Provider" (all cloud services offered) to an individual "Cloud Service" (a specific set of cloud resources). Please note that the EUCS [11] defines its target of certification at Cloud Service level.

Table 4. MEDINA Roles and Levels of Visibility

Roles	Explanation (cf. D6.3 [9])	Level of Access
IT Security Governance	Its main objective is the protection of Bosch business models, products, services, and data.	Cloud Service Provider ⁷
Security Analyst	Responsible for ensuring that the Bosch Group's digital assets and sensitive information are protected as well as evaluating and reporting on the efficiency of the security policies in place.	Cloud Service Provider
Domain Governance	Acts as the core competence holder and responsible topic owner for product security.	One or more Cloud Services
Product and Service Owner	The Product & Service Owner is the central point of contact for all questions concerning a specific Bosch IT product or service.	Cloud Service ⁸
Product (Security) Engineer	Oversees the build, deploy, and run of a product and its system components.	Cloud Service
Chief Information Security Office (CISO)	The Chief Information Security Officer (CISO) is who the Compliance Manager has to report to.	Cloud Service Provider
Customer	The customer ⁹ is either a company consuming cloud products or services (B2B, business-to-business context), or an individual (B2C, business-to-customer context).	Cloud Service
Auditor¹⁰	The Conformity Assessment Body (CAB) is a body that performs conformity assessment services with the goal of demonstrating that specified requirements are fulfilled.	One or more Cloud Services

⁷ Including all underlying certifiable Cloud Services.

⁸ For the purposes of MEDINA, we consider access to at most one Cloud Service.

⁹ For the purposes of MEDINA, the Customer is the only non-authenticated role in the framework.

¹⁰ This role refers to internal Auditors and NCCAs (National Cybersecurity Certification Authority).

For example, in Table 4. one can observe that the CISO role has visibility to all cloud services from its organization (i.e., cloud service provider), whereas the more technical role “Product (Security) Engineer” can only access information related to a specific cloud service under its responsibility.

Each defined role is mapped to a local user in the MEDINA Integrated UI for the purposes of framework validation (WP6). Furthermore, Cloud Services are created and related to those users also to validate the designed filtering concept. Table 5 shows this mapping.

Table 5. Mapping local MEDINA users and Cloud Services

Local Username	Cloud Service ID Token
UC1_SecGov	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
UC1_SecAnalyst	dc1eff0d-4afd-4e2f-8b62-13458c56a540
UC1_CISO	f7c1e5c3-511e-45d1-bf7c-867cdc6a8db9
UC1_DomGov	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
UC1_Auditor	dc1eff0d-4afd-4e2f-8b62-13458c56a540
UC1_ProdOwn	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
UC1_ProdSec	
Non-Authenticated User	N/A

As a rule of thumb each defined user in MEDINA can only access the data related to its own CloudServiceID¹¹ token.

Next, for each defined role the set of allowed actions (authorization concept) is presented based on both the relevant WFs presented in the previous section, and the involved MEDINA framework components.

3.3 Authorization Model

MEDINA leverages the Role Based Access Control model (RBAC¹²) to enforce specific permissions on the Integrated UI for certain components. This section presents the final version of MEDINA’s RBAC concept based on the generic workflows, whereas details associated to its technical implementation are presented later on this document in *APPENDIX E: Generic Architectural Workflows*.

3.3.1 WF1 – Preparation of Target of Certification (ToC)

This initial workflow, despite not invoking any of the MEDINA components, is an evident pre-requisite for the CSP to fulfil before the certification process starts (see Table 6).

Its main goal is for the CSP to prepare the Target of Certification (ToC), both from a technical (e.g., deploying the actual cloud service in the hyperscaler) and organizational (e.g., gather the operational manuals in electronic format) perspectives.

¹¹ This token represents a unique Cloud Service ID on the validation testbed. It can map to any of the IaaS, PaaS or SaaS cloud services created for this purpose.

¹² Please refer to https://en.wikipedia.org/wiki/Role-based_access_control

Table 6. Workflow 1

Short Explanation	Associated MEDINA Components
Setup, configure and deploy the cloud service to certify (ToC) on top of the chosen hyperscaler(s). This process includes configuring the underlying PaaS/IaaS.	CSP testbed

For this initial workflow, the only role allowed to operate on the platform is the so-called Product (Security) Engineer, as shown in Table 7.

Table 7. RBAC Model for Workflow 1

Roles	Component	UI Actions
IT Security Governance	Testbed	None
Security Analyst	Testbed	None
Domain Governance	Testbed	None
Product and Service Owner	Testbed	None
Product (Security) Engineer	Testbed	Setup, configure, deploy
Chief Information Security Office (CISO)	Testbed	None
Customer	Testbed	None
Auditor	Testbed	None

3.3.2 WF2 – Preparation of MEDINA Components

The second generic workflow of the architecture (WF2) refers to the actual configuration and deployment of those MEDINA components which are needed for certifying the Cloud Service (see Table 8). This WF2 does not perform any actual assessment but executes a required set of deployment actions before WF3 triggers the certification process.

Table 8. Workflow 2

Short Explanation	Associated MEDINA Components
Setup, configure and deploy the MEDINA components. Only related to those components under the responsibility of the CSP.	Evidence Collectors, Integrated UI, Catalogue of Controls and Metrics

The evidence collectors (e.g., *Clouditor* and *Wazuh*), along with the *Integrated UI* are deployed and configured by the Product (Security) Engineer exclusively, as shown in Table 9.

Table 9. RBAC Model for Workflow 2

Roles	Component	UI Actions
IT Security Governance	Testbed	None
Security Analyst	Testbed	None
Domain Governance	Testbed	None
Product and Service Owner	Testbed	None
Product (Security) Engineer	Testbed	Setup, configure, deploy (Catalogue, SSO, Clouditor, Wazuh, Codyze, VAT)
Chief Information Security Office (CISO)	Testbed	None
Customer	Testbed	None
Auditor	Testbed	None

It must be noted that standards and their associated data as contained in the *Catalogue* cannot be modified by any of the MEDINA roles (with exception of its Description which can be changed by the IT Security Governance), as shown in Table 10.

Information from the *Catalogue* is pre-filled by the “MEDINA framework provider” with the EUCS standard. Also, the Customer role (non-authenticated user) is denied any access to the actual Integrated UI for the interactions taking place in WF2.

Table 10. RBAC Model for Catalogue UI in Workflow 2

Role	Component	UI Actions
IT Security Governance	Catalogue UI	Update Description ¹³ Read all Entities
Security Analyst	Catalogue UI	Read all Entities
Domain Governance	Catalogue UI	Read all Entities
Product and Service Owner	Catalogue UI	Read all Entities
Product (Security) Engineer	Catalogue UI	Read all Entities
Chief Information Security Office (CISO)	Catalogue UI	Read all Entities
Customer	Catalogue UI	None
Auditor	Catalogue UI	Read all Entities

3.3.3 WF3 – EUCS deployment on ToC

After the ToC has been deployed on the hyperscaler (WF1) and the corresponding MEDINA components have been configured/deployed by the CSP (WF2), then it is possible to use the later for certifying the Cloud Service. That is the goal of this WF3 shown in Table 11.

Table 11. Workflow 3

Short Explanation	Associated MEDINA Components
Setup, configure and deploy the corresponding EUCS framework (for the chosen assurance level basic/substantial/high) on the ToC.	Orchestrator, CNL Editor, AMOE

With the exception of *AMOE*, in this WF3 read-only roles without permission to modify information related to the components have been identified, and once again the Product (Security) Engineer becomes the only role with write permissions. Furthermore, with exception of the latter role, all the others cannot access *AMOE* at all (see Table 14).

Table 12. RBAC Model for Orchestrator UI in Workflow 3

Role	Component	UI Actions
IT Security Governance	Orchestrator UI	Read (Requirements, Cloud Service)
Security Analyst	Orchestrator UI	Read (Requirements, Cloud Service)
Domain Governance	Orchestrator UI	Read (Requirements, Cloud Service)
Product and Service Owner	Orchestrator UI	Read (Requirements, Cloud Service)
Product (Security) Engineer	Orchestrator UI	Add/Remove Requirements, and Add/Remove Cloud Service to Orchestrator UI
Chief Information Security Office (CISO)	Orchestrator UI	Read (Requirements, Cloud Service)

¹³ Descriptions for all entities can be updated for all Entities within the *Catalogue* e.g., to match specific CSP requirements / internal regulations.

Role	Component	UI Actions
Customer	Orchestrator UI	None
Auditor	Orchestrator UI	Read (Requirements, Cloud Service)

Table 13. RBAC Model for CNL Editor UI in Workflow 3

Role	Component	UI Actions
IT Security Governance	CNL Editor UI	Read
Security Analyst	CNL Editor UI	Read
Domain Governance	CNL Editor UI	Read
Product and Service Owner	CNL Editor UI	Read
Product (Security) Engineer	CNL Editor UI	Show/Edit/Complete/Map to CNL Editor UI
Chief Information Security Office (CISO)	CNL Editor UI	Read
Customer	CNL Editor UI	None
Auditor	CNL Editor UI	Read

Table 14. RBAC Model for AMOE UI in Workflow 3

Role	Component	UI Actions
IT Security Governance	AMOE UI	None ¹⁴
Security Analyst	AMOE UI	None
Domain Governance	AMOE UI	None
Product and Service Owner	AMOE UI	None
Product (Security) Engineer	AMOE UI	Upload New File/Delete File to AMOE UI
Chief Information Security Office (CISO)	AMOE UI	None
Customer	AMOE UI	None
Auditor	AMOE UI	None

3.3.4 WF4 – EUCS Preparedness – ToC Self-Assessment

This workflow relates to the components in charge of performing the static risk management (SATRA) and the self-assessment questionnaires (*Catalogue of Controls and Metrics*) as documented by D2.8 [12] and D2.2 [13] respectively (see Table 15).

Although SATRA implements a “stand alone functionality”, which does not need to be technically deployed in the Cloud Service (cf. WF3), it is integrated into the whole MEDINA framework thanks to the Integrated UI.

Table 15. Workflow 4

Short Explanation	Associated MEDINA Components
Self-assess preparedness for EUCS certification based on the chosen assurance level following a risk-based approach.	SATRA, Catalogue Questionnaire

¹⁴ Not even access AMOE at all i.e., AMOE does not appear as menu option for these roles on the Integrated UI.

As shown in Table 16, only the Product and Service Owner can perform all actions available on the SATRA UI. All other roles (except the Customer) are assigned read-only/reporting actions according to the least privilege principle.

Table 16. RBAC Model for SATRA in Workflow 4

Role	Component	UI Actions
IT Security Governance	SATRA UI	Risk Computation (Reporting)
Security Analyst	SATRA UI	Risk Computation (Reporting)
Domain Governance	SATRA UI	Risk Computation (Reporting)
Product and Service Owner	SATRA UI	ToE Info, Questionnaire, Asset Information, Risk Computation (Reporting)
Product (Security) Engineer	SATRA UI	Risk Computation (Reporting)
Chief Information Security Office (CISO)	SATRA UI	Risk Computation (Reporting)
Customer	SATRA UI	None
Auditor	SATRA UI	Risk Computation (Reporting)

As shown in Table 17, only the Product and Service Owner can perform all actions available on the *Catalogue Questionnaire* UI. All other roles (except the Customer) can load existing questionnaires and can generate reports. Finally, the Auditor can only edit the “non-conformities” field.

Table 17. RBAC Model for Catalogue Questionnaire in Workflow 4

Role	Component	UI Actions
IT Security Governance	Catalogue Questionnaire UI	Load Questionnaire Generate Report
Security Analyst	Catalogue Questionnaire UI	Load Questionnaire Generate Report
Domain Governance	Catalogue Questionnaire UI	Load Questionnaire Generate Report
Product and Service Owner	Catalogue Questionnaire UI	Start a New Questionnaire ¹⁵ Load Questionnaire Generate Report Remove Questionnaire
Product (Security) Engineer	Catalogue Questionnaire UI	Load Questionnaire Generate Report
Chief Information Security Office (CISO)	Catalogue Questionnaire UI	Load Questionnaire Generate Report
Customer	Catalogue Questionnaire UI	None
Auditor	Catalogue Questionnaire UI	Load Questionnaire Edit non-conformities Generate Report

3.3.5 WF5 – EUCS Compliance Assessment

This WF5 describes **discrete compliance assessments** (see Table 18), which should then be periodically executed for the MEDINA framework to start the certification lifecycle (cf. WF6).

¹⁵ The Product and Service Owner can edit all questionnaire fields except non-conformities

Table 18. Workflow 5

Short Explanation	Associated MEDINA Components
Performs a point-in-time (discrete) EUCS compliance assessment for the ToC. When such discrete assessment is periodically executed, then we achieve the MEDINA notion of “continuous”.	AMOE ¹⁶ , Orchestrator ¹⁷

WF5 contains the interactions for performing discrete assessments, where only the role (internal/external) Auditor is allowed to change AMOE recommended assessments and submit them for evaluation to the Orchestrator (see Table 20). All additional roles, with exception of the non-authenticated Customer, can only perform “read” actions from the components’ UIs. The Customer role has no access to any of the UIs in this WF5.

Table 19. RBAC Model for Orchestrator in Workflow 5

Role	Component	UI Actions
IT Security Governance	Orchestrator UI	Read ¹⁸
Security Analyst	Orchestrator UI	Read
Domain Governance	Orchestrator UI	Read
Product and Service Owner	Orchestrator UI	Read
Product (Security) Engineer	Orchestrator UI	Read
Chief Information Security Office (CISO)	Orchestrator UI	Read
Customer	Orchestrator UI	None
Auditor	Orchestrator UI	Read

Table 20. RBAC Model for AMOE UI in Workflow 5

Role	Component	UI Actions
IT Security Governance	AMOE UI	Read
Security Analyst	AMOE UI	Read
Domain Governance	AMOE UI	Read
Product and Service Owner	AMOE UI	Read
Product (Security) Engineer	AMOE UI	Read
Chief Information Security Office (CISO)	AMOE UI	Read
Customer	AMOE UI	None
Auditor	AMOE UI	All ¹⁹

3.3.6 WF6 – EUCS – Maintenance of ToC certificate

This WF6 departs from the current definition of certificate maintenance in the EUCS core document [11], and for the purposes of MEDINA, it also adds an initial stage of “certificate issuance”.

¹⁶ AMOE UI for visualizing events and compliance statuses

¹⁷ Orchestrator UI menu option for visualizing the results of Security Assessments

¹⁸ Filtering assessment results in the Orchestrator is considered a “Read” action.

¹⁹ The Auditor role can change CAB assessments and sent those to Clouditor.

Despite WF6 plays an important role in MEDINA (i.e., continuous execution and analysis of discrete assessments), there is no user interaction envisioned within the Integrated UI. **For this reason, WF6 is not associated to any RBAC model.**

3.3.7 WF7 – EUCS –Report on ToC Certificate

The goal of this WF7 is to report about the status of an EUCS certificate corresponding to the ToC and at different levels of detail, depending on the targeted audience (CAB, CSP, etc.). As shown in Table 21, WP7 takes care of reporting the status of the certificate (and related evidence) to authorized stakeholders.

Table 21. Workflow 7

Short Explanation	Associated MEDINA Components
Report on EUCS certificate status for a ToC. The report can be obtained by the CAB or by the CSP, in which case the level of provided details might vary.	CCE, Orchestrator

In this case, the proposed RBAC model considers read-only actions for all roles associated to the components used to visualize the certificates and corresponding assessment results, as shown in Table 22 and Table 23.

Table 22. RBAC Model for CCE UI in Workflow 7

Role	Component	UI Actions
IT Security Governance	CCE UI	Read
Security Analyst	CCE UI	Read
Domain Governance	CCE UI	Read
Product and Service Owner	CCE UI	Read
Product (Security) Engineer	CCE UI	Read
Chief Information Security Office (CISO)	CCE UI	Read
Customer	CCE UI	None
Auditor	CCE UI	Read

Table 23. RBAC Model for Orchestrator UI in Workflow 7

Role	Component	UI Actions
IT Security Governance	Orchestrator UI	Read
Security Analyst	Orchestrator UI	Read
Domain Governance	Orchestrator UI	Read
Product and Service Owner	Orchestrator UI	Read
Product (Security) Engineer	Orchestrator UI	Read
Chief Information Security Office (CISO)	Orchestrator UI	Read
Customer	Orchestrator UI	Read ²⁰
Auditor	Orchestrator UI	Read

²⁰ For non-authenticated users (i.e., Customer role), the ACLM UI publishes the status of EUCS certificates on a publicly available repository.

3.3.8 WF8 – Auditor - Verifiable credentials for certificates (NEW)

The goal of this WF8 is to issue verifiable credentials to the CSPs and generate verifiable proofs to CSP customers (see Table 24). For this purpose, a Self-Sovereign Identity (SSI) Framework is considered. The *SSI Framework* provides CSPs with the capability to manage their own security certificates as part of their identity through verifiable credentials. “To manage their own identity” ultimately means that they store their identity on their own “user space” without intervention of a third-party.

Table 24. Workflow 8

Short Explanation	Associated MEDINA Components
Issue and verify credentials related to certificates.	Self-Sovereign Identity (SSI)

As shown in Table 25, the related authorization model only allows Auditors (internal, CAB/NCCA) to leverage the SSI UI for the issuance and verification of credentials.

Table 25. RBAC Model for SSI UI in Workflow 8

Role	Component	UI Actions
IT Security Governance	SSI UI	None
Security Analyst	SSI UI	None
Domain Governance	SSI UI	None
Product and Service Owner	SSI UI	None
Product (Security) Engineer	SSI UI	None
Chief Information Security Office (CISO)	SSI UI	None
Customer	SSI UI	None
Auditor	SSI UI	Issuance and verification of credentials

3.3.9 WF9 – Auditor - Integrity verification (NEW)

The goal of this WF9 is to validate the integrity of both evidence and assessment results (see Table 26). For this purpose, it is necessary to compare the information currently available on the *Orchestrator* with the information recorded on the *MEDINA Evidence Trustworthiness Management System*. As a result, integrity is verified.

Table 26. Workflow 9

Short Explanation	Associated MEDINA Components
Verify the integrity of evidence and assessment results.	Evidence Trustworthiness Management System (DLT)

As shown in Table 27, the proposed RBAC model considers the Auditor (internal or CAB/NCCA) is the only role able to perform verification operations.

Table 27. RBAC Model for DLT UI in Workflow 9

Role	Component	UI Actions
IT Security Governance	DLT UI	None
Security Analyst	DLT UI	None
Domain Governance	DLT UI	None
Product and Service Owner	DLT UI	None
Product (Security) Engineer	DLT UI	None

Role	Component	UI Actions
Chief Information Security Office (CISO)	DLT UI	None
Customer	DLT UI	None
Auditor	DLT UI	Verification of evidence and assessment results

3.4 Authorization Model for the MEDINA Integrated UI

The MEDINA *Integrated UI* itself also applies an authorization model for the defined roles in the framework as shown in Table 28. The Integrated UI detects the role of the logged user and adapts the user interface accordingly.

One of the adaptations refer to the tools the user has access to. For example, a non-authenticated Customer only sees the *Orchestrator* tool in the left menu (see Figure 8). While a Product and Service Owner can see several other tools: *Catalogue of Controls and Metrics*, *Orchestrator*, *Customization of Requirements*, *Risk Assessment*, *Organizational Evidence Assessment* and *Continuous Certificate Evaluation* (see Figure 9).

Another adaptation concerns the diagram displayed on the home page (see section 3.4.1). Here, the MEDINA IUI includes a different diagram for each user role, representing the MEDINA tool set (on the left) and the possible actions that the role can perform with them (in the centre). For example, Figure 9 shows the actions that the Product and Service Owner can carry out:

- Report assessments and certificates (*Orchestrator*)
- Report policy assessment (*Organizational Evidence Assessment*)
- Report non-compliances (*Continuous Certificate Evaluation*)
- Perform CS-Basic self-assessment (*Catalogue*)
- Perform static risk-assessment (*Risk Assessment*)

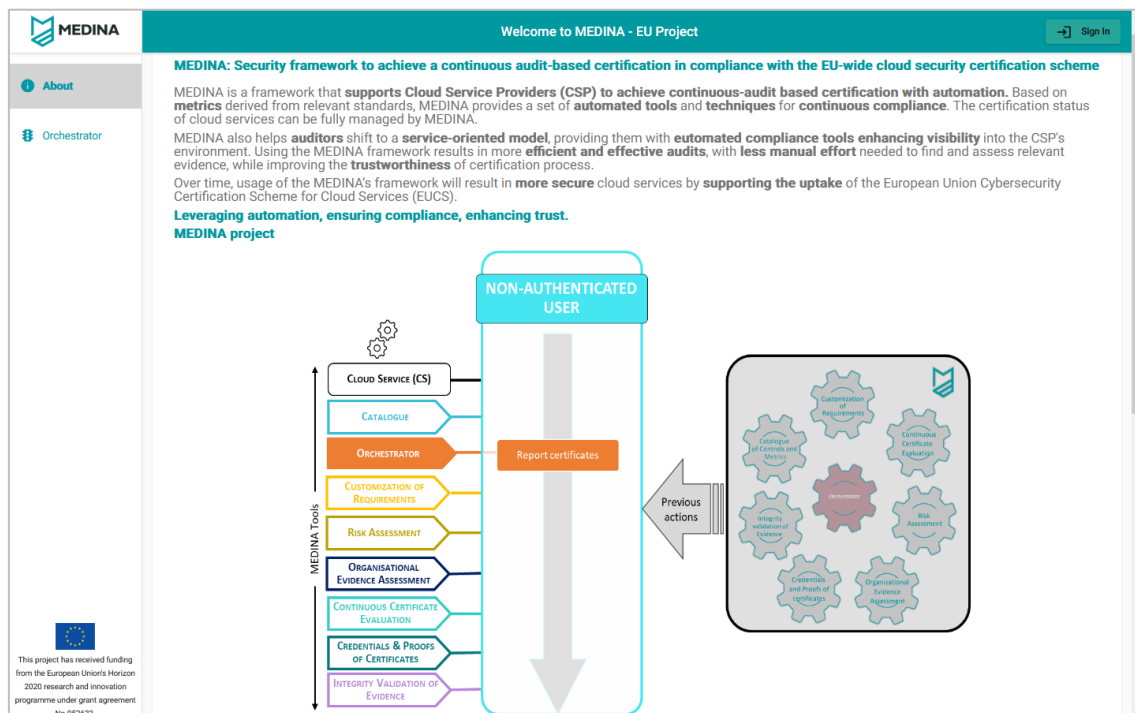


Figure 8. Home page for the non-authenticated customer

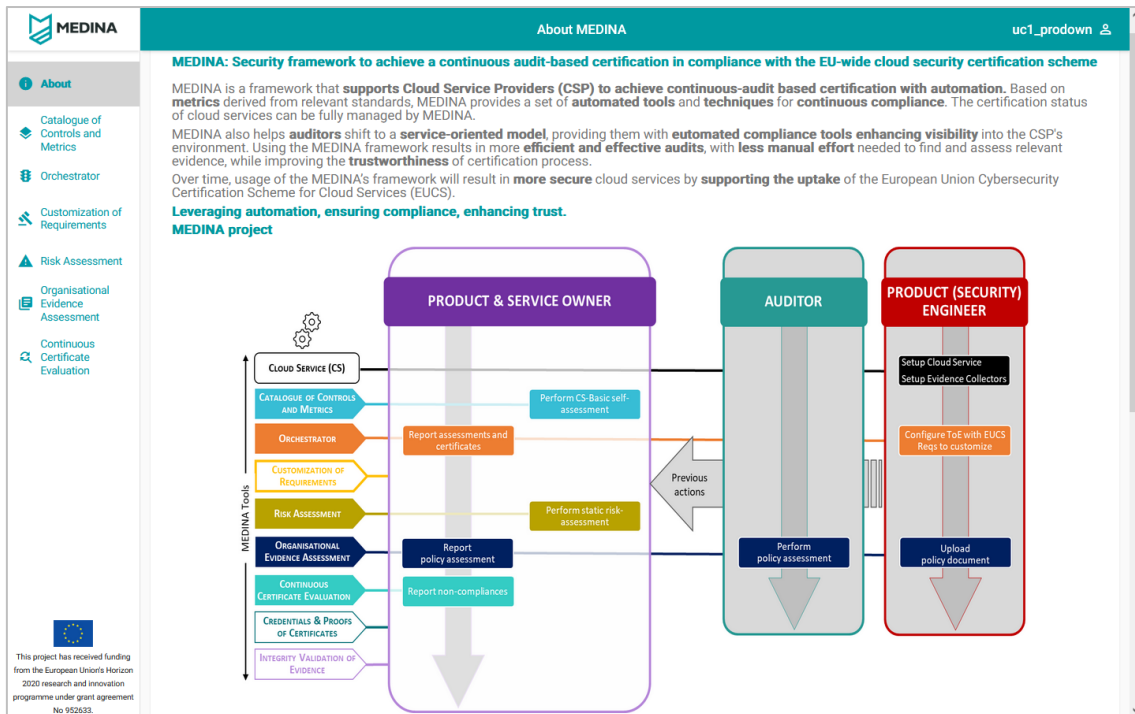


Figure 9. Home page for the Product and Service Owner user role

Table 28. RBAC Model for Integrated UI (applies to all WFs)

Role	Catalogue of Controls & Metrics	Orchestrator	Customization of Requirements	Continuous Certificate Evaluation	Risk Assessment	Organisational Evidence Assessment	Credentials & Proofs of Certificates	Integrity Validation of Evidence
IT Security Governance	X	X	X	X	X	X		
Security Analyst	X	X	X	X	X	X		
Domain Governance	X	X	X	X	X	X		
Product and Service Owner	X	X	X	X	X	X		
Product (Security) Engineer	X	X	X	X	X	X		
Chief Information Security Office (CISO)	X	X	X	X	X	X		
Customer		X ²¹						
Auditor	X	X	X	X	X	X	X	X

²¹ For non-authenticated users (i.e., Customer role), the Life-cycle Manager UI publishes the status of EUCS certificates on a publicly available repository.

3.4.1 Role diagrams

The MEDINA UII includes a different diagram for each user role (see Figure 10 to Figure 15). The purpose of these diagrams is to help the user to navigate in the MEDINA framework.

The figures represent the **MEDINA Tool set** (on the left), and the actions the user role is allowed to perform with them (in the centre). Each **role** is represented by a different colour in the box. Each tool is also represented by a different colour, which is propagated to the related actions. Some tools are not available to certain roles, and in these cases the tool has been depicted with no-background colour.

The **actions** are tied to the corresponding tool by the colour of the box and by a line that connects tool and action. The actions are ordered from top-to-down in the same workflow, which is represented by an arrow. Some actions are, however, independent of the workflow, and are represented in parallel, outside of the main workflow.

On the right side of the figure, a smaller size box with a grey background represents other actions that must have been executed previously by other roles in the MEDINA framework for the actual role to be able to perform the allowed actions (i.e., they are **pre-requirements**).

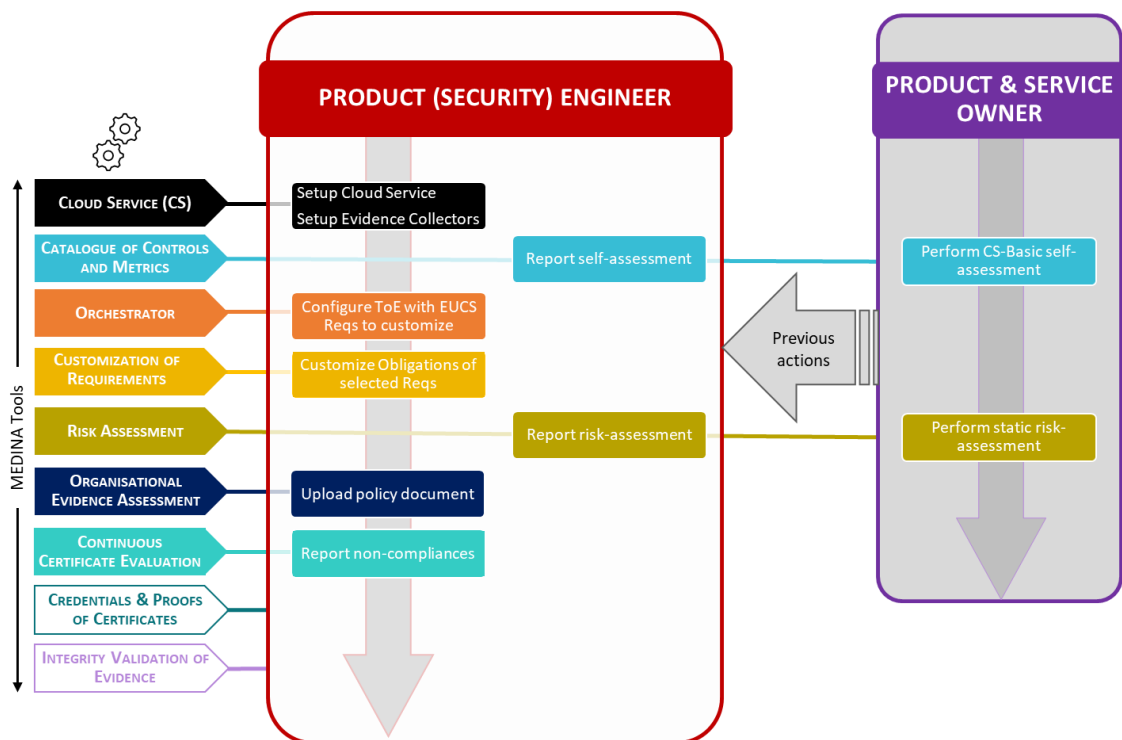


Figure 10. Workflow diagram for the Product (Security) Engineer user role

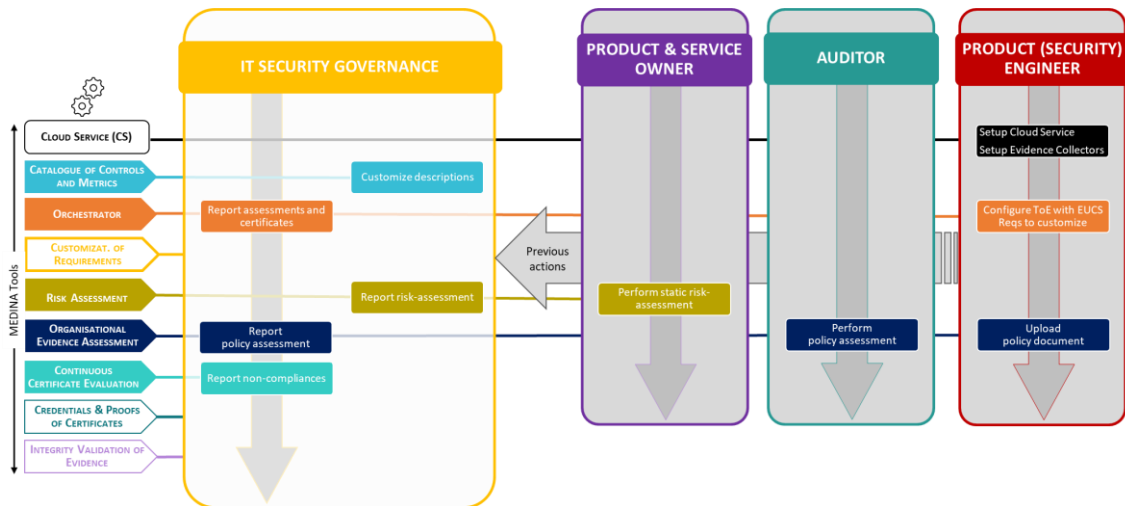


Figure 11. Workflow diagram for the IT Security Governance user role

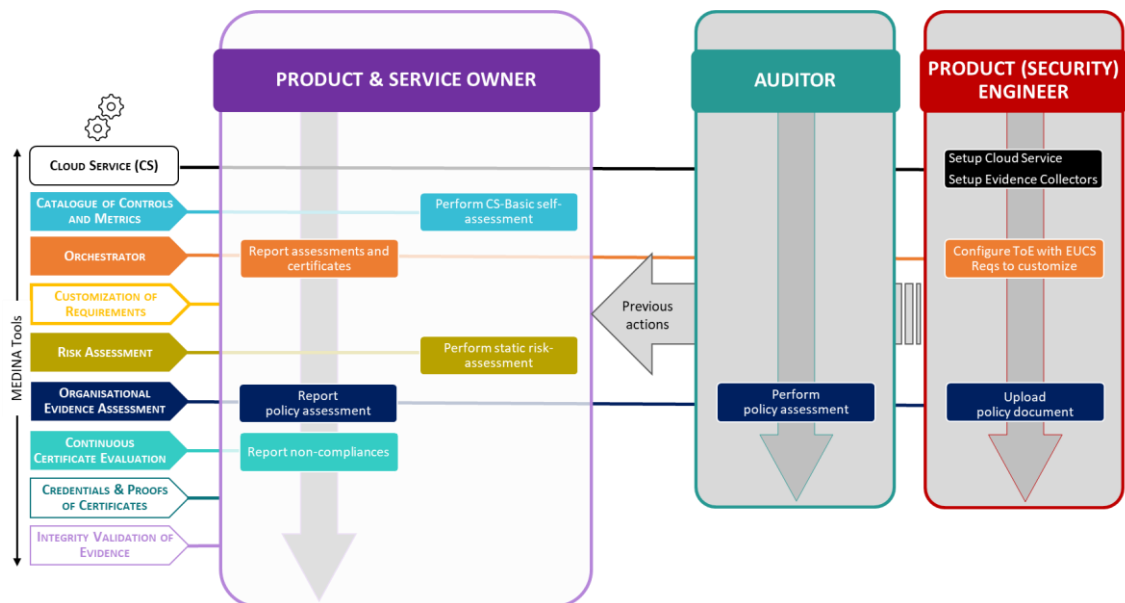


Figure 12. Workflow diagram for the Product and Service Owner user role

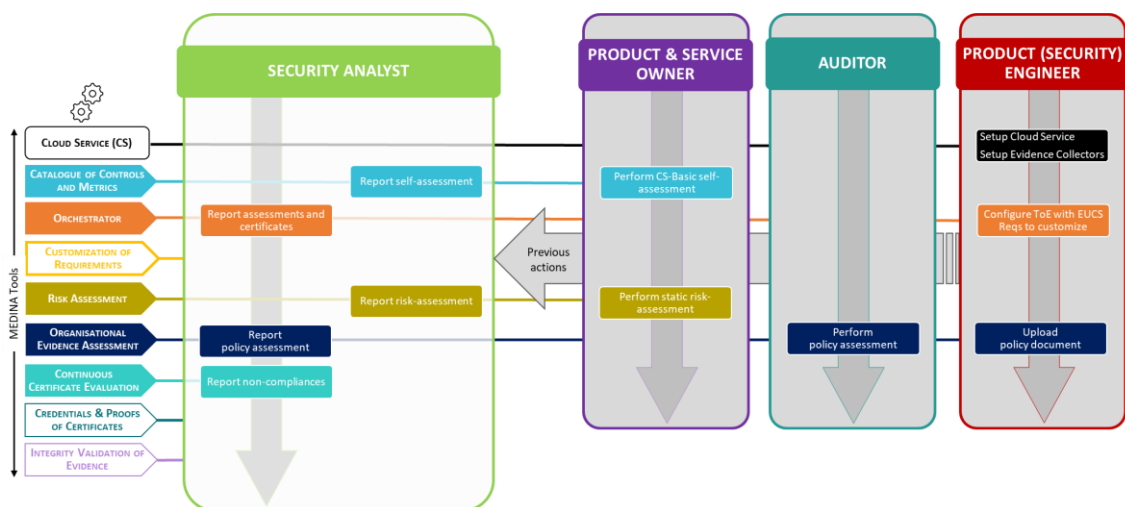


Figure 13. Workflow diagram for the Security Analyst user role

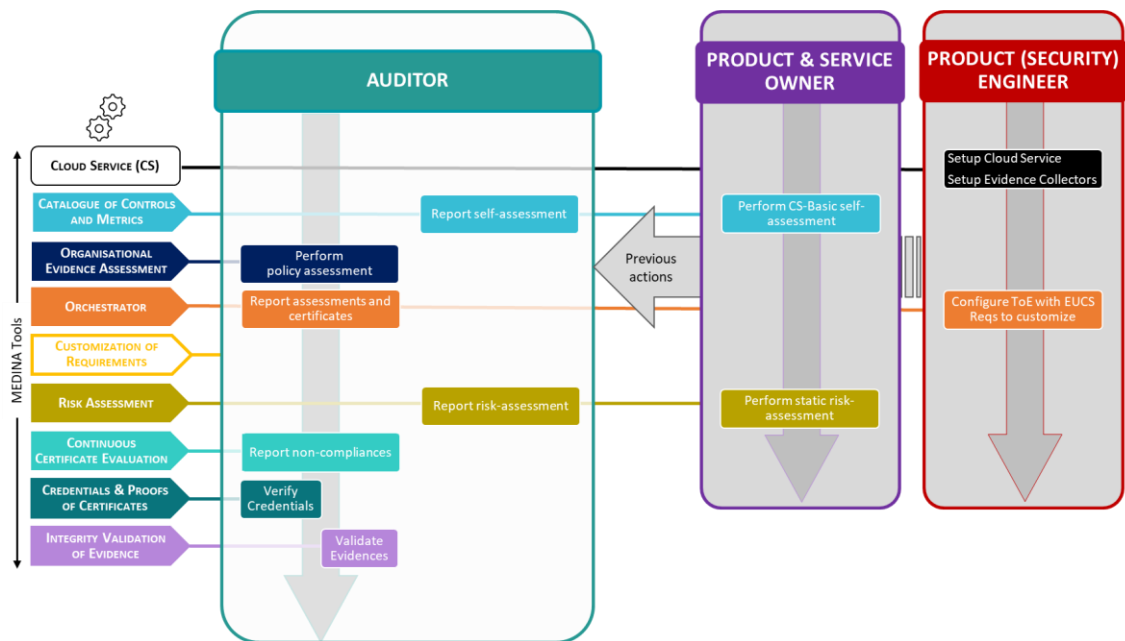


Figure 14. Workflow diagram for the Auditor role

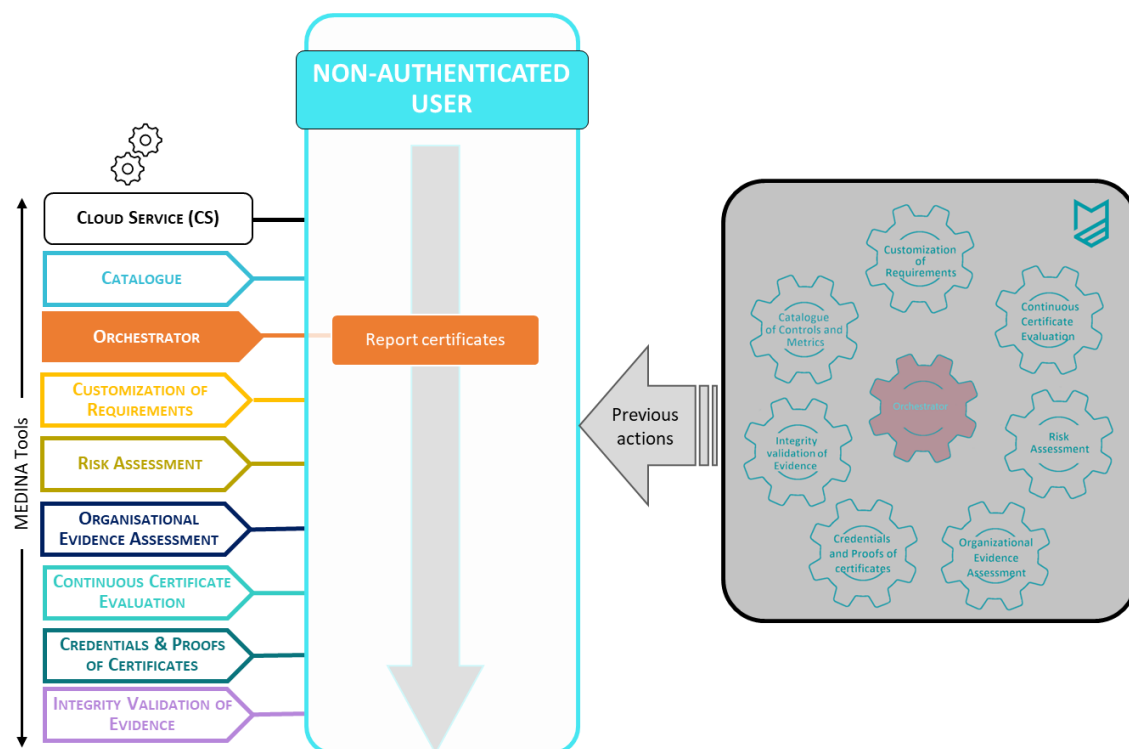


Figure 15. Workflow diagram for a non-authenticated user

4 MEDINA Framework Components and Integration

This section describes the status of the integration activities of the MEDINA components. Figure 16 represents the evolution of the architecture presented in D5.4 [2] and identifies the eight building blocks of the MEDINA framework, each one corresponding to a different functionality. Further information about the MEDINA architecture can be found in Deliverable D5.2 [3].

1. Catalogue
2. Certification Metrics and Language
3. Risk assessment and Optimisation Framework
4. Continuous Evaluation and Certification Life-Cycle
5. Organizational Evidence Gathering and Processing
6. Orchestrator and Databases
7. Evidence Collection and Security Assessment
8. Graphical User Interface

For each block there is a dedicated subsection below presenting the components that make up the block. An exception is block#8, which represents the User Interface block in MEDINA and integrates two components: Component Compliance Dashboard (CCD) and Integrated UI (IUI). While the CCD is described at the end of this section 4, the IUI is described in the dedicated section 5.

For each block component, we present a brief description of its role in the MEDINA framework and a reference to the deliverable containing more details about it. This is followed by information on the integration of the component with the other MEDINA components, the improvements achieved during the third integration round and the APIs exposed. Finally, if available, a brief description of the implemented Graphical User Interface (GUI) is included, as well as the TRL evaluation.

All component REST APIs are detailed in *APPENDIX F: Published APIs* and the user manuals of the components are included in *APPENDIX G: User Manuals*.

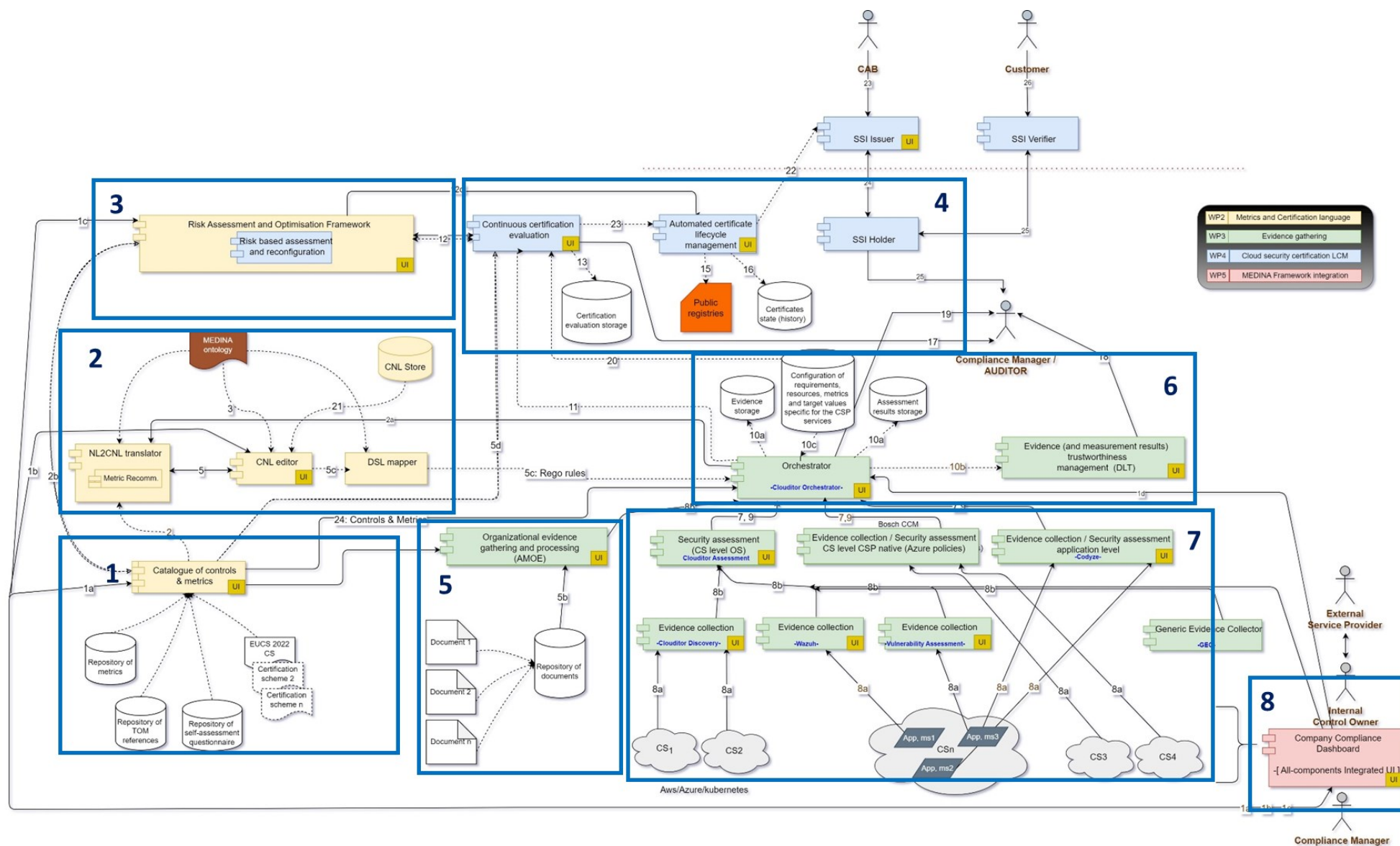


Figure 16. MEDINA Architecture and data flow

4.1 Catalogue (block #1)

The Catalogue is the component that implements most of the KR1 (Repository of metrics and measures). The main goal is to have an automated tool where a CSP compliance manager or an auditor can obtain all the information related to a security scheme, which in MEDINA is restricted to the EUCS (controls, security requirements, assurance levels, etc.). That is, everything that can be considered “static” information that appears in the standard.

As a result of the research performed in MEDINA, the Catalogue has been extended with extra information/functionalities such as metrics, implementation guidance, mapping of controls that are similar in other schemes, and self-assessment questionnaires.

4.1.1 Catalogue of Controls and Metrics

The *Catalogue of Controls and Metrics* (aka *Catalogue*) is one of the main entry points of the MEDINA framework. It provides the following functionalities:

- Endorsement of Security Control Frameworks and related entities: security requirements, categories, controls, metrics, and assurance levels.
- Displaying and filtering entity information based on some attribute values, including:
 - Displaying controls by category
 - Allowing navigation through categories, controls, requirements, and metrics
 - Selection of requirements of a certain assurance level
 - Selection of metrics related to a requirement.
- Provision of implementation guidelines²² for the set of requirements considered in MEDINA, i.e., requirements of assurance level high that require ‘continuous (automated)’ monitoring.
- Mapping of certification schemes, providing information about related controls from different frameworks, with respect to controls in EUCS.
- Self-assessment questionnaires (of about 1.000 questions) to check the degree of compliance of the August 2022 version of the European candidate draft EUCS [14].

The final version of the *Catalogue* incorporates the integration with the *Risk Assessment and Optimisation Framework*, through an API call that sends the results of the self-assessment questionnaire to SATRA.

The *Catalogue* is Open Source with license Apache 2.0 and the source code is available on the public GitLab repository²³.

The interested reader can find more information on the *Catalogue* in Deliverable D2.2 [13].

4.1.1.1 Implementation and Integration Status

The main updates implemented in the *Catalogue* from M27 to M33 are as follows:

- The GUI has been improved to facilitate navigation through the EUCS hierarchy. The same applies to the mapping of controls.
- The definition of the Implementation guidelines has been extended and better presented.

²² “Implementation guidelines” are what in previous deliverables we called “reference TOMs” (TOM refers to requirements or “Technical and Organizational Measures”). The name was changed because users indicated that the term “implementation guideline” better reflects the purpose of the information provided.

²³ <https://git.code.tecnalia.com/medina/public/catalogue-of-controls>

- The GUI has been adapted to the MEDINA *Integrated UI* visual guidelines (CSS based) and a Help menu button has been added to the toolbar with a link to the User Manual.
- The role-based authorization rules defined in Table 10 and Table 17 have been implemented. Different users now have different privileges depending on their role. For example, with regard to editing the EUCS framework, the creation of questionnaires, or the insertion of non-conformities.
- Self-assessment questionnaires have been implemented for the three levels of EUCS certification (Basic / Substantial / High). A full report of the assessment is generated in PDF format.
- The results of a questionnaires are sent to the SATRA component, via API, whenever the user finishes working on it.

The final version of the *Catalogue* in M33 fully meets 100% of the requirements (10) defined in deliverable D2.2 [13]. Two requirements that were partially fulfilled in the second integration round are now fully fulfilled. Specifically, requirement #7 has been fulfilled by implementing the connection to the *Risk Assessment* component (SATRA); and requirement #9, which refers to the self-assessment questionnaires, has been fulfilled by providing all the required features (i.e., selecting the assurance level, covering all EUCS requirements, allowing to enter comments and evidence references, and providing a summary dashboard).

The *Catalogue* frontend is integrated with the MEDINA *Integrated UI*, the user can access it by clicking on the left menu option “Catalogue of Controls and Metrics” (see Figure 18). The *Catalogue* is also integrated with the user management tool (Keycloak), so it is able to control the logged user and its properties, specifically the role. The *Catalogue* provides a GUI for end users (see section 4.1.1.3), as well as a RESTful API to interact with it (see section 4.1.1.2).

The *Catalogue* provides data to the following MEDINA components: *Orchestrator* (controls and metrics), *NL2CNL Translator* (metrics), *SATRA* (answers to the questionnaires), *AMOE* (control and metrics), and *CCE* (relations between metrics, requirements, controls, categories).

The connection of the two questionnaires avoids the user having to fulfil two overlapping questionnaires, so that the information collected from the user’s responses is consolidated in the *Catalogue* and then shared with *SATRA*. It is worth mentioning that both tools implement questionnaires, but differ in several details:

1. The questionnaires are used for different purposes: the *SATRA* questionnaire implements a risk-based analysis of failed requirements, while the *Catalogue* questionnaire focuses on compliance. Hence, the reports provided by the two tools are quite different.
2. The granularity differs: the *SATRA* questionnaire is performed at the control-level while the *Catalogue* questionnaire includes questions at the requirement-level.
3. The audiences may also be different: both questionnaires can be used by CSPs for self-assessment, but the *Catalogue* questionnaire can also be used by auditors.

Every time a Questionnaire is saved, those requirements for which compliance has been calculated are sent to *SATRA* (see Figure 17). The degree of compliance with a Requirement is calculated based on the answers provided by the CSP to the corresponding questions according to the rules displayed in Figure 17.

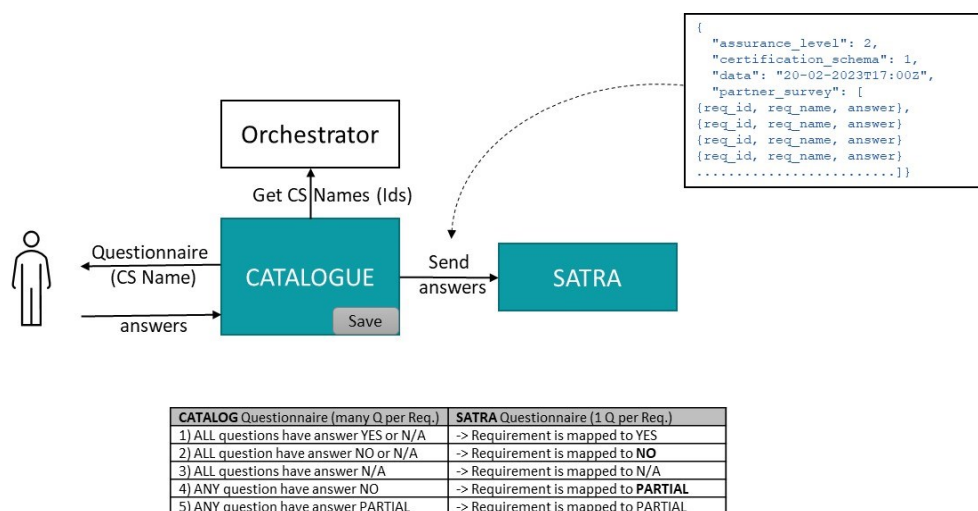


Figure 17. Connection Catalogue-SATRA for questionnaire management

4.1.1.2 Published APIs

The *Catalogue* has implemented all internal functionality for other components to access and modify the database elements as a REST API, so the number of interfaces and endpoints is quite extensive. The list of the available APIs is gathered in *APPENDIX F: Published APIs, Component: Catalogue of Controls*. All of them are available to components that wish to interact with the *Catalogue*.

The full API is also available online in the repository²⁴ as an OpenAPI definition.

4.1.1.3 Graphical User Interface

The *Catalogue* offers a GUI to access and manipulate the different entities that are stored in the database (see Figure 18). A CRUD screen (Create/Retrieve/Update/Delete) has been developed for each of the main entities, although the actions allowed depend on the user role.

The GUI allows the user to navigate through the EUCS framework entities, using the visual elements on the different screens -like buttons, links, and filters-. For example, the user can select the requirements of a certain assurance level, the controls of a category, the metrics related to a specific requirement, etc.

As for all the other components – in the right upper corner of the graphical interface there is a “Help” button that opens the User Manual in a new tab. The User Manual is also available in *APPENDIX G: User Manuals*.

²⁴ <https://git.code.tecnalia.com/medina/public/catalogue-of-controls/-/blob/main/openapi.json>

Code	Name	Description	Framework	Controls	
A1	Organisation of Information Security	Plan, implement, maintain and continuously improve the information security framework within the organisation	EUCS ↑	Controls ↓	View
A2	Information Security Policies	Provide a global information security policy derived into policies and procedures regarding security requirements and to support business requirements	EUCS ↑	Controls ↓	View
A3	Risk Management	Provide a global information security policy, derived into policies and procedures regarding security requirements and to support business requirements	EUCS ↑	Controls ↓	View
A4	Human Resources	Ensure that employees understand their responsibilities, are aware of their responsibilities with regard to information security, and that the organisation's assets are protected in the event of changes in responsibilities or termination	EUCS ↑	Controls ↓	View
A5	Asset Management	Identify the organisation's own assets and ensure an appropriate level of protection throughout their lifecycle	EUCS ↑	Controls ↓	View
A6	Physical Security	Prevent unauthorised physical access and protect against theft, damage, loss and outage of operations	EUCS ↑	Controls ↓	View
A7	Operational Security	Ensure proper and regular operation, including appropriate measures for planning and monitoring capacity, protection against malware, logging and monitoring events, and dealing with vulnerabilities, malfunctions and failures	EUCS ↑	Controls ↓	View
A8	Identity, Authentication and Access Control Management	Limit access to information and information processing facilities	EUCS ↑	Controls ↓	View
A9	Cryptography and Key Management	Ensure appropriate and effective use of cryptography to protect the confidentiality, authenticity or integrity of information	EUCS ↑	Controls ↓	View
A10	Communication Security	Ensure the protection of information in networks and the corresponding information processing systems	EUCS ↑	Controls ↓	View
A11	Portability and Interoperability	Enable the ability to access the cloud service via other cloud services or IT systems of the cloud customers, to obtain the stored data at the end of the contractual relationship and to securely delete it from the Cloud Service Provider	EUCS ↑	Controls ↓	View

Figure 18. Catalogue GUI

In the following, some screenshots are presented as samples of the GUI, in particular to show:

- Controls (see Figure 19)
- Requirements (see Figure 20)
- Filters (see Figure 21 and Figure 22)
- Metrics (see Figure 23) and details of a metric (see Figure 24)
- Implementation guidelines (Figure 25)
- Questionnaires (see Figure 26)

Code	Name	Description	Category	Requirements	Other Frameworks	
OIS-01	INFORMATION SECURITY MANAGEMENT SYSTEM	The CSP operates an information security management system (ISMS). The scope of the ISMS covers the CSPs organisational units, locations and processes for providing the cloud service.	Organisation of Information Security ↑	Requirements ↓	Similar Controls ↓	View Edit
OIS-02	SEGREGATION OF DUTIES	Conflicting tasks and responsibilities are separated based on an RM-01 risk assessment to reduce the risk of unauthorised or unintended changes or misuse of cloud customer data processed, stored or transmitted in the cloud service.	Organisation of Information Security ↑	Requirements ↓	Similar Controls ↓	View Edit
OIS-03	CONTACT WITH AUTHORITIES AND INTEREST GROUPS	The CSP stays informed about current threats and vulnerabilities by maintaining the cooperation and coordination of security-related aspects with relevant authorities and special interest groups. The information flows into the procedures for handling risks (cf. RM-01) and vulnerabilities (cf.	Organisation of Information Security ↑	Requirements ↓	Similar Controls ↓	View Edit

Figure 19. Catalogue - Controls of the "Organisation of Information Security" category

Requirements							Show/Hide filter
Home » Frameworks » Categories » Controls » Requirements							
Code	Description	Assurance Level	Type	Control	Implementation guidelines	Metrics	
OIS-01.1B	The CSP shall have an information security management system (ISMS), covering at least the operational units, locations, people and processes for providing the cloud service.	translation-not-found[cocGatewayApp.AssuranceLevel.Basic]	Organizational	OIS-01 ↑	----	No Metrics	View Edit
OIS-01.2B	The CSP shall provide documented information of the ISMS applied to the cloud service.	translation-not-found[cocGatewayApp.AssuranceLevel.Basic]	Organizational	OIS-01 ↑	----	No Metrics	View Edit
OIS-01.1S	The CSP shall have an information security management system (ISMS), covering at least the operational units, locations, people and processes for providing the cloud service, in accordance with EN ISO/IEC 27001. Where the controls referred to in ISO/IEC 27001 6.1.3 shall be the controls in this TS on level Substantial	translation-not-found[cocGatewayApp.AssuranceLevel.Substantial]	Organizational	OIS-01 ↑	----	No Metrics	View Edit

Figure 20. Catalogue - List of Requirements

Search for requirements							
Security Control Framework							
EUCS							
Select an assurance Level							
High							
ID	Code	Name	Description	Assurance Level	Type	Security Control	
5	OIS-01.1H	OIS-01.1H	The CSP shall have an information security management system (ISMS), covering at least the operational units, locations, people and processes for providing the cloud service, with a valid certification of compliance with the requirements of EN ISO/IEC 27001 or with national schemes based on ISO 27001, issued by an accredited CAB covering the cloud service.	High	Organizational	OIS-01	View
6	OIS-01.2H	OIS-01.2H	The CSP shall provide documented information of the ISMS applied to the cloud service, including at least - (1) ISO/IEC 27001 requirement 6.1.3 item c) shall be used for the cloud service using the controls in this document for comparison, with the restriction that all controls shall apply - (2) ISO/IEC 27001 requirement 6.1.3 item d) producing a Statement of Applicability referring to the controls in this document for the cloud service.	High	Organizational	OIS-01	View
12	OIS-02.1H	OIS-02.1H	The CSP shall perform a risk assessment as defined in RM-01 about the accumulation of responsibilities or tasks on roles or individuals, regarding the provision of the cloud service, covering at least the following areas, insofar as these are applicable to the provision of the cloud service and are in the area of responsibility of the CSP - (1) Administration of rights profiles, approval and assignment of access and access authorisations (cf. IAM-01); - (2) Development, testing and release of changes (cf. DEV-01, CCM-01); and - (3) Operation of the system components.	High	Organizational	OIS-02	View
13	OIS-02.2H	OIS-02.2H	The CSP shall implement the mitigating measures defined in the risk treatment plan, privileging separation of duties, unless impossible for organisational or technical reasons, in which case the measures shall include the monitoring of activities in order to detect unauthorised or unintended changes as well as misuse and the subsequent appropriate actions.	High	Organizational	OIS-02	View
14	OIS-02.3H	OIS-02.3H	The CSP introduces and maintains an inventory of conflicting roles and enforces the segregation of duties during the assignment or modification of roles as part of the role management process.	High	Organizational	OIS-02	View
15	OIS-02.4H	OIS-02.4H	The CSP shall automatically monitor the assignment of responsibilities and tasks to ensure that measures related to segregation of duties are enforced.	High	Organizational	OIS-02	View
18	ISP-	ISP-	The CSP shall review the global information security policy at least annually	High	Organizational	ISP-01	View

Figure 21. Catalogue - Filter to search for “EUCS & High” requirements

Controls Show/Hide filter

Home » Frameworks » Categories » Controls

Code: Name: Description:

Category:

Clear filter Search

Code	Name	Description	Category	Requirements	Other Frameworks
CKM-01	POLICIES FOR THE USE OF ENCRYPTION MECHANISMS AND KEY MANAGEMENT	Policies and procedures for cryptography and key management including technical and organisational safeguards are documented, communicated, and implemented, in order to ensure the confidentiality, authenticity and integrity of the information.	Cryptography and Key Management ↑	Requirements ↓	Similar Controls ↓ View Edit
CKM-02	ENCRYPTION OF DATA IN TRANSIT	CCSC data communicated over public networks is protected in confidentiality, integrity, and authenticity.	Cryptography and Key Management ↑	Requirements ↓	Similar Controls ↓ View Edit
CKM-03	ENCRYPTION OF DATA AT REST	The CSP has established procedures and technical safeguards to prevent the disclosure of cloud customers' data during storage.	Cryptography and Key Management ↑	Requirements ↓	Similar Controls ↓ View Edit
CKM-04	SECURE KEY MANAGEMENT	Appropriate mechanisms for key management are in place to protect the confidentiality, authenticity or integrity of cryptographic keys.	Cryptography and Key Management ↑	Requirements ↓	Similar Controls ↓ View Edit

Figure 22. Catalogue - Filter to search for “CKM” controls

Metrics (Requirement: OPS-05.3H) Show/Hide filter

Home » Frameworks » Category: Operational Security » Control: OPS-05 » Requirement: OPS-05.3H » Metrics

Category	Name	Source	Description	Operator	Requirements	
Operational security	MalwareProtectionEnabled	Technical	This metric is used to assess if the antimalware solution is enabled on the respective resource.	==	OPS-05.3H ↑	View
Operational security	NumberOfThreatsFound	Technical	This metric is used to assess if the antimalware solution reports no irregularities.	==	OPS-05.3H ↑	View
Operational security	MalwareProtectionOutput	Technical	This metric states whether automatic notifications are enabled (e.g. e-mail) about malware threats. This relates to EUCS' definition of "continuous monitoring".	==	OPS-05.3H ↑	View
Operational security	MalwareProtectionCheckQ3	Organizational	What antivirus system is used?	n/a	OPS-05.3H ↑	View
Operational security	AntimalwareScanFrequencyQ1	Organizational	How frequent are antimalware scans done?	<=	OPS-05.3H ↑	View

Figure 23. Catalogue - Metrics implemented for the “OPS-05.3H” requirement

Catalogue

Questionnaires

Help

Administration

Metric

Metric Id	1
Category	Operational security
Name	MalwareProtectionEnabled
Source	Technical
Description	This metric is used to assess if the antimalware solution is enabled on the respective resource.
Scale	[true, false]
Operator	==
Target Value	true
Target Value Datatype	Boolean
Interval	1
Target Resource Type	VirtualMachine
Requirements	OPS-05.3H
Resource Type	
Security feature	malwareProtection.enabled
Keywords	

← Back

Figure 24. Catalogue - Details of a the “MalwareProtectionEnabled” metric

Implementation guideline

Requirement **ISP-03.5H**

The EUCS requirement (ISP-03.5H) states:
 "The list of exceptions shall be automatically monitored to ensure that the validity of approved exceptions has not expired and that all reviews and approvals are up-to-date" and references the following requirement also from (ISP-03) Exceptions:

ISP-03.1H "The CSP shall maintain a list of exceptions, limited in time, to the security policies and procedures, including associated controls."

EUCS Security Control

Code	Name	Objective
ISP-03	Exceptions	Exceptions to the policies and procedures for information security as well as respective controls are explicitly listed

References

Internal references:

- EUCS - RM-01: Risk Management Policy

External references:

- 2020 GERMANY - SP-03: Exceptions from Existing Policies and Instructions
- Cisco CCF - CCF 108: Policy and Standard Exceptions

Key concepts

Term	Definition
Information security	Preservation of confidentiality, integrity and availability of information
Information Security Management system (ISMS)	An information security management system (ISMS) is a framework of policies, processes, and controls that organizations use to manage and reduce their information security risks. Generally, an ISMS is designed to protect the confidentiality, integrity, and availability of the organization's information assets, and can include both technical and non-technical measures
Exception	Exceptions to information security policies, standards, guidelines, and procedures
Risk Management	Risk management is the identification, evaluation, and prioritization of risks. An ISMS includes a process for identifying and assessing the organization's information security risks, and for developing plans to mitigate those risks

Guidelines

This security control ensures that exceptions to the policies and procedures for information security as well as respective controls are explicitly listed. Deviation from the Information Security policy implemented by the CSP is discouraged. However, exception may be considered if a presentation of a reasonable and justifiable reason is provided. The expression "there is an exception to every rule" is also true in information security policies context. There are often legitimate reasons why an exception to a policy is needed. In these cases, the policy should define how approval for the exception to the policy is obtained, and management should be aware of exceptions to security policies as the exception to the policy could introduce risks that need to be mitigated in another way.

In the context of EUCS requirements exceptions can have organizational or technical causes, such as:

- An organizational unit deviating from the intended processes and procedures in order to meet the requirements of a cloud customer.
- A system component lacking technical properties to be configured according to the applicable requirements.

Cloud customers can use appropriate controls to ensure that they obtain information from the Cloud Service Provider about deviations from information security policies and instructions in order to assess and appropriately manage the associated risks to their own information security.

While at basic assurance level, maintaining a list of exceptions is sufficient, at substantial level, those exceptions are required to be approved and taken into account by the risk management. Therefore, the exceptions need to be collected and approved, as part of the risk management process. A complete description of the exception shall be maintained including relevant information such as exception description, exception duration, compensating controls for managing the risk associated with the exception, proposed review date, or others. The approvals of exceptions may be documented, limited in time and reviewed for appropriateness at least annually by the risk owners.

At the high assurance level, the list of the exception must also be automatically monitored. The continuous monitoring of the exceptions list should be automated to ensure that they do not exceed their "lifetime" in the system and that exceptions do not remain active after approval has been revoked. Such a monitoring tool should be capable of issuing notifications and regular status updates when an exception expires, has been approved or has been revoked by the risk owner. This could be achieved through languages similar to the one defined in the OSCAL (Open Security Controls Assessment Language).

← Back Edit

Figure 25. Catalogue - Implementation guideline for the ISP-03.5 requirement

Questionnaire

2023-06-01 >> FabasoftTestCCD >> EUCS >> Basic

Categories

- A1: Organisation of Information Security
- A2: Information Security Policies
- A3: Risk Management
- A4: Human Resources
- A5: Asset Management
- A6: Physical Security
- A7: Operational Security
- A8: Identity, Authentication and Access Control Management
- A9: Cryptography and Key Management
- A10: Communication Security
- A11: Portability and Interoperability
- A12: Change and Configuration Management
- A13: Development of Information Systems
- A14: Procurement Management
- A15: Incident Management
- A16: Business Continuity
- A17: Compliance
- A18: User Documentation
- A19: Dealing with Investigation Requests from Government Agencies
- A20: Product Safety and Security

A1: Organisation of Information Security

Choose a Control: **OIS-01** OIS-02 OIS-03 OIS-04

OIS-01: The CSP operates an information security management system (ISMS). The scope of the ISMS covers the CSPs organisational units, locations and processes for providing the cloud service.

OIS-01.1B: The CSP shall have an information security management system (ISMS), covering at least the operational units, locations, people and processes for providing the cloud service.

Q1: Does the CSP have an information security management system (ISMS) documented?

☒ Fully supported.
☐ Partially supported.
☐ Not supported at all.
☐ Not applicable.

Evidence: - Documented Information Security Management System (ISMS)

Comments:

Q2: Does the information security management system cover the operational units?

☐ Fully supported.
☒ Partially supported.
☐ Not supported at all.
☐ Not applicable.

Evidence: - ISMS scope (operational units)

Comments:

Q3: Does the information security management system (ISMS), cover locations?

☐ Fully supported.
☐ Partially supported.
☐ Not supported at all.
☒ Not applicable.

Evidence: - ISMS scope (locations)

Comments:

Q4: Does the CSP cover processes for providing the cloud service?

☐ Fully supported.
☐ Partially supported.
☐ Not supported at all.
☒ Not applicable.

Evidence: - ISMS scope (processes for providing the cloud service)

Comments:

Non-conformities of the requirement:

Compliance: **PARTIAL**

Figure 26. Catalogue - Questionnaire - Questions for the OIS-01 Security control, Basic level.

4.1.1.4 TRL

The TRL of the *Catalogue of Controls and Metrics* is 4 at the moment of writing. After the validation phase ends, it is expected to be 5.

4.2 Certification Metrics and Language (block #2)

The components belonging to the “Certification Metrics and Languages” block are mainly related with KR3 (Certification Language), whose objective is to provide a language specification which expresses the most relevant aspects of a security certification scheme in machine-readable format using a Domain Specific Language (DSL).

The updated detailed description of these components can be found in the deliverable D2.5 [15].

4.2.1 NL2CNL Translator

The *NL2CNL Translator* is the MEDINA component used to map EUCS Natural Language (NL) requirements into their MEDINA Controlled Natural Language (CNL) translation. This translation is performed in two steps: the first one selects a set of metrics that could be useful to evaluate a certain security requirement. After associating a set of metrics with a requirement, the second step translates those metrics into policies. Specifically, requirements and metrics are expressed in NL, while the translated policies are expressed in CNL. The final version of the *NL2CNL Translator* component incorporates the integration with the *Catalogue of Controls and Metrics*, the *CNL Store* through the *CNL Editor* APIs, and the *Orchestrator*.

The *NL2CNL Translator* is Open Source with license Apache 2.0 and the source code is available on the public GitLab repository²⁵.

More details on the *NL2CNL Translator* are described in deliverable D2.5 [15].

4.2.1.1 Implementation and Integration Status

The *NL2CNL Translator* was already in a mature stage at M27, thus the main changes until M33 concern implementation refinement, testing and bug fixing.

Some changes have been made to keep the implementation updated with respect to the other components. Specifically, the vocabulary has been updated and the NLP features have been recomputed each time a new requirement/metric has been added to the *Catalogue*. Moreover, as regards the connection with the *Catalogue*, the *Orchestrator* and the *CNL Editor*, all the 34 EUCS requirements considered in MEDINA have been tested and the correct generation of the correspondent REOs has been verified. An additional modification concerning the previous version regards the implementation of the filtering concept, i.e., the *NL2CNL Translator* in its final version receives from the *Orchestrator* the specification of the Cloud Service ID to which a requirement relates.

4.2.1.2 Published APIs

The *NL2CNL Translator* provides a REST API that can be used by the other components interacting with it. The list of the available APIs is provided in *APPENDIX F: Published APIs, Component: NL2CNL Translator and DSL Mapper*.

²⁵ <https://git.code.tecnalia.com/medina/public/nl2cnl-translator>

4.2.1.3 Graphical User Interface

This component does not have a graphical interface and interacts with the other components of the framework via the API.

4.2.1.4 TRL

The TRL of the *NL2CNL Translator* is 4 at the moment of writing. After the end of validation phase, it is expected to be 5.

4.2.2 CNL Editor

CNL Editor is the component that allows a CSP user to manage, with a Graphical Interface, the Requirements and Obligations (named REO) objects that are the association, in CNL format, between Requirements and Policies as compiled from *NL2CNL Translator*. *CNL Editor* takes as input REOs created by *NL2CNL Translator* and produces as output updated REOs to be processed by *DSL Mapper*. With the Editor Frontend, the user can visualize REOs, change Target Value specified for the Metrics and delete Obligations not considered suitable for the CSP. Finally, the user can send the REO to the *DSL Mapper* with the “map” operation which convert CNL obligations into Rego²⁶ Code (see section 4.2.3).

During the third integration round, we worked on the filtering facility, so we changed the accessibility to a REO based on Cloud Service Id of the REO and not on the REO user creator as before. In addition, we reviewed some UI details to be more compliant to the MEDINA standards defined.

CNL Editor is close source/proprietary code (Copyright by HPE) and is stored in a private GitLab repository.

More details on *CNL Editor* are described in deliverable D2.5 [15].

4.2.2.1 Implementation and Integration Status

CNL Editor is composed by the following modules:

- CNL Editor Interface (the web GUI to access *CNL Editor*) and CNL Editor core (application core)
- Vocabulary: a RDF file with .owl extension defining the Ontology structures and terms necessary for the Editor to control user changes to the Obligations .
- CNL Editor REST API: APIs used by the Editor and eventually by other Certification Languages tools, *NL2CNL Translator* and *DSL Mapper*, for basic operations.
- CNL Store: database with REO xml files.
- Back Store Interface: REST APIs for access to the CNL Store used by *CNL Editor*.

CNL Editor was partially containerized on a VM standalone in M15. From M15 onwards the xml structure of the REO to reflect the needs of MEDINA based on partners requests was revised, and the API was also renamed and adapted to better fit the MEDINA context.

At the time of writing *CNL Editor* is implemented in a mature version and has been fully deployed in the MEDINA Kubernetes cluster. It provides both a GUI for end users and a set of RESTful APIs to interact with it. The vocabulary used by Editor was updated to the Metrics available in the final version of *Catalogue of Controls and Metrics*.

²⁶ <https://www.openpolicyagent.org/>

The main updates from M27 to M33 include the implementation of the Cloud Service Id filtering, the building of a new API, and changing the REO list selection for a particular user. Users do not see any more only REOs created with their account name but can visualize all REOs that are defined for Cloud Service Ids that are present in their Keycloak user profile.

At the time of writing, it is being considered, as an enhancement to the *CNL Editor*, the implementation of the Role Authorization feature with two different authorizations as specified in Table 13. Thus, a user with “Read” role will only be able to “Show” REOs, and a user with “Write” Role [Product (Security) Engineer] will be authorized to perform all available operations on REOs.

CNL Editor frontend is integrated with the *MEDINA Integrated UI*, the user can access it by clicking on the left menu option “Customization of Requirements” (see Figure 27). User authentication is done via the MEDINA Keycloak service.

CNL Editor interacts with the other components, *NL2CNL Translator* and *DSL Mapper*, by REST APIs (see section 4.2.2.2).

4.2.2.2 Published APIs

CNL Editor makes available APIs that can be used from other components (e.g., create by NL2CNL Translator) to manage REOs, and that are listed in *APPENDIX F: Published APIs, Component: CNL Editor*.

To implement REO filtering, on Cloud Service Id, a new API was defined

/reo/filterby/cloudservice

that retrieves the list of REOs that are associated to a list of Cloud Service Ids.

4.2.2.3 Graphical User Interface

CNL Editor has a Web Interface that allows a user visualizing and managing some changes to the REOs. Operations allowed for a REO include: delete obligations or change the Target Values of obligations.

The final version of the tool in M33 was optimized in terms of look and feel with these enhancements:

- alignment to the MEDINA UI style for characters, buttons colour, header, footer, and background
- addition of the Help button with a link to the User Manual, which is also available in *APPENDIX G: User Manuals*.

When the user invokes *CNL Editor* a list of REOs is displayed, as shown in Figure 27.

MEDINA

1

About

Catalogue of Controls and Metrics

Orchestrator

Customization of Requirements

Risk Assessment

Organisational Evidence Assessment

Continuous Certificate Evaluation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 952931

Customization of Requirements

uc1_proddsec

Help

Filter by Name, ID, or Status

Name	Creator	Status	Creation Date	ID	Cloud Service ID
REO from CKM-04.4H	UC1_ProdSec	Completed	2023-05-05	DSA-05634f2c-1ad2-4e85-b3a4-bd47d2df10d	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from CKM-04.1H	UC1_ProdSec	Completed	2023-07-06	DSA-144e8b14-a0e4-46c5-a3d6-e5a714aea8c6	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from CKM-04.2H	UC1_ProdSec	Completed	2023-07-06	DSA-15435991-59f4-4eff-8729-fc642ef76d5	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from CKM-02.1H	UC1_ProdSec	Completed	2023-05-05	DSA-17a4a631-3a21-4150-819b-5aa5c1f71a35	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from CS-07.3H	UC1_ProdSec	Completed	2023-05-12	DSA-254cb5df-6416-4a3c-b136-ec0ee99456d8	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from BC-01.1H	UC1_ProdSec	Customised	2023-07-12	DSA-2a131ba1-d055-4a2f-9419-fa8c3756894a	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from OPS-13.1H	UC1_ProdSec	Completed	2023-05-05	DSA-2af76a5a-1b99-4f3e-96c6-fb7b7c47e667	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from AM-04.2H	UC1_ProdSec	Completed	2023-07-10	DSA-3a6af73c-1f0b-4179-8c23-9a483d56b48b	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from CS-07.2H	UC1_ProdSec	Completed	2023-05-12	DSA-437d0bc7-1ca4-47e2-9636-573f5bae8b78	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from CKM-04.2H	UC1_ProdSec	Completed	2023-07-06	DSA-4b82a7fb-4866-4257-b518-5fd21deb9661	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from AM-01.4H	UC1_ProdSec	Customised	2023-05-04	DSA-5dfcbe9d-694a-4f70-90ec-8f568eb46d18	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from CCM-03.4H	UC1_ProdSec	Customised	2023-05-08	DSA-5e15cd00-4801-4cf4-93dc-f0653cd1ab1b	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from AM-04.3H	UC1_ProdSec	Customised	2023-07-10	DSA-72213b28-e762-426b-9167-6941cc8fe73b	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from CKM-04.3H	UC1_ProdSec	Available	2023-05-05	DSA-a310778d-e753-435f-b1e6-c591f8035796	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from AM-05.3H	UC1_ProdSec	Customised	2023-05-08	DSA-a761ab31-817b-4f05-9926-2e5381bfaa09	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from OPS-21.1H	UC1_ProdSec	Available	2023-05-05	DSA-ac19cdee-fdc5-4ddb-bcd4-f934dfb1efc1	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from CCM-01.1H	UC1_ProdSec	Completed	2023-07-07	DSA-ad8a7c69-f77a-449a-96c8-2b81e59e82df	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from CKM-03.3H	UC1_ProdSec	Customised	2023-07-06	DSA-b68ade38-ce57-45e6-bdf3-929b85490b0d	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from BC-01.2H	UC1_ProdSec	Customised	2023-07-12	DSA-b72d9c65-0168-41eb-9fd7-ac59602ba6d5	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from CKM-04.1H	UC1_ProdSec	Customised	2023-07-06	DSA-ddc45c2d-2cb2-461d-af4f-4ff61f0467e0	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from CS-07.1H	UC1_ProdSec	Customised	2023-05-12	DSA-de18bcaa-b9c9-4025-9ffb-2848c346014b	937210b1-a9f2-4929-bbbc-5a7ecc0f089f
REO from OPS-21.2H	UC1_ProdSec	Available	2023-05-05	DSA-efba7301-9a63-4abc-a021-3dc4cb1d2340	937210b1-a9f2-4929-bbbc-5a7ecc0f089f

Figure 27. CNL Editor – REOs visualization

When the user selects a specific REO the window shown in Figure 28 is displayed.

Back

Title

REO from BC-01.1H

Status

CUSTOMISED

Date

2023-07-12 13:41:37

Cloud Service ID

937210b1-a9f2-4929-bbbc-5a7ecc0f089f

Additional Information

UUID

DSA-2a131ba1-d055-4a2f-9419-fa8c3756894a.xml

Vocabulary URI

https://cnl-vocabulary-test.k8s.medina.es/labels/vocabularies/medina_vocabulary_test_v2.0.owl#

Requirement

Requirement Code

BC-01.1H

Security Control

BC-01

Framework

EUCS

Type

ORGANIZATIONAL

Description

The CSP shall define policies and procedures according to ISP-02 establishing the strategy and guidelines to ensure business continuity and contingency management.

Assurance level

High

Obligations

Policies

PolicyDocument MUST AssetManagementPolicy02 String(sln.[procurement, destruction, none])

AssetManagementPolicy02 / recommender

PolicyDocument MUST BusinessContinuityPolicy01 na(na,na)

BusinessContinuityPolicy01 / recommender

PolicyDocument MUST BusinessContinuityPolicy02 na(na,na)

BusinessContinuityPolicy02 / recommender

PolicyDocument MUST BusinessContinuityPolicy03 String(sln.[multi-datacenter architecture, other, none])

BusinessContinuityPolicy03 / recommender

PolicyDocument MUST ChangeManagementPolicy01 na(na,na)

ChangeManagementPolicy01 / recommender

PolicyDocument MUST GuidelinesCloudCustomersQ1 na(na,na)

GuidelinesCloudCustomersQ1 / recommender

PolicyDocument MUST GuidelinesCloudCustomersQ2 na(na,na)

GuidelinesCloudCustomersQ2 / recommender

PolicyDocument MUST GuidelinesCloudCustomersQ3 na(na,na)

GuidelinesCloudCustomersQ3 / recommender

PolicyDocument MUST PolicyUpToDateCheck Integer(< 180)

PolicyUpToDateCheck / recommender

Figure 28. CNL Editor – Showing a specific REO

4.2.2.4 TRL

The TRL of the *CNL Editor* is 4 at the moment of writing. After the validation phase ends, it is expected to be 5.

© MEDINA Consortium
www.medina-project.eu

Contract No. GA 952633

Page 54 of 147

(CC) BY-SA

4.2.3 DSL Mapper

The *DSL Mapper* is a component of the MEDINA framework that has the aim of mapping the obligations expressed in Controlled Natural Language into executable policies expressed in Domain Specific Language (DSL). In particular, the obligations generated by the *NL2CNL Translator* are read from the CNL Store in the form of a REO object, while the output generated by the *DSL Mapper* is expected to be compliant with the DSL chosen in MEDINA, i.e., the Rego language. The Rego language allows the creation of policies that can be used to automatically assess evidence, collected by the evidence collector components. The output of the *DSL Mapper* is sent to the *Orchestrator*, which performs the assessment of the policies.

The *DSL Mapper* is Open Source with license Apache 2.0 and the source code is available on the public GitLab repository²⁷.

More details on *DSL Mapper* are described in deliverable D2.5 [15].

4.2.3.1 Implementation and Integration Status

Compared with M27, this component is now in a mature state. The most important change is the refinement of the connection with the *Orchestrator*. In fact, in the previous version, the output sent to the *Orchestrator* sometimes presented some errors due to the presence of values not recognised by the *Orchestrator*. Furthermore, connection problems with the *Orchestrator* occurred due to misaligned metrics among *Catalogue*, *DSL Mapper* and *Orchestrator*. These problems were resolved, and an intensive test campaign was carried out to verify that all obligations translated into Rego policies were correctly received and interpreted by the *Orchestrator*. A further improvement that was introduced is the handling of the filtering concept, whereby requirements are currently linked to a specific Cloud Service.

4.2.3.2 Published APIs

The *DSL Mapper* provides a REST API that can be used by the other components interacting with it. The list of the available APIs is provided in *APPENDIX F: Published APIs, Component: NL2CNL Translator and DSL Mapper*.

4.2.3.3 Graphical User Interface

This component does not have a graphical interface and interacts with the other components of the framework via the API.

4.2.3.4 TRL

The TRL of the *DSL Mapper* is 4 at the moment of writing. After the validation phase ends, it is expected to be 5.

4.3 Risk Assessment and Optimisation Framework (block #3)

4.3.1 Risk Assessment and Optimisation Framework (RAOF)²⁸

RAOF is a service for supporting the non-conformity assessment process with a risk-based decision-making capability. This component evaluates the current risk of the CSP, by estimation of the CSP's needs and protection against possible threats. The computed risk value is used to evaluate how far is the CSP from full compliance with the selected certification scheme (and

²⁷ <https://git.code.tecnalia.com/medina/public/dsl-mapper>

²⁸ For historical reasons, we use the terms RAOF and SATRA indistinctly though the document. Strictly speaking, RAOF is the name of the component, while SATRA is the tool which implements this component.

assurance level). Not only does this analysis help to identify which security requirements are missing, but also how risky it is for this CSP if these requirements are not fulfilled. By implementing these functionalities, *RAOF* contributes to two Key Results: KR2 (by providing the risk-aware support to a compliance manager before applying for certification) and KR6 (supporting the MEDINA's auditor, i.e., *Automated Certificate Life-Cycle Manager*, with a risk-based evaluation of detected non-conformities).

RAOF is an Open-Source project with license Apache 2.0 and the source code is available on the public GitLab repository²⁹.

More details about this component are available in deliverables D2.8 [12] and D4.5 [16].

4.3.1.1 Implementation and Integration Status

The *RAOF* component is used in two parts of the MEDINA process. First, the component provides the support during the bootstrapping, when a compliance manager evaluates if the cloud service could be certified (i.e., fulfil the requirements for certification). In this case, the compliance manager interacts with the *RAOF* directly through the GUI.

RAOF is also used during the dynamic evaluation of compliance. The *CCE* component notifies *RAOF* about the requirements which have been evaluated by assessment tools and the result of these assessments. If non-conformities are detected, *RAOF* re-computes the risk using initially provided input and the assessment results and analyses the non-conformity gap. The result of this analysis (i.e., whether the non-conformity is to be counted as major or minor) is provided to the *LCM* for further evaluation of the status of the certificate.

The final version of the component in M33 implements all planned features. The engine for the non-conformity gap analysis engine is set up to compute and compare risk values for different assurance levels and different cloud market types. The computation is based on the cloud resources expected values of which should be initially provided by the CSP and the fulfilled requirements of the certification scheme. Moreover, the recently added functionality helps the compliance manager to optimise its investment in covering certification scheme's requirements to achieve at most minor non-compliance. The dynamic part implements the communication between *Continuous Certification Evaluation* (CCE) and *Life-Cycle Manager* (LCM) components and is set up to perform the risk-based non-conformity gap assessment automatically. Moreover, the latest changes introduced in the operation of the component also provide a quick assessment of the impact of every failed requirement. This information can be used by the CSP to prioritise its effort and focus first on fixing the most significant failures. Small modifications in the logic of the dynamic risk computation have been implemented, to enhance its computation of risks per resource.

The *RAOF* frontend is integrated with the MEDINA *Integrated UI* and implements the common functionalities for it, the user can access *RAOF* by clicking on the left menu option "Risk management" (see Figure 29). In particular, *RAOF* uses the Keycloak mechanism to authenticate users and authorise access to the risk analysis functionalities only for associated Targets of Evaluations. During the third integration round, a more fine-grained authorisation procedure is implemented, allowing only specific roles (e.g., a service owner) to modify the parameters of the analysis. Other roles may only see the results of the analysis.

Another functionality implemented by *RAOF* is importing results of the questionnaire provided by the Catalogue. This option aims to ensure that a user can report which EUCS requirements

²⁹ <https://git.code.tecnalia.com/medina/public/static-risk-assessment-and-optimization-framework>

the considered service satisfies only once, but benefit from the both analyses provided as by the *Catalogue* (compliance score) as well as *RAOF* (risk-based analysis of non-conformities and optimised planning for implementation of additional requirements).

From M27 to M33 the most attention was dedicated to improving the integration with other components and addressing the issues detected during testing the overall framework. The latest changes in *RAOF* and the components communicating with it are considered. The values used inside the framework (and used for computation of risk) have been adjusted to improve the risk computation procedure.

4.3.1.2 Published APIs

RAOF provides a REST API with a number of endpoints. This API is to be used by the compliance manager dashboard during the bootstrapping phase. All Targets of Evaluation (ToEs) managed by the *RAOF* are created and could be modified by the Cluditor using this API. As well, the *CCE* is supposed to invoke *RAOF* using a dedicated endpoint of this API.

The list of the available APIs is provided in *APPENDIX F: Published APIs, Component: Risk Assessment and Optimisation Framework*.

4.3.1.3 Graphical User Interface

RAOF provides a GUI for the direct interaction with a compliance manager (see Figure 29). As for all the other components – in the right upper corner of the GUI there is a “Help” button that opens the User Manual in a new tab. The User Manual is also available in *APPENDIX G: User Manuals*.

This GUI can be used to select (see Figure 30), set up all settings for a Target of Evaluation (see Figure 31), add the list of resources and their sensitivity (see Figure 32), and report fulfilled requirements (see Figure 33). Also, the GUI displays the results of the analysis and the computed risk values, as shown in Figure 34.

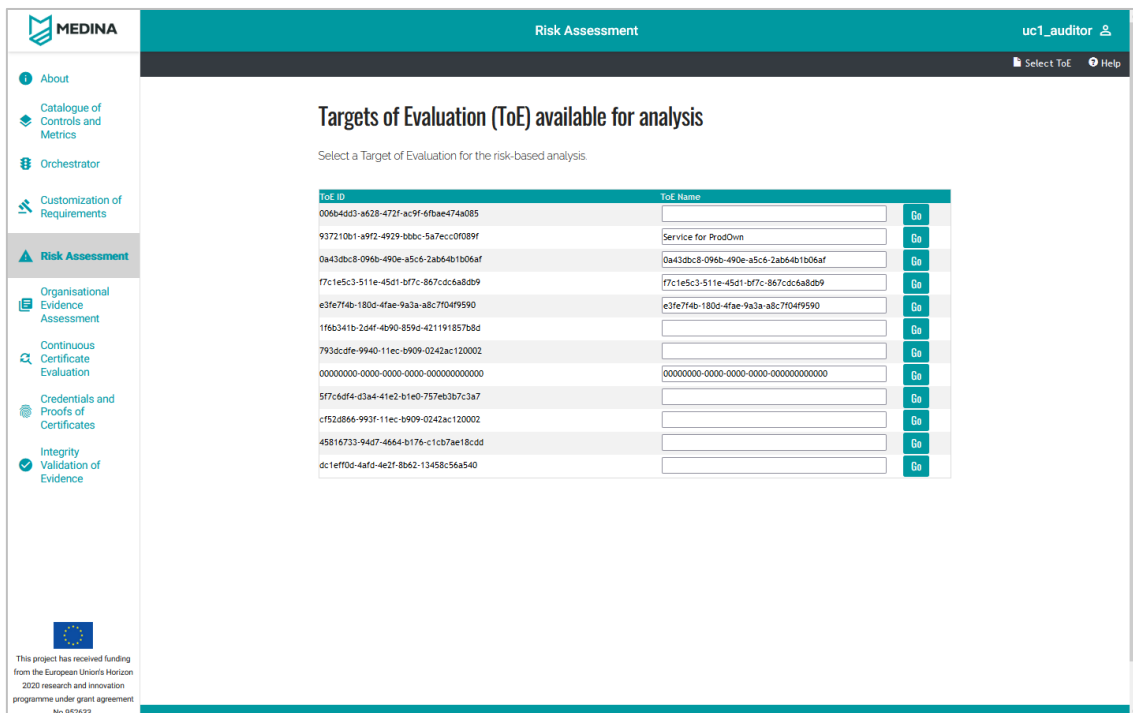


Figure 29. RAOF GUI

Select ToE

Targets of Evaluation (ToE) available for analysis

Select a Target of Evaluation for the risk-based analysis.

ToE ID	ToE Name	
0e37d39c-d3ba-4a24-aeaa-380c41cde64c	TestArt2	Go
1bca421e-c708-11ed-afa1-0242ac120002	First ToE	Go
1e67df1a-2127-421a-ba0f-c8731b5e5d3c	TEST STEFANO	Go
2a9c09a7-ebb7-4a97-835c-ea436c1b38b1		Go
2fea17b3-1298-4f8a-af6d-f80e355438d4		Go
34175106-a188-4c7f-9720-53dc7eeaa490	First ToE	Go
4b60d24a-c6f4-11ed-afa1-0242ac120002	TEST ToE	Go
5b51b1d2-bb00-4512-be37-24819b5d99ab	TestHigh	Go
600e0e76-df6b-11ed-b5ea-0242ac120002	ToE TEST CCE	Go
8cd8c7d0-1446-4cac-ab96-c3f82cd91ab2		Go
90acc728-dfd0-41a7-acd6-0b86500f4568		Go

Figure 30. RAOF - Select ToE

Select ToE

Target of Evaluation Info

Please, provide the cloud service type, select a Certification Scheme and the corresponding Assurance Level (if applicable).

CLOUD SERVICE LAYER

SaaS

CERTIFICATION SCHEME

EUCS

ASSURANCE LEVEL

High

Start Questionnaire

Figure 31. RAOF - Setup of Targets of Evaluation

CLOUD RESOURCE IDENTIFICATION

ID	Cloud Resource	Cloud Resource Type	Number Of Unit	Confidentiality Level	Integrity Level	Availability Level
A1	Insel	IoT Device Provisioning Service	1	6	3	3
A2	Insel	CI CD Service	1	6	6	3
A3	Insel	Function	1	1	4	3
A4	Insert	Database	1	7	3	3
A5	Insert	Virtual Machine	1	2	3	5
A6	Insert	Client trust	1	7	3	5

+ Create row Delete row Submit

Figure 32. RAOF - List of resources

Select ToE

Questionnaire

Please, answer all questions selecting the most suitable answer from the lists of available answers. Then press Submit.

Page 1/20. Organisation Of Information Security

Information Security Management System

OIS-01.1H - The CSP shall have an information security management system (ISMS), covering at least the operational units, locations, people and processes for providing the cloud service, with a valid certification of compliance with the requirements of EN ISO/IEC 27001 or with national schemes based on ISO 27001, issued by an accredited CAB covering the cloud service.

☒ Yes.
☐ Partial
☐ No
☐ Not Applicable

OIS-01.2H - The CSP shall provide documented information of the ISMS applied to the cloud service, including at least: (1) ISO/IEC 27001 requirement 6.1.3 item c) shall be used for the cloud service using the controls in this document for comparison, with the restriction that all controls shall apply. (2) ISO/IEC 27001 requirement 6.1.3 item d) producing a Statement of Applicability referring to the controls in this document for the cloud service

☒ Yes.
☐ Partial
☐ No
☐ Not Applicable

Contact With Authorities And Interest Groups


OIS-03.1H - The CSP shall maintain regular contacts with relevant authorities in terms of information security and relevant technical groups to stay informed about current threats and vulnerabilities.


☒ Yes.
☐ Partial
☐ No
☐ Not Applicable

Information Security In Project Management

OIS-04.1H - The CSP shall perform a risk assessment according to RM-01 to assess and treat the risks on all projects that may affect the provision of the cloud service, regardless of the nature of the project.

☒ Yes.
☐ Partial
☐ No
☐ Not Applicable

 Save and Leave

 Go to Cloud Resource


 Go Forward

Figure 33. RAOF - Requirements to be fulfilled

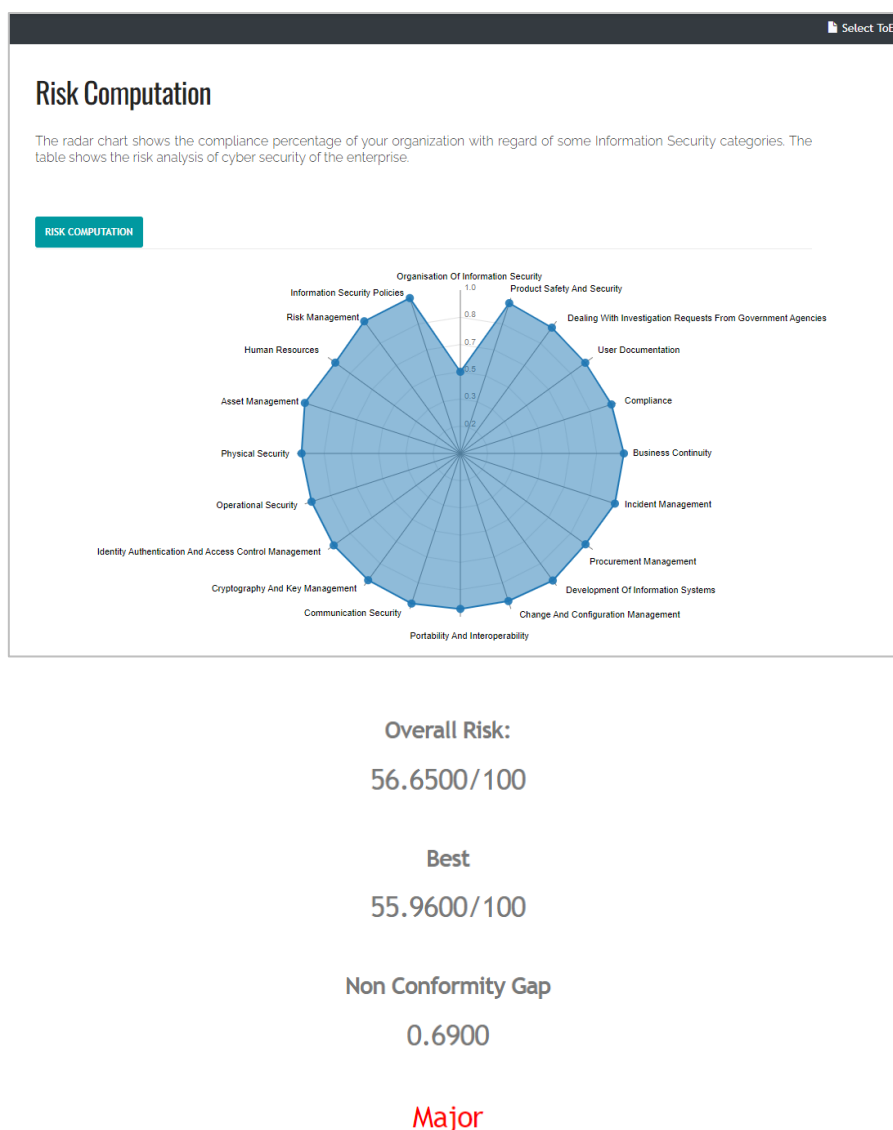


Figure 34. RAOF - Results of the static risk analysis

4.3.1.4 TRL

The TRL of RAOF is 6 at the moment of writing. After addressing the issues identified by the ongoing validation process, it is expected to be changed to 7.

4.4 Continuous Evaluation and Certification Life-Cycle (block #4)

4.4.1 Continuous Certification Evaluation

The *Continuous Certification Evaluation* component (CCE) collects assessment results and builds an evaluation tree representing the aggregated assessment results on higher levels of the certification scheme to determine compliance with the different certification elements (requirements, controls, control groups, etc.).

The components belonging to the “Continuous Certification Evaluation” are mainly related with KR5 (Continuous Cloud Certificate Evaluator, CCE), whose objective is to collect assessment results gathered by Security Assessment components and continuously build an evaluation tree representing aggregation of assessment results to determine compliance with the different controls.

CCE is an Open-Source project with license Apache 2.0 and the source code is available on the public GitLab repository: CCE core³⁰ (back-end) and the UI³¹ (front-end).

Additional details about the component's architecture and methodology used is available in deliverable D4.3 [17].

4.4.1.1 Implementation and Integration Status

As described in D5.4 [2], all the elicited functional requirements are implemented in the CCE. They are implemented using three microservices: CCE core (back-end), CCE UI (front-end) and MongoDB database.

The final version of the component in M27 supports full integrations with the *Catalogue of Controls and Metrics*, *Orchestrator*, *RAOF* and *CNL Editor*.

CCE receives assessment results gathered by the *Security Assessment* components through the *Orchestrator* and continuously builds an evaluation tree representing the aggregation of assessment results to determine compliance with the different certification elements.

Beside the assessment results, CCE also receives data about the Cloud Services and related Targets of Evaluation from the *Orchestrator*. Another required input is the structure of the evaluation scheme used (relations between metrics, requirements, controls, categories) that is obtained from the *Catalogue*. In addition, CCE also provides improved overview of assessed metrics by integrating information (added or modified metrics) from the *CNL Editor*.

Outputs of the CCE are consumed by the *Risk Assessment and Optimisation Framework (RAOF)* and the *Life-Cycle Manager (LCM)*. CCE periodically sends the changed values of the evaluation tree to *RAOF* for the risk-based evaluation of the severity of incompliances. The *LCM* queries the CCE's API to obtain operational effectiveness values which help determine the overall certification state.

The evaluation aggregation is implemented for multiple Targets of Evaluation (multi-tenancy support), history of evaluation tree states is being stored in a database and is exposed through an API, the operational effectiveness values are being calculated and integration with all components needed for the complete functionality is complete.

The CCE frontend is integrated with the *MEDINA Integrated UI*, the user can access it by clicking on the left menu option "Continuous Certificate Evaluation" (see Figure 35),

The following functionalities and features have been implemented and integrated between M27-M33:

- Full integration with Keycloak enabling filtering (AuthT/AuthZ) of Cloud services for different users
- New Button to show/hide yellow nodes for better visualisation of CCE tree results
- The non-compliant (red) requirements contain a link to the Implementation guidelines/Reference TOMs from the *Catalogue*
- An aggregated compliance view has been implemented for users with access to more than one Cloud Service
- Minor UI/UX improvements (e.g., unified look with the MEDINA IUI, human-friendly description of the Target of Evaluation).

³⁰ <https://git.code.tecnalia.com/medina/public/continuous-certification-evaluation>

³¹ <https://git.code.tecnalia.com/medina/public/cce-frontend>

Based on the feedback from validation, we have started to develop the following features which will be finished and reported in D6.4 [10]:

- Obtain detailed metrics data from *CNL Editor* and visualise them in the *CCE* evaluation tree.
- Highlight compliance changes between two “consecutive” *CCE* trees.
- Display *RAOF* results in the *CCE* tree.

The source code of both the *CCE* core³² (back-end) and the UI³³ (front-end) is available on the public GitLab repository. Dockerfiles are available for simple deployment and are integrated with the project’s development and testing environments on Kubernetes.

4.4.1.2 Published APIs

CCE exposes two APIs:

- HTTP REST-like API, mainly used for the communication with the web front-end (UI).
- gRPC API, for the communication with the *Orchestrator* and the *Life-Cycle Manager*.

The list of the available APIs is provided in *APPENDIX F: Published APIs, Component: Continuous Certification Evaluation*.

4.4.1.3 Graphical User Interface

The CCE frontend provides a tree visualization of the assessment results (see Figure 35).

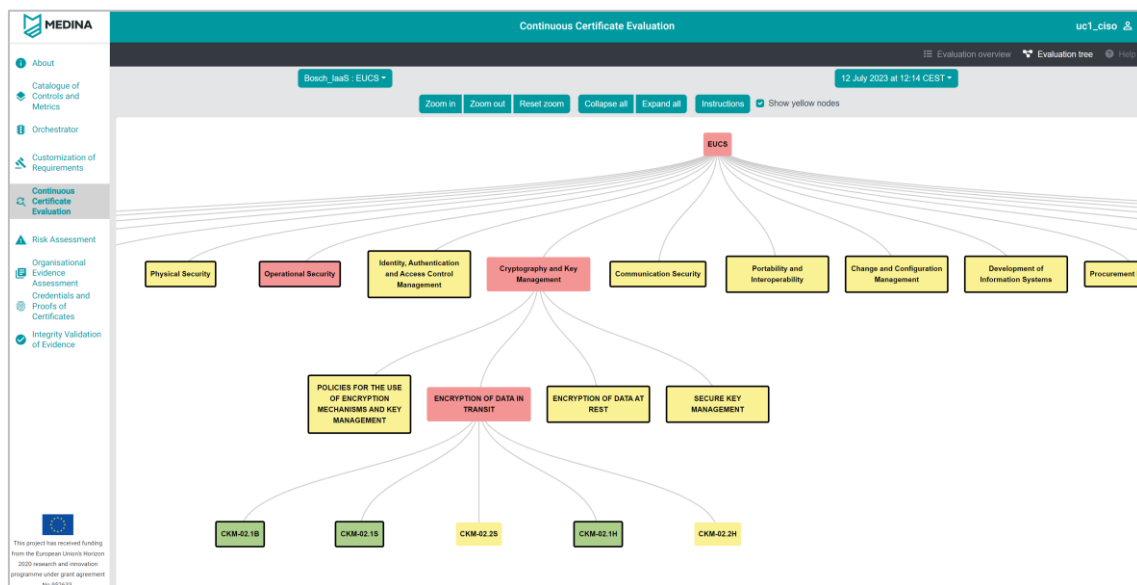


Figure 35. CCE GUI - Evaluation tree

As for all the other components – in the right upper corner of the graphical interface there is a “Help” button that opens the User Manual in a new tab. The User Manual is also available in *APPENDIX G: User Manuals*.

³² <https://git.code.tecnalia.com/medina/public/continuous-certification-evaluation>

³³ <https://git.code.tecnalia.com/medina/public/cce-frontend>

4.4.1.4 TRL

The TRL of the CCE is 4 at the moment of writing. After the validation phase ends, it is expected to be 5.

4.4.2 Automated Certificate Life Cycle Manager

The *Automated Certificate Life Cycle Manager (LCM)* integrates data from multiple other components to decide about the certificate's status. It addresses KR5 (Continuous Cloud Certificate Evaluator).

Changing a certificate state is a complex decision that is difficult to do meaningfully in an automated way. For that reason, the *LCM* combines information about a cloud service's risk level, its compliance status over time, as well as other information to make this decision. In the context of the SSI system, the decision can be checked by a human auditor before it is published to ensure that the state is valid.

The *LCM* is Open Source with license Apache 2.0 and the source code is available on the public GitLab repository³⁴.

More details about this component are available in deliverable D4.3 [17].

4.4.2.1 Implementation and Integration Status

The prototype of the *LCM* has been designed and developed in the months M1-M30 of the project. It implements the certificate states defined by the EUCS.

To make its deployment and maintenance easier, the *LCM* has been integrated with the *Orchestrator* to reuse its database and its user interface. This way, certificate information is stored in the *Orchestrator* database rather than requiring a dedicated database for the *LCM*. Also, it reuses the *Orchestrator* UI to present certificate information to end users.

In the project period from M27-M33, the *LCM* has been advanced with UI improvements and minor API updates. The main focus in this period was on testing its interaction with other components (*Orchestrator*, *CCE*, *RAOF*, *SSI Framework*) and deliver bug fixes accordingly.

The *LCM* is furthermore integrated with three other components to manage certificates. First, it receives evaluation results from the *RAOF*. The *RAOF* performs risk assessments of cloud services and forwards the results to the *LCM* which integrates them into its decision-making process for the respective certificate(s).

Second, the *LCM* retrieves statistical data about the historical compliance of a certificate of a cloud service, i.e., a Target of Evaluation, from the *CCE*. The *CCE* calculates statistics about the compliance status of a Target of Evaluation over time, such as the ratio of non-compliant to compliant times. This presents a second source of data for certificate state decisions.

Third, the *LCM* is integrated with the *SSI Framework*. After having changed a certificate status, or after the creation of a new certificate, it forwards the new state to the *SSI Framework* (see section 4.4.3).

³⁴ <https://git.code.tecnalia.com/medina/public/life-cycle-manager>

4.4.2.2 Published APIs

The *LCM* exposes several APIs that allow the management of certificates, including their creation, modification, deletion, etc. The APIs are described in detail in D4.3 [18]. They are also available in *APPENDIX F: Published APIs, Component: Life Cycle Manager*.

4.4.2.3 Graphical User Interface

The *LCM* is integrated with the *Orchestrator* to provide database capabilities and to visualize them in a user interface. It thus does not provide a dedicated GUI but reuses the *Orchestrator's* GUI to show information about existing certificates. The GUI also presents data about certificates' state histories, i.e., which states they have had in the past and due to which reason, e.g., a suspended state due to a major deviation.

Figure 36 shows an example of how a certificate is displayed in the GUI along with its state history.

"Bosch_iaaS"			
ID: 2111 Name: Bosch_iaaS Service ID: 945d9c38-b2ad-4db5-9d33-cd10b7d5d840 Issue Date: 2023-03-27T10:06:55Z Expiration Date: 2024-03-27T10:06:54Z Schema: EUCS Assurance Level: high CAB: CAB123 Description: Bosch IaaS			
State History			
State	Deviation	Timestamp	Tree ID
new		28 Jun 23 10:00 UTC	123456
suspended	major	30 Jun 23 08:01 UTC	223456
continued	minor	30 Jun 23 08:16 UTC	234567

Figure 36. An example of a certificate as displayed in the Certificates view in the Orchestrator GUI

4.4.2.4 TRL

The TRL of the *Automated Certificate Life Cycle Manager* is 4 at the moment of writing. After the validation phase ends, it is expected to be 5.

4.4.3 Automated Self-Sovereign Identity-based certificates management

The *Self-Sovereign Identity (SSI) Framework* provides the CSPs with the capability to manage their own security certificates as part of their identity through verifiable credentials. "To manage their own identity" ultimately means that they store their identity on their own "user space" without intervention of a third-party.

The *SSI Framework* is not only composed of the *CSP component* to store and control the credentials. It is also composed of the *Issuer component* which provides the CAB a way to issue verifiable credentials about the security certificates related to the CSP; and the *Client component* which provides a way to ask and verify proofs of different security certificate features. In this sense, privacy is an important requirement within MEDINA, as several security certificate features are considered sensitive and must be treated carefully. The *SSI Framework* is capable of sharing sensitive information in a confidential way by keeping the user's identity out of third

parties, which act as identity silos, reducing the risk of identity theft; but also, by using Zero-Knowledge Proofs (ZKPs). ZKPs preserve user's privacy using cryptography to proof that a CSP has some attributes without disclosing these attributes.

The *SSI Framework* is part of KR5 (Cloud Certificate Evaluator). It has a Proprietary license, Copyright by TECNALIA.

Details about this component are available in deliverable D4.3 [18].

4.4.3.1 Implementation and Integration Status

A complete prototype of the *SSI Framework* was implemented by M30. It is composed by one SSI-network, three SSI-agents (issuer, holder, and verifier, for the complete SSI flow), one SSI-API (for receiving information from the LCM), and two SSI-webapps (one for the holder and another one for the issuer and verifier).

The SSI-network, two of the SSI-agents (issuer and verifier), one of the SSI-webapps (the one for the issuer and verifier) and the SSI-API are provided as a service by TECNALIA emulating the CAB and a potential CSP customer. All these components are correctly deployed and integrated with each other. Additionally, the SSI-API is also correctly integrated with the LCM for receiving the certificate state after the MEDINA framework execution.

Additionally, one SSI-agent (holder) and one SSI-webapp (the one for the holder) are correctly deployed on the MEDINA environment and are correctly integrated with the Keycloak instance of MEDINA and the MEDINA *Integrated UI*. The user can access them by clicking on the left menu option "Credentials and Proofs of Certificates" (see Figure 37). These components are also correctly integrated with the rest of the SSI components deployed at TECNALIA. No integration with additional components is needed in this case.

The main updates of the final release of the *SSI Framework* (M33) with respect to the previous version (M27) are related to:

- GUI has been adapted to the visual guidelines (CSS based).
- GUI has been improved for easier navigation.
- Integration of SSI-agent and SSI-webapp of the holder (CSP) with the *Integrated UI* and the keycloak instance of MEDINA.
- Role-based authorization rules defined in Table 25 have been applied.
- Zero-Knowledge Proofs (from SSI) applicability in security certification.

The associated functional requirements are fully covered.

4.4.3.2 Published APIs

The SSI-API component of the *SSI Framework* exposes an API described in detail in Section 6.3.1.2.3 in D4.3 [18] The list of these APIs is also available in Annex F, *Component: Automated Self-Sovereign Identity-based certificates management (SSI)*.

4.4.3.3 Graphical User Interface

The *SSI Framework* is controlled by means of a web-app application:

- Figure 37 shows the home page of the graphical interface for the CSP (holder) integrated in the MEDINA *Integrated UI*.

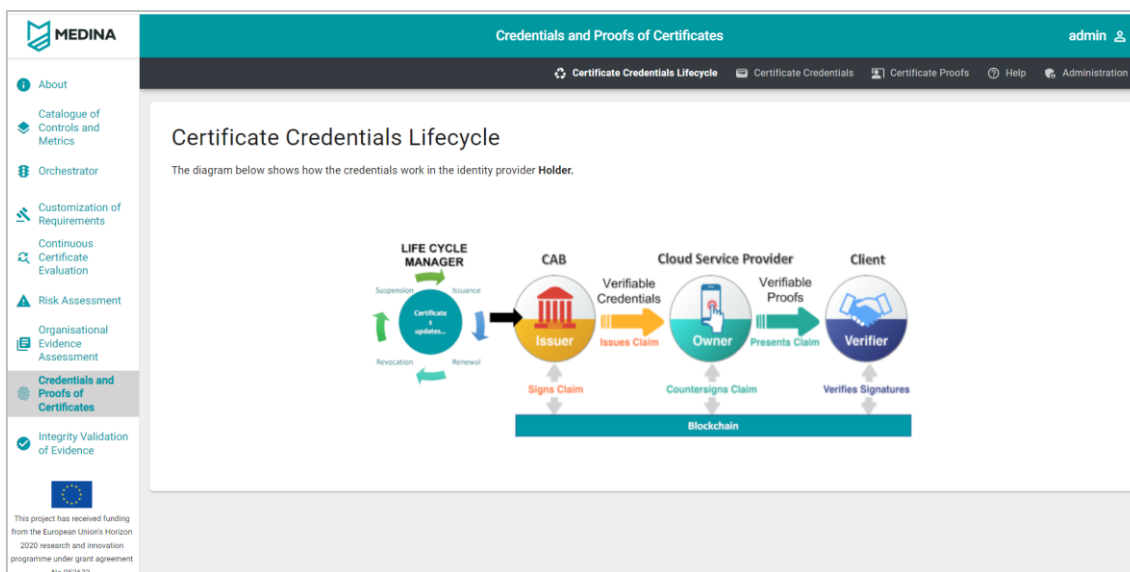


Figure 37. SSI Framework GUI for CSPs

- Figure 38 shows an example of the webapp for the CAB (issuer), showing the issuance of new credentials.

Figure 38. SSI Framework GUI for the CAB

- Figure 39 shows an example of the webapp for the CSP customers (verifier), showing the request of new proofs.

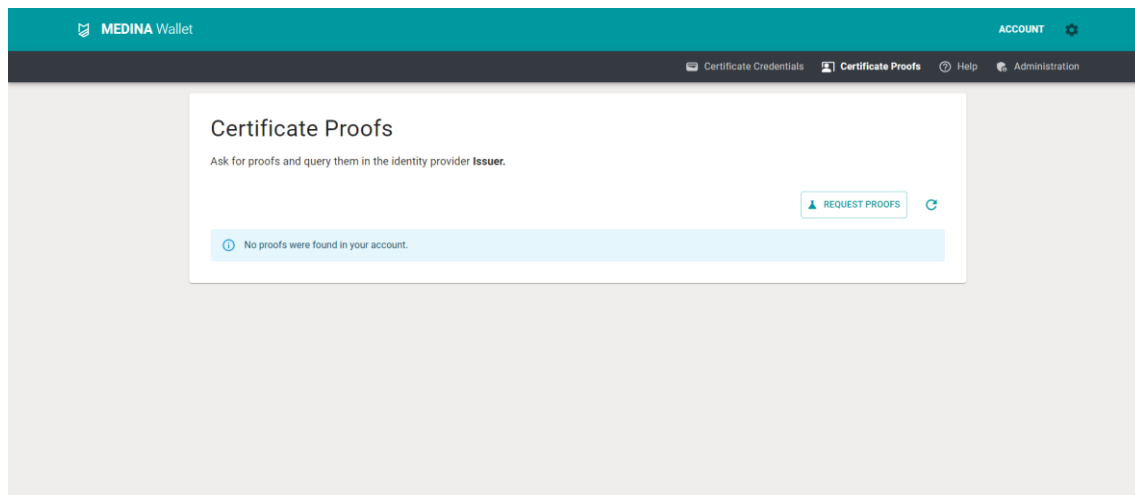


Figure 39. SSI Framework GUI for the CSP customers

More details about the graphical interfaces of the *SSI Framework* can be found in Appendix E: SSI Webapp Manual in D4.3 [18].

As for all the other components – in the right upper corner of the graphical interface there is a “Help” button that opens the User Manual in a new tab. The User Manual is also available in *APPENDIX G: User Manuals*.

4.4.3.4 TRL

The TRL of the *Self-Sovereign Identity (SSI) Framework* is 4 at the moment of writing. After the validation phase ends, it is expected to be 5.

4.5 Organizational Evidence Gathering and Processing (block #5)

4.5.1 Assessment and Management of Organizational Evidence

The *Assessment and Management of Organizational Evidence (AMOE)* component extracts and collects evidence from policy documents. The component is addressing the NLP and organizational measure aspects of KR4 (Continuous Evidence Management Tools). It can compute pre-assessments (hints) that can be used to speed up the audit process. After uploading a document, the component extracts the evidence for a set of organizational metrics with the help of the built-in Natural Language Processing (NLP) pipeline.

The processed data can be analysed in the UI and assessment results can be set/confirmed. Once complete, the assessment results can be forwarded to the *Orchestrator* on demand.

The *AMOE* component is licensed under Apache 2.0 and the source code is available on the public GitLab repository³⁵.

Additional details about this component are available in deliverables D3.3 [19] and D3.6 [20].

4.5.1.1 Implementation and Integration Status

For the evidence gathering functionality the following subprocesses have been implemented. Pre-processing for PDF to transform unstructured policy documents into semi-structured content usable for faster and more accurate extraction. The evidence extraction pipeline itself,

³⁵<https://git.code.tecnalia.com/medina/public/amoee>

which consists of one main method (keyword-based approach) that is used by default. For research purposes three other similar evidence extraction pipelines have been built, however, tests have shown they would need additional work. All evidence extraction approaches make use of standard NLP techniques and utilize the pre-trained question answering system roberta-base-squad2³⁶.

The integration of the component into the MEDINA framework uses the API of the *Catalogue of Controls and Metrics* and has a hardcoded fall back to a static metric file if the connection would fail. Furthermore, the connection to the *Orchestrator* for metric implementation details and sending assessment results and evidence has been implemented.

To store the metadata, logging and extracted evidence internally, a connector to internal data base (MongoDB) has been added. The user action information (on edit/upload/delete/submit) is logged into the data base.

The AMOE frontend is integrated with the MEDINA *Integrated UI*, the user can access it by clicking on the left menu option “Organizational Evidence Assessment” (see Figure 40). User authentication is done via the MEDINA Keycloak service and respective component client. Role based access (Keycloak roles) as well as filtering of information based on cloud service information in the authentication token has been implemented. A dockerfile and kubernetes configuration for deployment of webservice, db and redis cache have been created.

A quality check pipeline for manual checks on the status and aid of research tasks for evidence extraction has been implemented. It enables comparison of annotated information in the tool Inception³⁷ to the evidence extraction approaches.

The main updates of the final release of AMOE (M33) with respect to the previous version (M27) are as follows:

- The GUI has been adapted to the MEDINA *Integrated UI* visual guidelines (CSS based).
- A help button has been added linking to the user manual.
- Buttons have been added to the navigation bar, as well as some additional navigation in the evidence view.
- The functionality to submit multiple metrics at once has been added.

4.5.1.2 Published APIs

The AMOE APIs are listed in *APPENDIX F: Published APIs, Component: Assessment and Management of Organizational Evidence – AMOE*.

4.5.1.3 Graphical User Interface

AMOE provides a GUI for users to interact (see Figure 40). The following access types are defined, configurable through the Keycloak authentication token roles:

- no access
- read only access
- upload/delete files or stop running processes
- edit/submit assessment results
- admin (full read/write access)

³⁶ <https://huggingface.co/deepset/roberta-base-squad2>

³⁷ <https://inception-project.github.io/>

As for all the other components – in the right upper corner of *AMOE* there is a “Help” button that opens the User Manual in a new tab. The User Manual is also available in *APPENDIX G: User Manuals*.

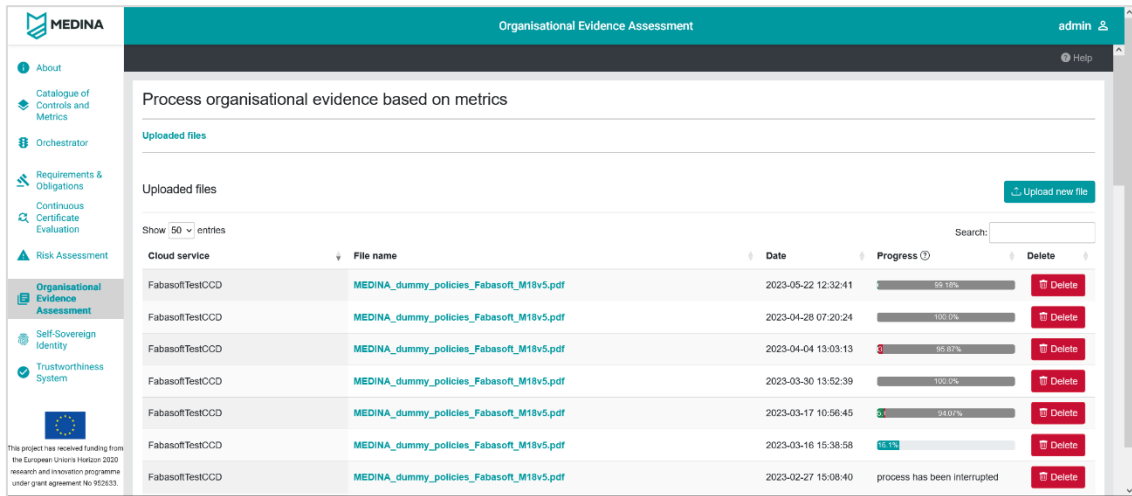


Figure 40. AMOE landing page

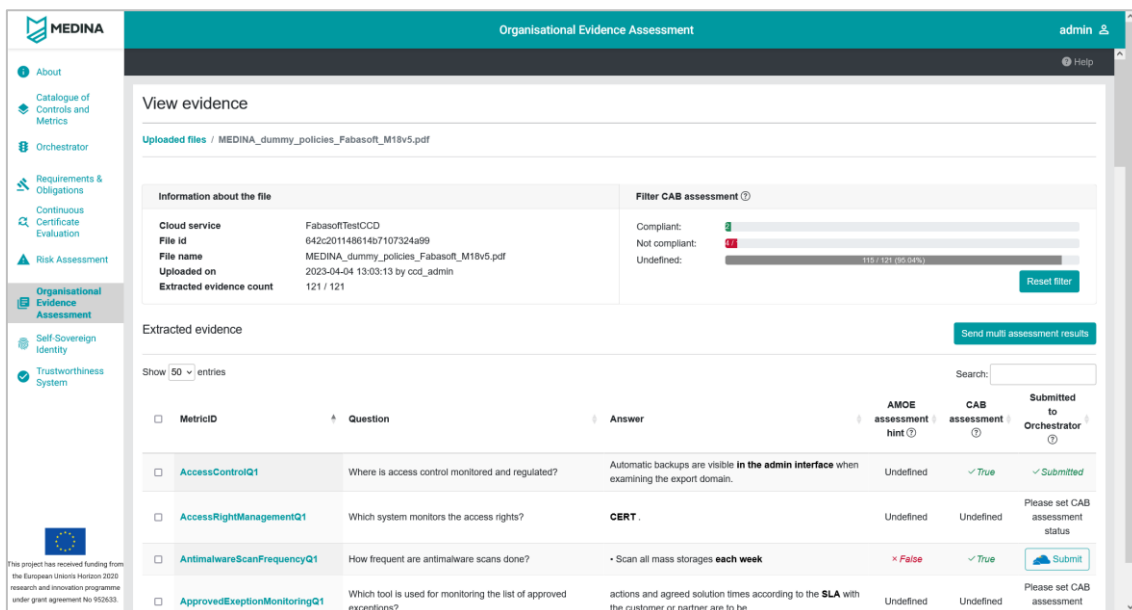


Figure 41 AMOE file overview

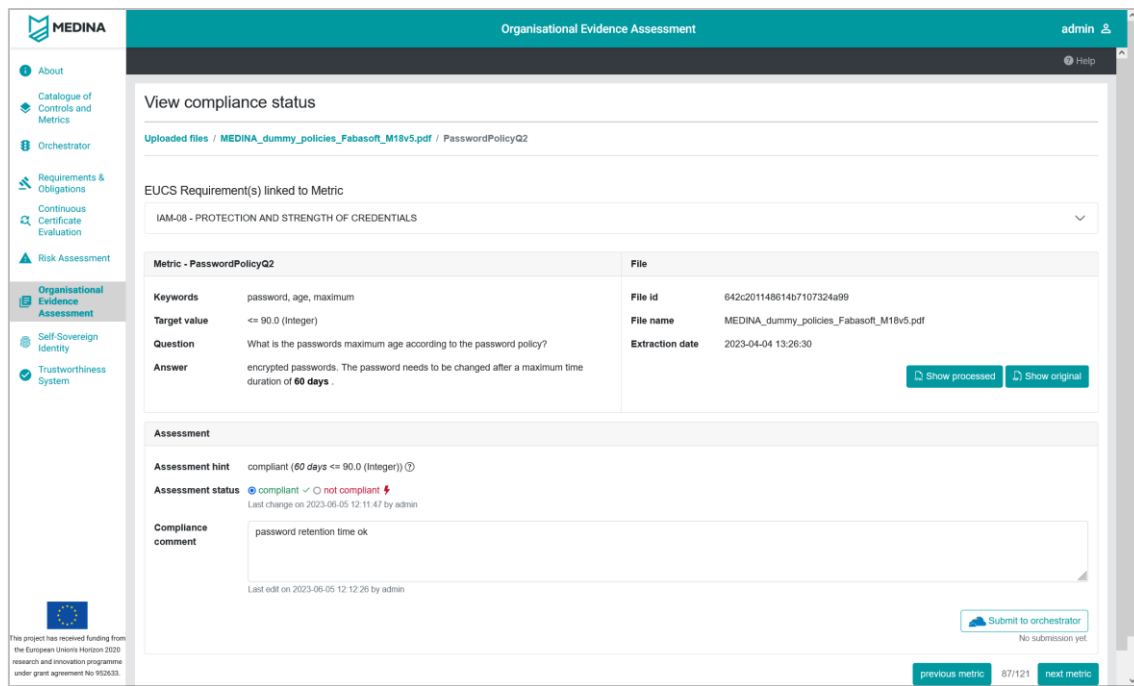


Figure 42 AMOE compliance status view

4.5.1.4 TRL

The TRL of *Assessment and Management of Organizational Evidence* (AMOE) is 4 at the moment of writing. After the validation phase ends, it is expected to be 5³⁸.

4.6 Orchestrator and Databases (block #6)

4.6.1 Orchestrator and Databases

The *Orchestrator* is a central component of the MEDINA framework which processes all evidence and assessment results. It receives these data from the security assessment tools, and forwards them to the appropriate components, such as the CCE. Furthermore, it provides a database that stores evidence and assessment results, as well as metrics, and other data.

Additionally, the *Orchestrator* provides users with the possibility to create new cloud services and Targets of Evaluation, and it propagates this data to other components such as the CCE and the RAOF. This way, components in the MEDINA framework are up-to-date about which cloud services should be assessed for which certification schema, and they receive the appropriate data to do so.

Overall, the *Orchestrator* is the central management component in MEDINA that is integrated with many components and provides many APIs for the management of evidence, assessment results, cloud services, metrics, certificates, etc.

The *Orchestrator* component is licensed under Apache 2.0 and the source code is available on the public GitLab repository³⁹.

³⁸ AMOE is validated by the Bosch use case via UI and in the Fabasoft use case via the Company Compliance Dashboard (CCD) utilizing the API AMOE provides.

³⁹ <https://git.code.tecnalia.com/medina/public>

Additional details about this component are available in deliverables D3.3 [19] and D3.6 [20].

4.6.1.1 Implementation and Integration Status

The *Orchestrator* has been designed and developed up until M30 of the project. Main updates of its final release in M33 comprise extended authorization features for showing the certificate status: first, a filtering for certificates based on the user's claims has been implemented, i.e., only showing the certificates the user is allowed to manage, and second, a dedicated certificate view has been implemented that is publicly available and shows reduced, basic information about existing certificates.

The *Orchestrator* is integrated with numerous other components, including *Cloud evidence collector*, *Security assessment*, *DSL Mapper*, *Continuous Certification Evaluation*, *Risk assessment and optimisation framework*, *Life Cycle Manager*, and *Catalogue of controls and metrics*.

The *Orchestrator* frontend is integrated with the MEDINA *Integrated UI*, the user can access it by clicking on the left menu option “Orchestrator” (see Figure 43). The *Orchestrator* is also integrated with the user management tool (Keycloak), so it is able to control the logged user and its properties, specifically the role.

4.6.1.2 Published APIs

The *Orchestrator* exposes numerous APIs which are described in more detail in D3.3 [19]. The list is also available in *APPENDIX F: Published APIs, Component: Orchestrator*

4.6.1.3 Graphical User Interface

The *Orchestrator's* graphical interface comprises multiple views for cloud services (see Figure 43), their resources, assessment results, and more information, as well as for metrics, and certificates.

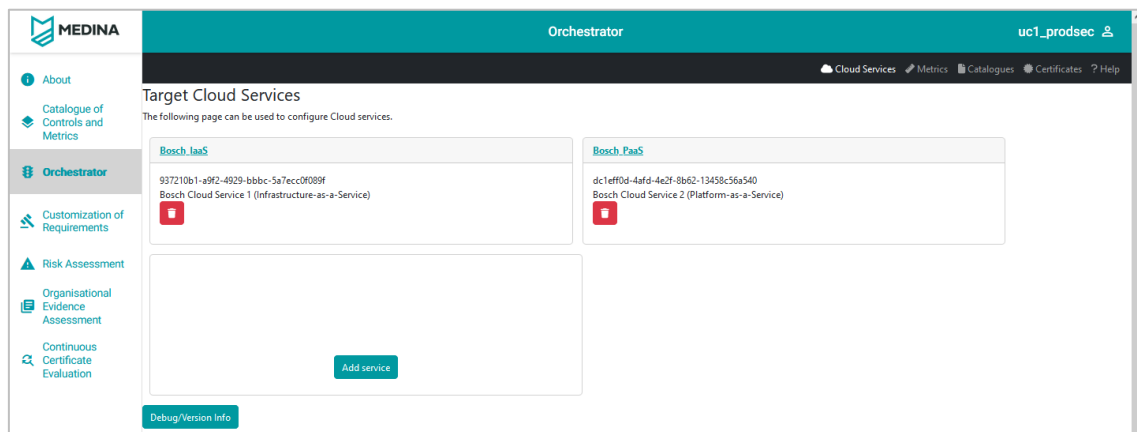


Figure 43. Orchestrator GUI

For reasons of brevity, we only show a selected set of screenshots of the *Orchestrator's* GUI in the following. Figure 44, for example, shows how a cloud service is presented to the user, Figure 45 shows how the assessment results that pertain to a certain cloud service are presented, while Figure 46 shows how the configured metrics are presented.

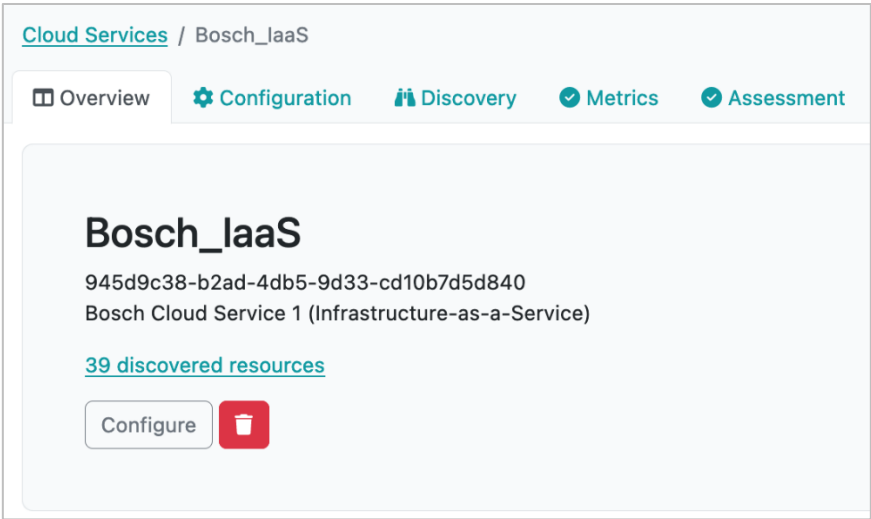


Figure 44. The view of a single cloud service: It shows the service's name, ID, description, and other information. The tabs at the top allow to configure the cloud service, review its discovered resources, review its metrics, and its assessment results.

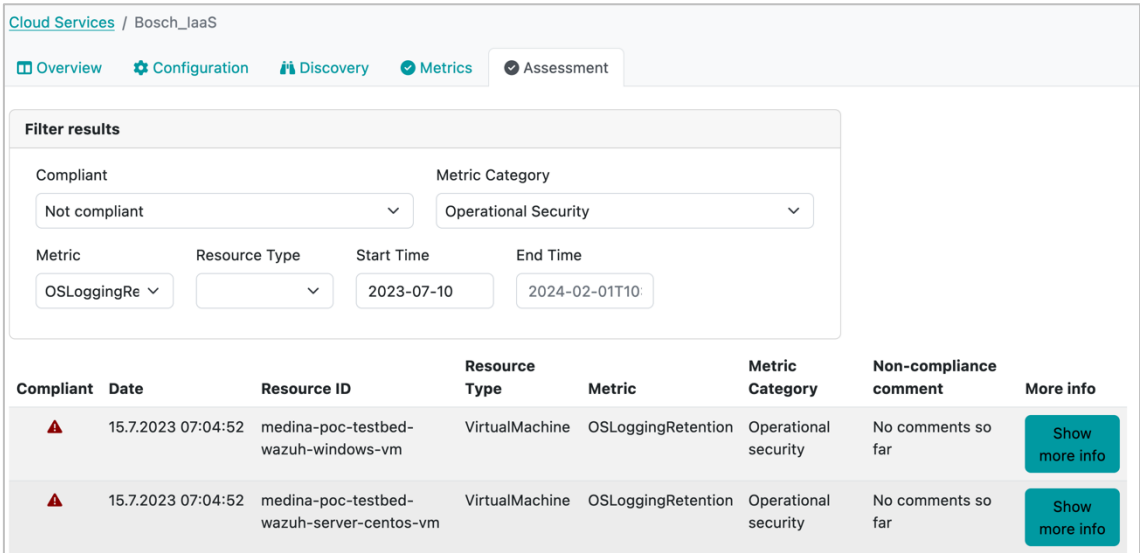


Figure 45. The Assessment view of a cloud service: It allows to filter existing assessment results for the selected cloud service for different parameters like resource type and timeframe. Also, the "Show more info" button reveals details about an assessment result's information



Figure 46. The Metrics view: In this view, configured metrics can be reviewed, including their Rego code that possibly has been generated by the DSL mapper

As for all the other components – in the right upper corner of the graphical interface there is a “Help” button that opens the User Manual in a new tab. The User Manual is also available in *APPENDIX G: User Manuals*.

4.6.1.4 TRL

The TRL of the *Orchestrator* is 4 at the moment of writing. After the validation phase, it is expected to be 5.

4.6.2 Trustworthiness System

The *MEDINA Evidence Trustworthiness Management System* provides a secure storage for evidence and assessment results hashes. It is implemented through Smart Contracts backboneed by a common Blockchain network for all the MEDINA framework instances, providing the following functionalities:

- It includes the logic for all *Orchestrator* instances in MEDINA to provide the required information to be audited (about evidence and assessment results). For this purpose, an API is exposed by the Blockchain client.
- It provides secure long-term information recording, thanks to the inherent advantages of Blockchain (integrity, decentralization, authenticity...):
 - It provides a record of information on a verifiable way (verification).
 - It provides a record of information on a permanent way (traceability).
 - It guarantees resistance to modification of stored data (integrity).
- It includes the logic for external users to access MEDINA's audited information (about evidence and assessment results) in a graphical and user-friendly way through a kibana⁴⁰-based dashboard.
- It includes the logic for automatic verification of hashes from currently recorded information on the *Orchestrator* with hashes recorded on the Blockchain.

The *MEDINA Evidence Trustworthiness Management System* is part of the KR4 (Continuous Evidence Management tools). It has a Proprietary license, Copyright by TECNALIA.

More details about this component are available in D3.3 [19].

4.6.2.1 Implementation and Integration Status

The *MEDINA Evidence Trustworthiness Management System*, which was completely implemented, deployed, and integrated in M30, is composed by five main components:

- Blockchain client, to be executed on the *Orchestrator* for providing the information (evidence/assessment results) to be saved on the Blockchain. The Blockchain client received the information directly from the *Orchestrator*.
- Smart contract, deployed on Blockchain nodes, for information (evidence/assessment results) writing and reading operations as well as events generation indicating the provision of new information.
- Viewer tool, for subscription to the Blockchain based events and notification to the different viewer clients.
- Graphical viewer client, for gathering and showing all the information saved on the Blockchain (and be able to manually verify it, without needing any interaction with the Blockchain). It has been integrated with the keycloak instance of MEDINA.
- Automatic verification service, for evidence and assessment results integrity automatic check. It automatically interacts with the *Orchestrator* for obtaining the currently recorded evidence/assessment results, and with the Blockchain client for obtaining the associated hashes recorded on the Blockchain. This service has been integrated with the

⁴⁰ <https://www.elastic.co/kibana>

MEDINA *Integrated UI* as well as with the keycloak instance of MEDINA. The user can access it by clicking on the left menu option “Integrity Validation of Evidence” (see Figure 48).

The main updates of the final release of the *MEDINA Evidence Trustworthiness Management System* (M33) with respect to the previous version (M27) are related to:

- Automatic verification service tool design, implementation, and integration.
- GUI has been adapted to the visual guidelines (CSS based).
- GUI has been improved for easier navigation.
- Integration of the automatic verification service and the graphical viewer with keycloak instance of MEDINA.
- Role-based authorization rules defined in Table 27 have been applied.

In the final version of the *MEDINA Evidence Trustworthiness Management System*, 100% of the requirements defined in deliverable D5.2 [3] are fully implemented. One requirement that was partially fulfilled is now fully fulfilled. Concretely, requirement ETM.03 has been extended providing user-friendly manual (through the Blockchain viewer) and an automatic way (through the automatic verification service) to verify the evidence and assessment results integrity.

4.6.2.2 Published APIs


The Blockchain client exposes an API described in detail in Appendix C of D3.3 [19]. The list of APIs is also available in *APPENDIX F: Published APIs, Component: Trustworthiness System*.

4.6.2.3 Graphical User Interface

The *MEDINA Evidence Trustworthiness Management System* includes two different graphical interfaces:

1. The graphical viewer client exposes a Kibana-based graphical interface available at: <https://kibana.medina.bclab.dev/> [authentication required]. For more details, refer to Section 5.2.2.5 in D3.3 [19]. Figure 47 shows an example of dashboard of the graphical interface.

MEDINA Evidence Trustworthiness Management System



Number of pieces of evidence
3,458

Number of assessment results
1,608

Evidences

Look for a specific...

EVIDENCE ID
Select...

EVIDENCE HASH
Select...

RESOURCE
Select...

EVIDENCE LABEL
Select...

LOP
Select...

Apply changes Cancel changes Clear form

Evidence Id	Evidence Hash	Resource	Evidence Collector	CSP	Orchestrator timestamp	Blockchain timestamp per day
16de7d36-e89f-4347-8d44-245c76f9da70	ca83f377588407a50f1d19a5742d8df9cb603a3ef76f343...	Resource for MEDINA workflow tests	Fabasoft Cloud Service Assessment Tool	N/A	1686730429073910450	2023-06-14
105f600a-3be3-42f0-9eb8-937c9d08c6eb	44fc32f500c7912a64713f4e14acd2c9f14d1c3785184f0...	Resource for MEDINA workflow tests	Fabasoft Cloud Service Assessment Tool	N/A	1686730414562766220	2023-06-14
159c0216-149f-4794-9e43-0b8d5deef790	443c50c16811aca5a64828d193e8a25d393b4a042de77...	Resource for MEDINA workflow tests	Fabasoft Cloud Service Assessment Tool	N/A	1686730412521761029	2023-06-14
1a771cec-d9d2-4f03-a484-e55e98693033	d5a63dc34dfb89b61bc6d5d3d80c9283efaa9d48d074e8...	Resource for MEDINA workflow tests	Fabasoft Cloud Service Assessment Tool	N/A	1686730409455936503	2023-06-14
0beaf068-adfd-4fcb-933a-997b6f43568c	8bd81c38c35b62658f39cca79cfe2097c62a9060ee05b5...	Resource for MEDINA workflow tests	Fabasoft Cloud Service Assessment Tool	N/A	1686730394046231131	2023-06-14
189da950-709e-4ce6-9d72-2f84dec63086	809a019b0ec183cc1a55514542d0ecdcd9c83aabdcd0...	Resource for MEDINA workflow tests	Fabasoft Cloud Service Assessment Tool	N/A	168673038868654537	2023-06-14

Assessment Results

Look for a specific...

Assessment Result Id
Select...

Metric Id
Select...

Assessment Result Hash
Select...

Associated Evidence
Select...

Apply changes Cancel changes Clear form

Figure 47. MEDINA Evidence Trustworthiness Management System GUI

2. The automatic verification service includes a frontend that has been integrated in the MEDINA *Integrated UI*. For more details, refer to Section 5.2.2.6 in D3.3 [19]. The main functionalities are:

- Automatic verification status of the complete list of recorded evidence in the *Orchestrator*, as shown in Figure 48 for the filtering options, and in Figure 49 for the validation check. Assessment results share similar graphical interfaces.

MEDINA Evidence Trustworthiness System

Please, provide the **optional filters** for the integrity check of evidence recorded on the **MEDINA Orchestrator** against those recorded on the **MEDINA Trustworthiness System**

Cloud Service ID:

Tool ID:

Figure 48. MEDINA Evidence Trustworthiness Management System - Automatic verification service filters for recorded evidence

MEDINA Evidence Trustworthiness System

This is the current integrity check status of the **MEDINA evidence**

Evidence ID	Integrity Check
79f6d84d-1686-4605-b7d5-f1c789b74b6e	✓
176ab185-0a9f-434d-85df-3c0f2cd1a708	✓
ddc76419-0fbd-4daa-b9f5-a0bd5113b82c	✗
61da7d64-1143-460a-ba75-38f695641212	✓
156f5a1b-a006-4cc9-87c2-6eb72c2e1e5f	✓
19a9f11e-3c14-4973-b788-8bf9f30699	✓
b199daba-3645-4a69-a831-593c840d7101	✓
71fab566-37ec-49af-812b-038caa893925	✓
28eab3c5-5db2-4bb2-8d14-a894df145a50	✓
7925efef-cb32-49fe-ae07-32912e1ab8bf	✓
432a6c71-1dcf-4036-bd10-39d7b87e1c0e	✓
6b7f838e-3c34-49ac-a3df-de564d10ae46	✓
137db061-254f-4f87-a816-f26306a64b97	✓

Figure 49. MEDINA Evidence Trustworthiness Management System - Automatic verification service results for recorded evidence

- Automatic verification status of a specific evidence id, as shown in Figure 50 for the filtering options and in Figure 51 for the validation check. Assessment results share similar graphical interfaces.

Figure 50. MEDINA Evidence Trustworthiness Management System - Automatic verification service filter for specific evidence

These are the MEDINA evidence hashes:	
Evidence ID	ddc76419-0fbd-4daa-b9f5-a0bd5113b82c
Orchestrator Hash	e6a68d20508f138b205e9a832a8fcd1eeb388cc3ddc1e089a8e5b1692abe35ea
Blockchain Hash	Not found

Figure 51. MEDINA Evidence Trustworthiness Management System - Automatic verification service result for specific evidence

As for all the other components – in the right upper corner of the graphical interface there is a “Help” button that opens the User Manual in a new tab. The User Manual is also available in *APPENDIX G: User Manuals*.

4.6.2.4 TRL

The TRL of the *MEDINA Evidence Trustworthiness Management System* is 4 at the moment of writing. After the validation phase ends, it is expected to be 5.

4.7 Evidence Collection and Security Assessment (block #7)

4.7.1 Evidence Collection

Evidence collectors are the first automated step in the MEDINA evidence pipeline. They scan a certain resource and compile information about it to be assessed by the *Security Assessment* (see section 4.7.1.4).

4.7.1.1 Cloud Evidence Collector (Cloudditor Discovery)

The *Cloud Evidence Collector* provided by *Cloudditor* discovers existing cloud resources, e.g., from Microsoft Azure systems, and retrieves information about them. It then creates MEDINA evidence and sends it to the *Security Assessment*. This component addresses KR4 (Continuous Evidence Management Tools).

The *Cloud Evidence Collector* is now Open Source with license Apache 2.0 and the source code is available on the public GitLab repository⁴¹.

For more details, please refer to deliverables D3.3 [19] and D3.6 [16].

4.7.1.1.1 Implementation and Integration Status

The *Cloudditor Evidence Collector* implements all requirements defined in D5.2 [2]. Still, for the final version in M33 it has been extended with an ontology mapping, i.e., the resource properties that are discovered are enhanced with a mapping to a cloud resource ontology. For example, virtual machine's properties are extended with a mapping to the ontology concepts "computing" and "virtual machine". This approach allows to define metrics independently from the cloud provider and certification catalogue. It is furthermore integrated with the *Security Assessment* to which it sends the evidence, as well as with the *Orchestrator* which receives the raw evidence to be stored in a database.

4.7.1.1.2 Published APIs

The *Cloudditor Evidence Collector* offers two APIs: one for starting the discovery, and one for retrieving the evidence collected in the last iteration. See also *APPENDIX F: Published APIs, Component: Evidence Collection (Cloud Discovery)*.

4.7.1.1.3 Graphical User Interface

This component does not have a graphical interface.

4.7.1.1.4 TRL

The TRL of the *Cloudditor Evidence Collector* is 4 at the moment of writing. After the validation phase ends, it is expected to be 5.

4.7.1.2 Wazuh

Wazuh [21] is a host-based intrusion detection system that features several modules for threat detection, integrity monitoring, incident response, and basic compliance monitoring. It is deployed on individual machines in the CSP's infrastructure and gathers data about security-related events on these machines. An additional component, the *Wazuh & VAT Evidence Collector* is used to connect *Wazuh* with the rest of the MEDINA framework by querying *Wazuh* and producing evidence based on its state and reported events. While *Wazuh* is a standalone

⁴¹ <https://git.code.tecnalia.com/medina/public/cloud-evidence-collector>

component, *Wazuh & VAT Evidence Collector* functions as a microservice within the MEDINA framework.

Wazuh addresses the KR4 (Continuous Evidence Management Tools).

The *Wazuh & VAT Evidence Collector* component is now Open Source with license Apache 2.0 and the source code is available on the public GitLab repository⁴².

For more details, please refer to deliverables D3.3 [19] and D3.6 [16].

4.7.1.2.1 Implementation and Integration Status

All requirements defined in D5.2 [3] for evidence collection with *Wazuh* are implemented. It is integrated (through the *Wazuh & VAT Evidence Collector*) with the *Security Assessment* component to which it sends the produced evidence.

The updates for the final version in M33 include the complete implementation of the *Wazuh* and *VAT Evidence Collector* component, which now also supports multiple metrics, implementation of changes related to advancements in the MEDINA data model as well as updates to the deployment scripts and documentation, enabling easier installation and configuration of *Wazuh & VAT Evidence Collector* in the use cases infrastructure.

4.7.1.2.2 Published APIs

There are no APIs exposed externally, i.e., to other MEDINA component). Internally, *Wazuh* exposes an API for querying its state which is used by the *Wazuh & VAT Evidence Collector*.

4.7.1.2.1 Graphical User Interface

This component does not have a graphical interface.

4.7.1.2.2 TRL

Based on existing open-source *Wazuh* platform, *Wazuh* has TRL9. For *Wazuh & VAT Evidence Collector*, after the validation phase ends, it is expected to be 6.

4.7.1.3 Vulnerability Assessment Tools

Vulnerability Assessment Tools (VAT) act as a vulnerability scanning and detection framework. The component incorporates multiple web application scanning tools that can be configured to periodically scan the CSP's services in testing or in production environments and report about detected vulnerabilities. It also provides capabilities to run user-provided vulnerability detection scripts which can be used with VAT to produce MEDINA-compliant evidence.

VAT addresses the KR4 (Continuous Evidence Management Tools).

The *Vulnerability Assessment Tools* component is now Open Source with license Apache 2.0 and the source code is available on the public GitLab repository⁴³.

⁴²<https://git.code.tecnalia.com/medina/public/wazuh-vat-evidence-collector>
<https://git.code.tecnalia.com/medina/public/wazuh-deploy>

⁴³<https://git.code.tecnalia.com/medina/public/wazuh-vat-evidence-collector>
<https://git.code.tecnalia.com/medina/public/vat-deploy>
<https://git.code.tecnalia.com/medina/public/vat-genscan>

Additional details about this component are available in deliverables D3.3 [19] and D3.6 [16].

4.7.1.3.1 Implementation and Integration Status

Similar to *Wazuh*, *VAT* is also connected to MEDINA by means of the *Wazuh and VAT Evidence Collector* component. Also, all requirements defined in D5.2 [3] for evidence collection with *VAT* are implemented. The updates for the final version in M33 relate only to the complete implementation of the *Wazuh & VAT Evidence Collector*.

4.7.1.3.2 Published APIs

No APIs are externally exposed by *VAT*. Internally, *VAT* exposes an API to provide information about the configuration and results of all scheduled and completed tasks. This API is used by the *Wazuh & VAT Evidence Collector* to produce evidence based on the state of *VAT*. The evidence is forwarded to the *Security Assessment* component.

4.7.1.3.3 Graphical User Interface

This component does not have a graphical interface.

4.7.1.3.4 TRL

The TRL of the *Vulnerability Assessment Tools* is 5 at the moment of writing. After the validation phase ends, it is expected to be 6.

4.7.1.4 Security Assessment (Clouditor)

Once the evidence has been collected, it must be assessed regarding the requirements specified in the respective certification catalogue. The *Security Assessment* first obtains pre-defined metrics data and policies from the *Orchestrator*. It then uses this data to assess incoming evidence regarding their compliance with the metric data. Assessment Results are the output of this component and include the compliance state, resource ID, and other information that enable auditors to trace a non-compliance to its exact source.

The *Security Assessment* component addresses KR4 (Continuous Evidence Management Tools) and KR5 (Cloud Certificate Evaluator).

The *Security Assessment* component is now Open Source with license Apache 2.0 and the source code is available on the public GitLab repository⁴⁴.

For more details, please refer to deliverables D3.3 [19] and D3.6 [16].

4.7.1.4.1 Implementation and Integration Status

The *Security Assessment* component currently implements all mandatory requirements as defined in deliverable D5.2 [3]. It is integrated with the *Cloud Evidence Collector* and the *Orchestrator*, and therefore implements all necessary integrations. These also include the integration with the Keycloak component.

Also, the component has been reimplemented as a separate microservice as well to conform to the MEDINA guidelines and data model. Its usage of the OPA policy engine has been added, which is used to evaluate incoming evidence against metrics and their target values. These are defined using the OPA policy language Rego.

⁴⁴ <https://git.code.tecnalia.com/medina/public/security-assessment>

4.7.1.4.2 Published APIs

The *Security Assessment* component offers two APIs: one for providing evidence to be assessed, and one for querying assessment results. See also *APPENDIX F: Published APIs, Component: Security Assessment (Clouditor)*.

4.7.1.4.3 Graphical User Interface

This component does not have a graphical interface.

4.7.1.4.4 TRL

The TRL of the *Security Assessment* component is 5 at the moment of writing. After the validation phase ends, it is expected to be 6.

4.7.1.5 Codyze

Codyze is a static analysis tool for source code. It assesses source code to identify, for example, misconfigured security functions. In MEDINA, a wrapper for *Codyze* has been developed to translate findings from *Codyze* into MEDINA assessment results that are sent to the *Orchestrator*. This integration therefore represents an integration on the security assessment level.

Codyze addresses KR4 (Continuous Evidence Management Tools).

Codyze is now Open Source with license Apache 2.0 and the source code is available on the public GitLab repository⁴⁵.

For more details, please refer to deliverables D3.3 [19] and D3.6 [16].

4.7.1.5.1 Implementation and Integration Status

In MEDINA, *Codyze* currently implements all mandatory requirements as defined in deliverable D5.2 [3]. It has been extended to map findings in source code to several metrics that are relevant for the EUCS requirements.

The main integration of *Codyze* is with the *Orchestrator* as it sends its assessment results to this component. Upon analysing source code and the corresponding project structure, *Codyze* assess security related implementation details and secure developer interactions with the project. Findings are translated into MEDINA assessment results and submitted to the *Orchestrator* for further processing. To this end, *Codyze* uses the API specification provided with the *Orchestrator*.

During the third round from M27-M33, the integration of *Codyze* with the *Orchestrator* has been refined and its CI/CD integration has been finished, along with the development of new security metrics.

Codyze has to be integrated into a CI/CD pipeline that builds and deploys cloud services to the cloud environment. *Codyze* assesses the source code of cloud services before they are deployed ensuring a defined security level of deployed cloud services. In this role, *Codyze* acts as a quality and security assurance gate in a CI/CD pipeline. It prevents the roll out of cloud services if they violate specified security requirements. Integration of *Codyze* within MEDINA has been done

⁴⁵ <https://git.code.tecnalia.com/medina/public/codyze>

within a CI/CD pipeline set up by Bosch and within MEDINA's Jenkins. The latter is used to assess MEDINA components themselves and provide possible feedback to partners.

4.7.1.5.2 Published APIs

Codyze does not publish an API.

4.7.1.5.3 Graphical User Interface

This component does not have a graphical interface.

4.7.1.5.4 TRL

The TRL of the *Codyze* is 5 at the moment of writing. After the validation phase ends, it is expected to be 6.

4.8 Company Compliance Dashboard (block #8)

To work with the MEDINA framework, a potential customer has the freedom to choose between two different paths: easily open up the *Integrated User Interface* (see section 5) or connect an already existing tool/ software to the MEDINA core APIs – offered by most components – and work in a familiar environment. The *Company Compliance Dashboard (CCD)* is the showcase approach of Use Case 2 (WP6) to demonstrate this functionality and strengthen the modularity of the MEDINA framework.

The main purpose of the *Company Compliance Dashboard* is to demonstrate how MEDINA achieves a high level of modularity through its components and several core APIs, such that potential customers are able to integrate MEDINA seamlessly into their own ecosystem.

Recap of the CCD development

This approach was first addressed in early 2021 with deliverable D6.1 [22] in Section 2.2.3.1 “Fabasoft VDE and Cloud Apps”, and 2.2.3.2 “Demo-System of Use Case 2”. Deliverable D6.2 [23] extended this approach and showed explicitly the developed wireframes (D6.2, Section 3.3.2 “Wireframes”) and their approaches to address a third-party tool, which is able to connect to APIs of components like the *Orchestrator* or the *SATRA* tool.

D6.2 already demonstrated the use of the MEDINA framework for any business that is able to set-up their own front-end (i.e., UI). Section 3.3.3 “Implementation of Demo System” of D6.2 stated in fall 2021 that the demo system for Use Case 2 will be part of the Fabasoft Cloud.

Deliverable D6.3 [9] mapped the MEDINA workflows to the different *CCD* features and functionalities and described how the first APIs could already be addressed in summer 2022 (D6.3, Section 2 “Integration Approach and Results” and onwards). Appendix E of D6.3 presents a complete collection of the wireframes for the *CCD*.

4.8.1 Implementation and Integration Status

At the point of deliverable D5.5 submission, the *CCD* is a stable prototype, and a publicly available test installation exists that is used for user feedback and continuously interacts with the MEDINA components - especially the *Orchestrator* – to receive and send audit relevant information. A batch of test data could be prepared and is used for this installation. Deliverable D6.4 will present a complete documentation of the *CCD* implementation, testing and current capabilities within the MEDINA framework.

4.8.2 Graphical User Interface

The CCD uses the Fabasoft Cloud UI functionalities and is written in the Fabasoft domain specific language (DSL) app.ducx⁴⁶. The Dashboard (see Figure 52) uses elements that are integrated with a high-charts⁴⁷ functionality.

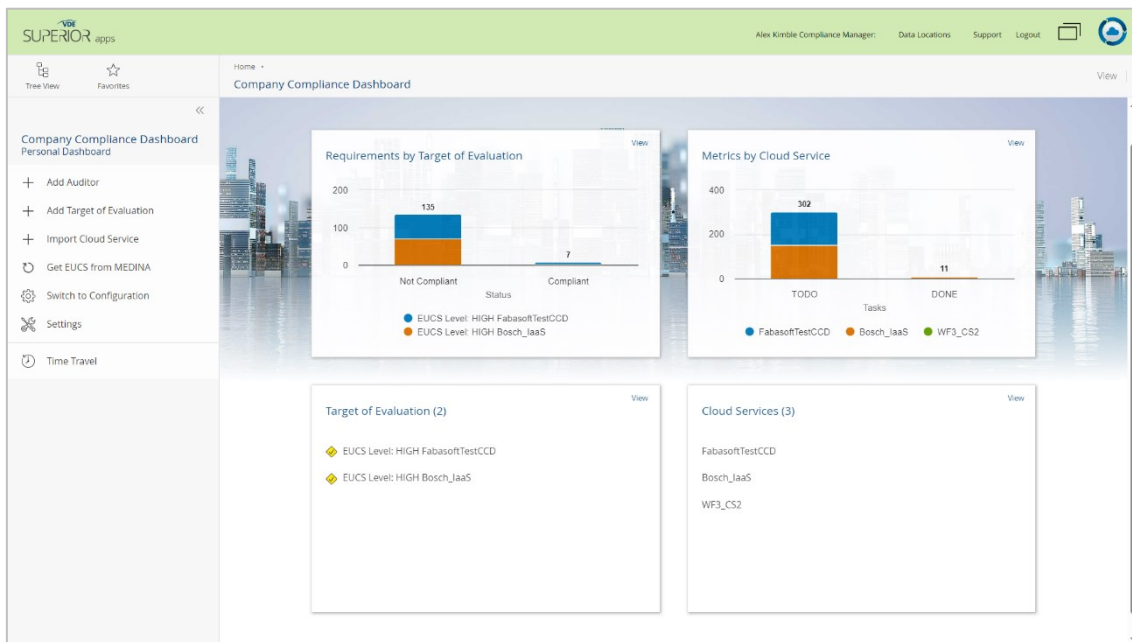


Figure 52. CCD main dashboard view

The first functionality that the CCD provided through MEDINA APIs was importing the EUCS from the *Catalogue of Controls and Metrics* (see Figure 53).

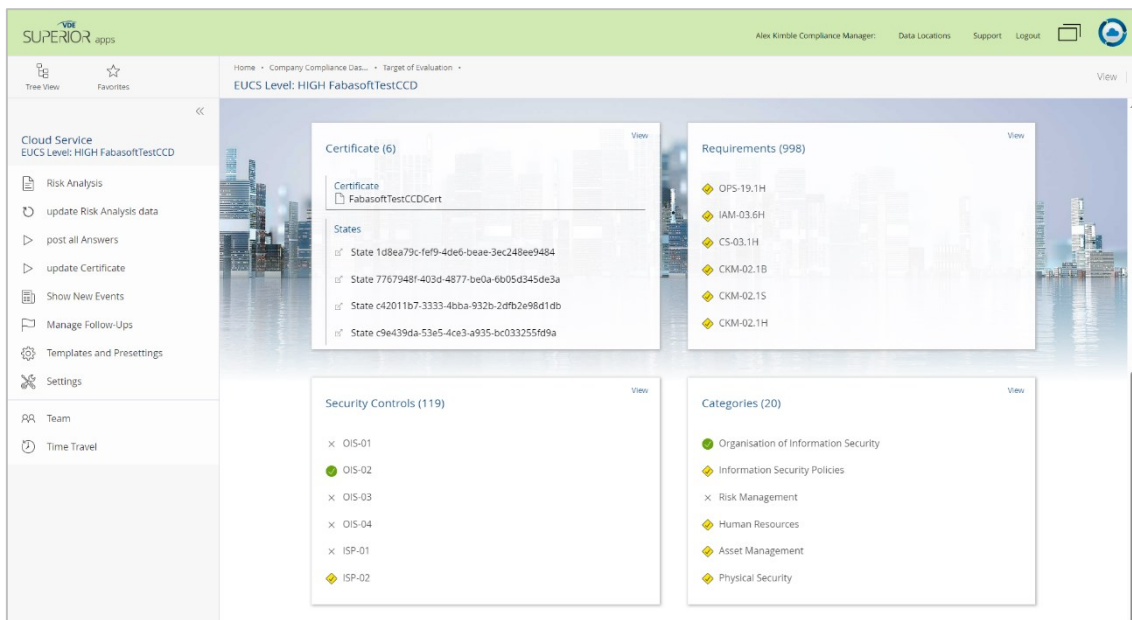


Figure 53. CCD - import of EUCS

⁴⁶ <https://help.cloud.fabasoft.com/doc/Model-Based-Customizing/introduction.htm>

⁴⁷ <https://www.highcharts.com/>

The CCD offers full transparency of workflows by logging every step as an approved verification in the process of conducting an example audit with MEDINA (see Figure 54).

Signatures

Last Signature Type

Release Metric Implementation

Last Signature by

Metric Owner: Christopher Carney

Last Signature on/at

06/09/2023 10:38:18 AM

Signatures

☐

Show Details (5)

	Signature Type	Signed by	Signed on/at	Remark
1	Assign Metric	Compliance Manager: Alex Kim...	05/22/2023 03:24:11 PM	for Demo
2	Assign Metric	Metric Owner: Christopher Car...	06/09/2023 10:34:40 AM	
3	Accept Metric Implementation Task	Metric Implementer: Julia Briere	06/09/2023 10:36:26 AM	i ll do it
4	Metric Implemented	Metric Implementer: Julia Briere	06/09/2023 10:37:02 AM	seems alright
5	Release Metric Implementation	Metric Owner: Christopher Car...	06/09/2023 10:38:18 AM	

Figure 54. CCD - view on the workflow verification information

4.8.3 TRL

The TRL of the CCD component can be evaluated as 5. As it is embedded in the Fabasoft Cloud Process Ecosystem, it already has several core features with TRL 9 – 10, however, the Cloud Solution itself is only tested as a demonstrator showcase in a controlled real-world scenario. More on that in deliverable D6.4, October 2023.

5 MEDINA Integrated User Interface (block #8)

This section provides an in-depth description of the MEDINA *Integrated User Interface* (IUI).

5.1 Implementation

During the third integration round (M27-M33) two more components were integrated into the MEDINA IUI, the *Trustworthiness System*, and the *Self-Sovereign Identity*. The focus was on the harmonization of the components GUIs and on extending the authorization strategy, including more roles and users.

The implementation of the authentication was reorganized introducing the “anonymous” user concept: the user now lands to the MEDINA IUI open access page⁴⁸ which provides an overview on what the platform offers (see Figure 8). Then, he/she can sign in and access to the whole framework functionalities.

5.1.1 Functional description

The *MEDINA Integrated User Interface* is the landing page the user access to when he/she is searching for the MEDINA framework over the Internet.

The MEDINA framework architecture adopted for the User Interface is the micro frontend architecture [24]. This type of architecture is a modern pattern introduced to decompose the monoliths frontend application into smaller and simpler parts. This type of approach makes it easier to develop and deploy the User Interface and allows teams to work independently.

The micro frontend architecture fits well with the microservices approach adopted in the MEDINA framework: the entire application is composed of multiple microservices that collaborate with each other to made up the overall functionalities of the application. At the same time, each microservice can have its own user interface, called micro-frontend. The MEDINA IUI encapsulates the graphical user interfaces (GUI) of all components.

During the M27-M33 period, the main focus relied on improving the look and feel, and user experience.

5.1.1.1 Fitting into overall MEDINA Architecture

The IUI is part of block #8 in the MEDINA Architecture (see Figure 16). It acts as entry point for users to the MEDINA framework: it integrates with existing authentication and guides users based on their authorization level to the user interfaces of specific components.

The IUI implements the authorization and authentication strategies and adopts the micro-frontend architecture. Our goal for this component is to realize a user comfortable experience and guide each component GUI to adopt the same look and feel and authorization strategies.

5.1.1.2 Component card

Table 29 show the component card of the *MEDINA Integrated User Interface*.

⁴⁸ <http://demo.medina-project.eu> redirects to <https://integrated-ui-test.k8s.medina.esilab.org/>

Table 29. IUI Component card

Component Name	Integrated User Interface (IUI)		
Main functionalities	The component provides the following functionalities: <ul style="list-style-type: none">Secure primary point of access for the MEDINA frameworkIntegration with the existing authenticationIntegration of all the separated components GUI into a single point of accessGuide the users based on their authorization level to specific components UIs		
Sub-components Description	No subcomponents exist in the IUI		
Main logical Interfaces	Interface name	Description	Interface technology
	IUI	Main point of access to the framework, integrates all the other micro frontends	HTTPS (browser)
Requirements Mapping	List of requirements covered by this component IUI.01, IUI.02, IUI.05, IUI.06		
Interaction with other components	Interfacing Component		Interface Description
	Catalogue		Integrates the Catalogue UI
	CNL Editor		Integrates the CNL Editor UI
	CCE		Integrates the CCE UI
	AMOE		Integrates the AMOE UI
	Orchestrator		Integrates the Orchestrator UI
	RAOF		Integrates the RAOF UI
	Trustworthiness System		Integrates the Trustworthiness System UI
	SSI Framework		Integrates the SSI Framework UI
Keycloak		Interacts with Keycloak for the authentication and authorization	
Relevant sequence diagram/s	Not applicable		
Current TRL	TRL5		
Programming language	AngularJS		
License	Proprietary. Copyright by HPE		
WP and task	WP5 – Task 5.3		
Workflows	WF2 Preparation of MEDINA Components		

5.1.1.3 Requirements

Below is the list of the requirements of the *Integrated User Interface*. For each requirement there are details on how it has been fulfilled.

Table 30. IUI.01 requirement

Requirement id	IUI.01
Short title	Authentication integration via Keycloak Adapter
Description	Every component must implement an adapter that allows it to authenticate with the Catalogue's Keycloak authentication service in order to prevent unauthenticated users to access its resources.
Status	Fully implemented
Priority	Should

The requirement IUI.01 is fully implemented: the IUI uses a Keycloak Adapter in the Angular application which decodes the token and verifies the user's authentication: if the user is not signed in, he/she will be redirected to the Keycloak login page, otherwise the MEDINA framework IUI will be displayed.

Table 31. IUI.02 requirement

Requirement id	IUI.02
Short title	Authorization integration via Keycloak
Description	Every component that has resources that should only be accessed by specific user roles must enforce authorization on its internal logic (e.g., in a REST API, define at controller level that a specific endpoint can be accessed only with the Product Engineer role). This can be obtained by defining appropriate configuration on the Catalogue's Keycloak (Role Mapping).
Status	Fully implemented
Priority	Should

The requirement IUI.02 is fully implemented. The IUI displays different sections which can be navigated through the left side menu: depending on the user's role, some menu sections are hidden and not accessible by the user.

Table 32. IUI.05 requirement

Requirement id	IUI.05
Short title	External Identity Provider Configuration
Description	Users should be able to authenticate using their existing enterprise identity provider once it has been configured to do so. Ideally, MEDINA Generic Roles should be inherited from existing claims / roles.
Status	Fully implemented
Priority	Should

The requirement IUI.05 has been implemented using the Bosch external identity provider as a proof of concept on how external providers can be integrated. In the current IUI version, user can login both through a MEDINA account registered in the MEDINA Keycloak or using existing Bosch user credentials.

Table 33. IUI.06 requirement

Requirement id	IUI.06
Short title	Homogeneous look and feel
Description	Each component micro-frontend embedded into IUI should abide to a set of graphical constraints and rules that the MEDINA consortium agreed on in order to homogenize look and feel.

Requirement id	IUI.06
Status	Implemented
Priority	Should

The requirement IUI.06 is the one that was implemented during the M27-M33 period. The IUI indeed embeds the micro-frontends, which are developed by each team independently. The consortium agreed on generic rules to be followed by each component in order to achieve a consistent and comfortable style. These rules have been reported both in a document and in HTML and CSS files shared on the private GitLab repository.

5.1.2 Technical description

This section describes how the MEDINA IUI implements the micro frontends architecture [25] and how it assures authentication and authorization. This kind of architecture allows to embed in a main frontend component (Integrated UI) any other UIs in the framework regardless of the underlying technology.

5.1.2.1 Component architecture

Figure 55 describes a simplified architecture from the Integrated UI perspective. The client searches for the MEDINA application over the Internet and its requests are processed by the nginx reverse proxy. The client lands on the Integrated UI and then can navigate to the GUIs provided by other components. The nginx proxy redirects the client requests to the right component. All the components need to implement a Keycloak adapter in order to enforce authentication and authorization.

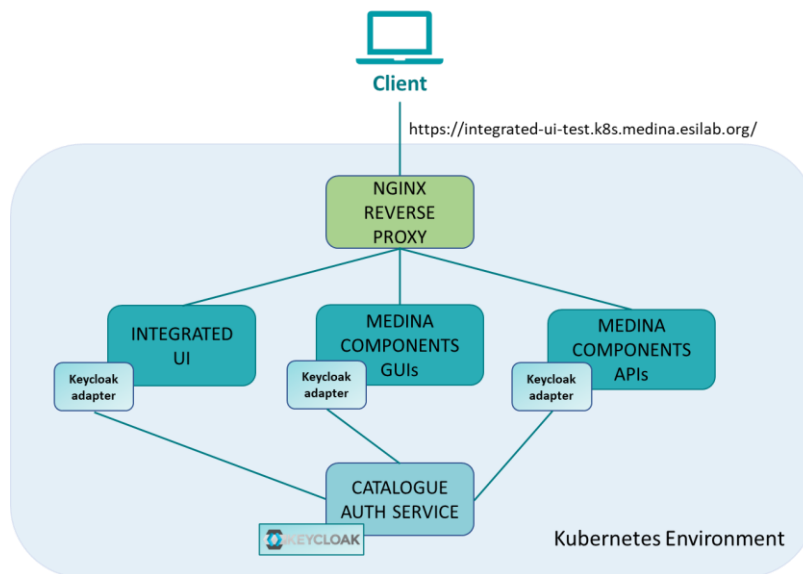


Figure 55. MEDINA UI Architecture

5.1.2.2 Description of components

The IUI interacts with the *Catalogue* auth service through the implementation of the Keycloak Adapter and embeds the components GUI using the IFrame strategy.

5.1.2.2.1 Authentication and authorization

The *Catalogue* auth service is managed by Keycloak [26] open-source identity and access management service. It is configured using its dashboard and makes possible advanced

authentication and authorization capabilities, including SSO, identity brokerage and role mapping.

Every component implements a “Keycloak adapter” which acts as an HTTP interceptor and checks on resources requests whether:

- The client requesting user authentication is a registered client.
- The user is authenticated, if not it redirects to login page.
- The user is authorized for the requested resource based on its role on Keycloak configuration, if not it redirects to an appropriate error page.

Once a user is authenticated, a JWT containing information about the user and roles is provided. This allows features such as conditional formatting and routing based on the user's role to be securely implemented. This feature is therefore essential to implement the role-based GUI adaptation described in section 3.4. For debugging purposes, the Keycloak JWT fields are listed into a dedicated IUI page, as shown in Figure 56.

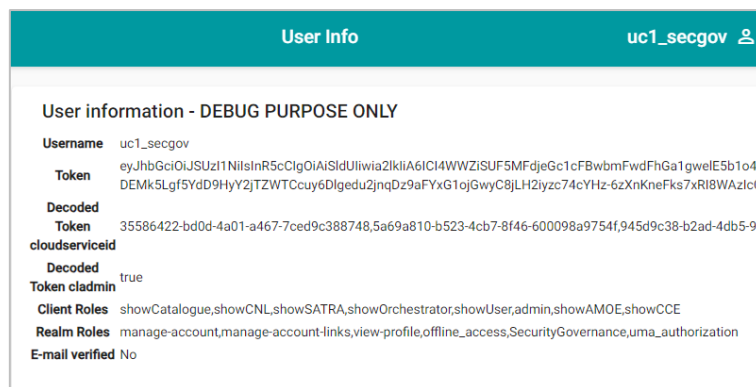


Figure 56. MEDINA JWT fields

Moreover, in MEDINA the user authentication is available also with an external identity provider, in particular the Enterprise Identity Provider authentication provided by UC1 (Bosch Active Directory) is successfully integrated and the login page allows the user to sign in as a Keycloak user or as a Bosch user, as shown in Figure 57.

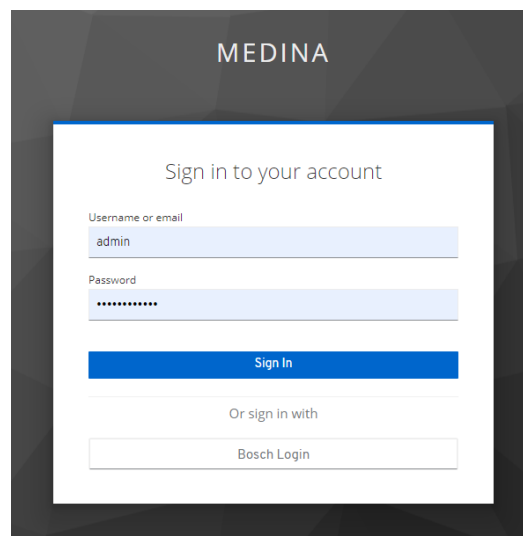


Figure 57. MEDINA login page

During this third integration round, Keycloak has been extended to provide more users and roles required by the UCs. Table 34 shows the list of all roles and usernames used.

Table 34. Roles and usernames implemented in Keycloak

Role	Keycloak Role	Username
IT Security Governance	SecurityGovernance	UC1_SecGov
Security Analyst	SecurityAnalyst	UC1_SecAnalyst
Domain Governance	DomainGovernance	UC1_DomGov
Product and Service Owner	ProductOwner	UC1_ProdOwn
Product (Security) Engineer	ProductSecurityEng	UC1_ProdSec
Chief Information Security Office (CISO)	CISO	UC1_CISO
Customer	Customer	Non-Authenticated User
Auditor	Auditor	UC1_Auditor

The logic implemented for the authorization feature depends on the user's role and is described in detail in section 3.4. For the UI, the authorization access for each role has been defined in Table 28.

In order not to lose the Keycloak configuration, a backup is periodically done.

5.1.2.2.2 Integration of components

The IUI embeds the MEDINA components GUIs, and during the M27-M33 period two more graphical user interfaces have been added: the *SSI Framework* UI and the *MEDINA Evidence Trustworthiness System* UI. The integration of micro-frontends is achieved by means of IFrames. In particular, since the micro-frontends are deployed in the Kubernetes cluster, we are able to integrate them by providing the URL of the component and automatically update the referred services in the application, with great benefits to productivity. Table 35 shows the list of all the components that have been integrated.

Table 35. List of all components integrated in the MEDINA IUI

Component Name	Integration Strategy
Catalogue of Metrics and Controls	IFrame
Orchestrator	IFrame
CNL Editor	IFrame
Continuous Certificate Evaluation	IFrame
Risk Assessment and Optimisation Framework	IFrame
Organizational Evidence Gathering and Processing	IFrame
MEDINA Evidence Trustworthiness System	IFrame
SSI Framework	IFrame

Since the GUIs of the components are developed by different teams, during the M27-M33 period we agreed with the consortium on a common set of style rules. The result of this decision are two HTML and CSS files available on the private GitLab, that all development teams can use as guidelines or import into their code. The main impact of this decision involves:

- A common toolbar, present on each micro frontend, containing navigation buttons with the same format. This toolbar has a new button, the “Help” button, which links to the specific User Manual of each component, described in the *APPENDIX G: User Manual*.
- A common footer, with the copyright and the logo of the European project and the partner.
- A unified background colour and common buttons for the most used operation (e.g., delete, create, modify).

Figure 58 shows an example of the results achieved.

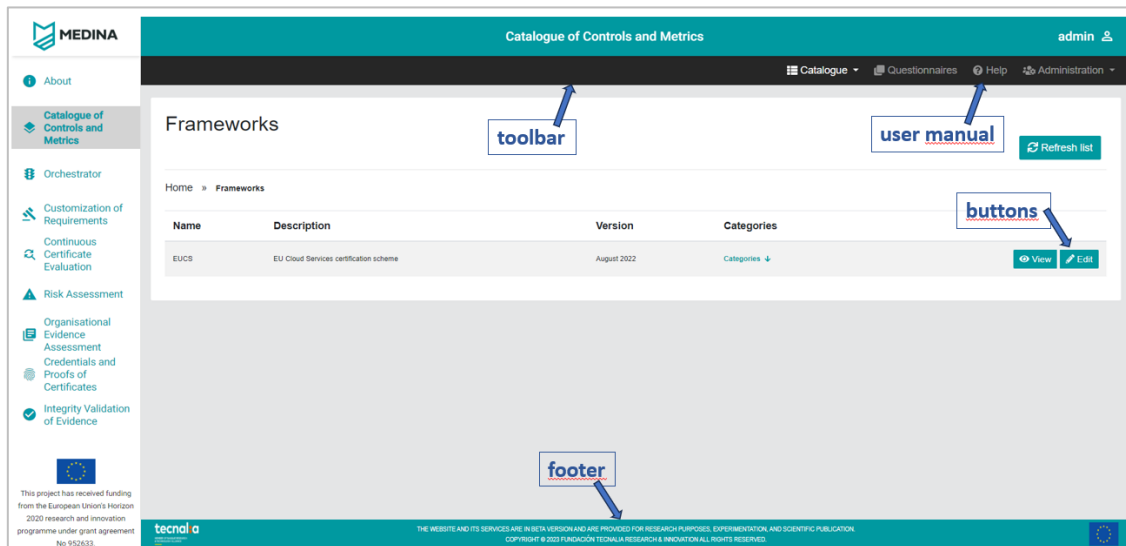


Figure 58. UI of components - common styles

5.1.2.3 Technical specifications

The UI prototype is developed using Angular 12 [27], a modern typescript framework that allows to build high-performance, scalable, component-based single page web applications. The framework is enriched with Angular Material 2 library [28], a set of high quality animated responsive components that follows Material Design UI specifications. The application runs on a Nginx web server [29].

The UI develops a cornice in common with all micro frontends, which is composed by a left menu navigation and a header with the component name and user information, as shown in Figure 59.

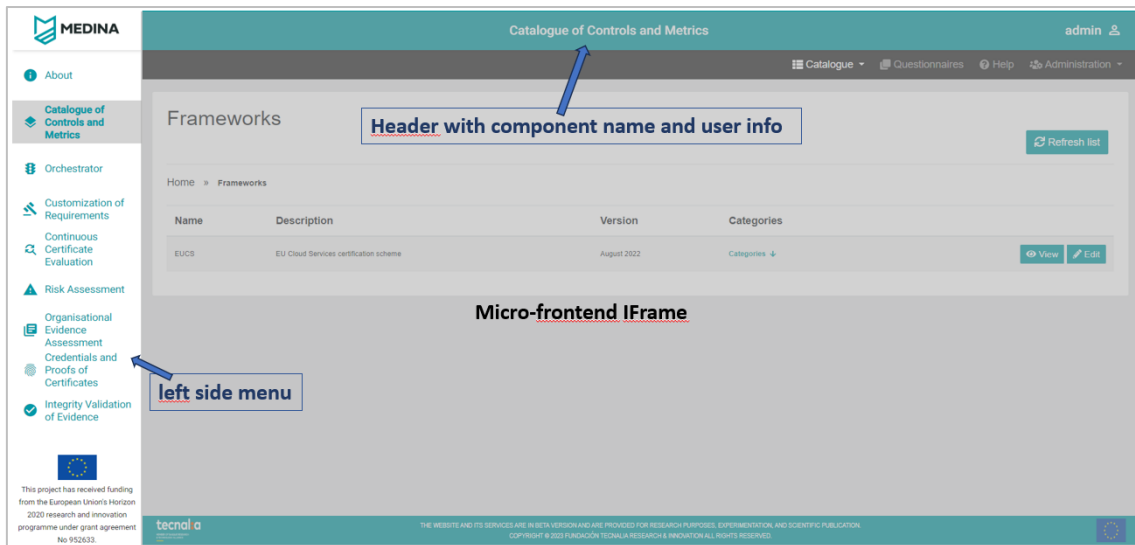


Figure 59. MEDINA IUI cornice

During the M27-M33 period, the main focus was on the look and feel of the IUI and the micro frontends. The IUI introduced the highlighting of the user's chosen menu option and the name of the section in the page header.

Other important change concerns the “About” page, which is the page the IUI is responsible for and contains an overview of the framework (see Figure 60). This page now contains a description of the MEDINA project, a role-based diagram, and links to the social media and a video showing an overview demo of the MEDINA framework.

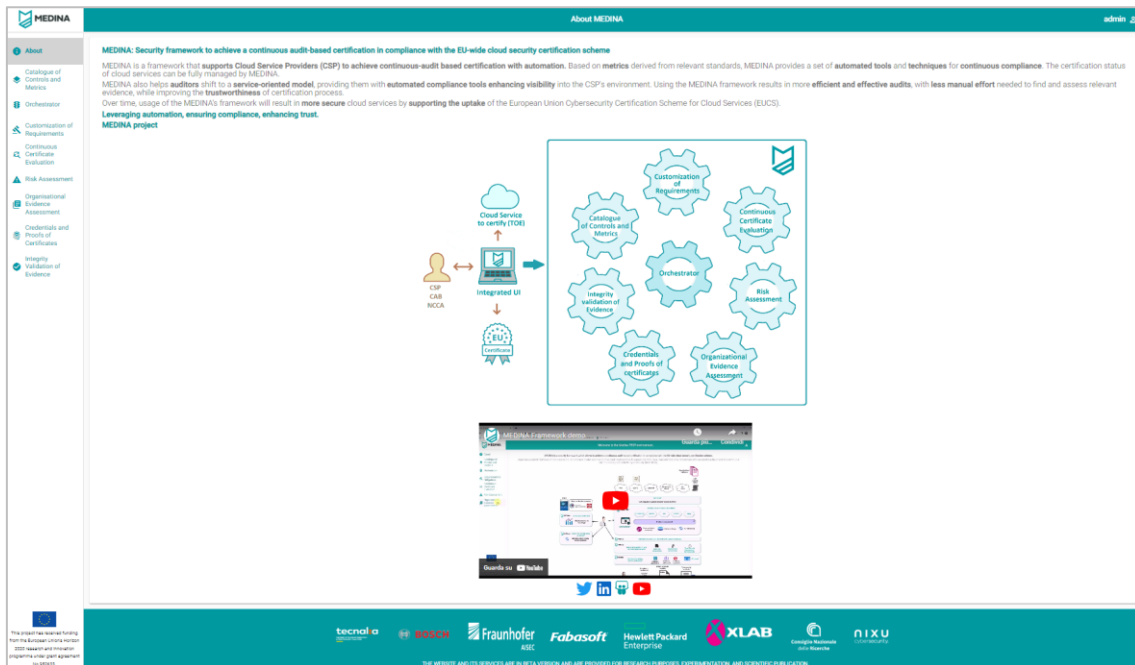


Figure 60. IUI - About page for the admin user role

5.1.3 Delivery and usage

5.1.3.1 Package information

Table 36 shows the IUI package has the following structure.

Table 36. Package Structure

Path	Description
/conf	Contains specifications that are used by docker when generating an image to configure the Nginx web server
/dist	Contains the result of the build
/keycloak-dev-docker-compose	The docker compose for local development described in the readme file
/kubernetes	Contains kubernetes configuration files for deployment
/Kubernetes-test	Contains kubernetes test configuration files for deployment
/node_modules	Contains installed npm modules
/realm-config	Contains a backup of the keycloak realm configurations
/src/Dockerfile	Contains specifications that are used in order to build a docker image
/src/assets/config/config.json	Contains application configuration which can be modified at runtime
/src/environments/	Contains static configurations based on environment (dev or test)
/src/app/services	Contains services that are generated via OpenAPI specs in order to integrate with other applications in the MEDINA framework
/src/app/	Contains the main components of the application

5.1.3.2 Download

The *Integrated User Interface* is a closed-source and published on the private TECNALIA GitLab at: https://git.code.tecnalia.com/medina/wp5/task_5.3/integrated-ui [internal use only – authentication required].

6 Conclusions

This document reports on how the objectives for M33 related to task 5.3 have been fulfilled. The adoption of the CI/CD strategy enables the automatic release of the components in the two virtual environments of the Kubernetes cluster, “dev” and “test”. The components that made up the eight building blocks of the MEDINA reference architecture have reached a high level of maturity and they have been completely integrated in the IUI. From M33 until the end of the project the environment will be kept in maintenance.

In addition, the document shows the nine scenarios of the generic MEDINA workflows extended by introducing roles and their level of visibility that define the allowed actions. At the same time, integration activities have been conducted with the support of technical webinars and demonstrations on different topics regarding the DevOps approach integrated with the Kubernetes environment, the Keycloak integration, and how to manage the authorization and filtering in MEDINA.

Finally, the solution has been improved with feedback coming from the Use Cases and by improving the security pipeline by adding the MEDINA component *Codyze*. A satisfactory status of completion has been reached for each component and the *Integrated User Interface* has been finalised with the fulfilment of all requirements, in particular regarding the look and feel and the introduction of the Welcome Page for an unauthenticated user.

7 References

- [1] MEDINA Consortium, “D5.3 MEDINA integrated solution-v1,” 2022.
- [2] MEDINA Consortium, “D5.4 MEDINA integrated solution - v2,” 2023.
- [3] MEDINA Consortium, “D5.2 MEDINA Requirements, Detailed architecture, DevOps infrastructure and CI/CD and verification strategy,” 2022.
- [4] MEDINA Consortium, “D5.1 MEDINA Requirements, Detailed architecture, DevOps infrastructure and CI/CD and verification strategy,” 2021.
- [5] “K8s,” [Online]. Available: <https://kubernetes.io/docs/home/>. [Accessed July 2023].
- [6] “SPDX license,” [Online]. Available: <https://spdx.org/licenses/>. [Accessed July 2023].
- [7] “JFrog Artifactory,” [Online]. Available: <https://jfrog.com/artifactory/>. [Accessed July 2023].
- [8] “Codyze,” [Online]. Available: <https://www.codyze.io/?ref=https://githubhelp.com>. [Accessed July 2023].
- [9] MEDINA Consortium, “D6.3 Use cases development and validation-prototypes-v1,” 2022.
- [10] MEDINA Consortium, “D6.4 Use cases development and validation-prototypes-v2,” 2023.
- [11] ENISA, “EUCS - Cloud Services Scheme,” [Online]. Available: <https://www.enisa.europa.eu/publications/eucs-cloud-service-scheme>. [Accessed July 2023].
- [12] MEDINA Consortium, “D2.8 Risk-based techniques and tools for Cloud Security Certification-v3,” 2023.
- [13] MEDINA Consortium, “D2.2 Continuously certifiable technical and organizational measures and catalogue of cloud security metrics-v2,” 2023.
- [14] ENISA, “EUCS -Cloud Service Scheme,” Draft version provided by ENISA (August 2022) - not intended for being used outside the context of MEDINA, 2022.
- [15] MEDINA Consortium, “D2.5 Specification of the Cloud Security Certification Language-v3,” 2023.
- [16] MEDINA Consortium, “D4.5 Methodology and tools for risk-based assessment and security control reconfiguration - v2,” 2023.
- [17] MEDINA Consortium, “D4.3 Tools and Techniques for the Management and Evaluation of Cloud Security Certifications - v3,” 2023.
- [18] MEDINA Consortium, “D4.2 Tools and Techniques for the Management and Evaluation of Cloud Security Certifications - v2,” 2022.

- [19] MEDINA Consortium, "D3.3 Tools and techniques for the management of trustworthy evidence-v3," 2023.
- [20] MEDINA Consortium, "D3.6 Tools and techniques for collecting evidence of technical and organisational measures-v3," 2023.
- [21] Wazuh Inc., "Wazuh," [Online]. Available: <https://wazuh.com>. [Accessed July 2023].
- [22] MEDINA Consortium, "D6.1 Use cases specification and evaluation methodology-v1," 2021.
- [23] MEDINA Consortium, "D6.2 Use cases specification and evaluation methodology-v2," 2021.
- [24] M. Fowler, "micro-frontends," [Online]. Available: <https://martinfowler.com/articles/micro-frontends.html>. [Accessed July 2023].
- [25] L. M. D. T. Severi Peltonen, "Motivations, benefits, and issues for adopting Micro-Frontends: A Multivocal Literature Review," DAZN, London, United Kingdom and Tampere University, Tampere, Finland, 24 03 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0950584921000549>. [Accessed July 2023].
- [26] "Keycloak," [Online]. Available: <https://www.keycloak.org/>. [Accessed July 2023].
- [27] "Angular," [Online]. Available: <https://angular.io/>. [Accessed July 2023].
- [28] Google, "Angular Material," [Online]. Available: <https://material.angular.io/>. [Accessed July 2023].
- [29] "Nginx," [Online]. Available: <https://www.nginx.com/>. [Accessed July 2023].
- [30] "RKE," [Online]. Available: <https://rancher.com/docs/rke/latest/en/os/>. [Accessed July 2023].
- [31] "Rook/Ceph," [Online]. Available: <https://rook.io/docs/rook/v1.8/>. [Accessed July 2023].
- [32] "METALLB," [Online]. Available: <https://metallb.universe.tf/>. [Accessed July 2023].
- [33] "SSH," [Online]. Available: <https://www.ssh.com/academy/ssh/protocol>. [Accessed July 2023].
- [34] Linux Foundation, "Helm package manager," [Online]. Available: <https://helm.sh/>. [Accessed July 2023].
- [35] Linux Foundation, "Cert manager," [Online]. Available: <https://cert-manager.io/docs/>. [Accessed July 2023].
- [36] "Apache Maven Project," [Online]. Available: <https://maven.apache.org/>. [Accessed July 2023].

8 APPENDIX A: Operating Environment

The MEDINA framework functionalities are made up by the collaboration of all the micro-services, which communicate each other through REST API, are packaged in Docker images and run in Docker containers. Kubernetes orchestrates all these containers in a virtual environment running on high-available cluster.

8.1 Kubernetes Installation and Configuration

This section illustrates the container orchestration solution that is executed over the setup infrastructure.

Different resources are needed to proceed with the installation and configuration of the cluster. We used Rancher Kubernetes Engine (RKE) [30] for the installation of Kubernetes [5] in the three nodes, Rook/Ceph [31] for the configuration of storage and MetalLB [32] for the network configuration.

The Kubernetes cluster is configured and managed by RKE, an open-source distribution that simplifies the installation and operations of Kubernetes (see Figure 61). The RKE client is installed on a console host at the `cicd.medina.esilab.org` VM and communicates with the nodes of the cluster through SSH [33]. Through RKE, we have configured each cluster node to be both Master and Worker, guaranteeing fault-tolerance and high availability. To do so, RKE creates on each of them the control plane, kubelet and kube-proxy resources in Docker containers.

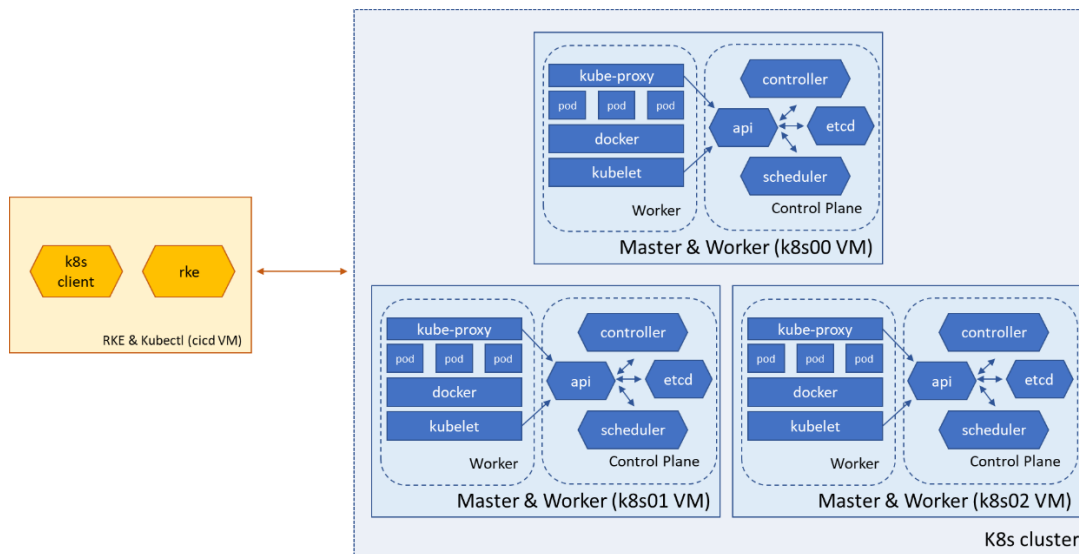


Figure 61. Kubernetes cluster installation with RKE

All the micro-services can store their data in an easy and secure way thanks to the configuration of a distributed filesystem. Indeed, each node of the cluster provides 200 GB of storage, managed by Rook/Ceph, and exposed as a single, unified cluster filesystem.

Ceph is an open-source distributed storage solution for deliver block storage, object storage and shared filesystem in a single, unified system. It ensures cluster state monitoring and handles data replication, recovery, and rebalancing.

Ceph is deployed to the Kubernetes cluster by Rook that is an open-source cloud-native storage orchestrator enabling Ceph to easily run on a Kubernetes cluster. The Rook operator is a Kubernetes resource that automates the Ceph management and installation and turns Ceph into a self-scaling, self-managing, and self-healing storage service.

Thanks to this configuration, the data are replicated across the three nodes, 200 GB of storage and fault-tolerance and high availability are assured.

The micro-services running on the Kubernetes cluster are packaged in Docker images and stored on a private Docker Registry running on Artifactory by Jfrog [7].

In order to have Kubernetes access the Docker Registry, a specific integration has been done: a *secret* has been created with the registry credentials. This allows Kubernetes to pull the micro-service image and then run it on the cluster.

The images are pushed to the Docker registry according to the structure shown in Figure 62, that was agreed in the project.

```
<medina_registry_url>/<work_package>/<task >/<image>:<tag>
```

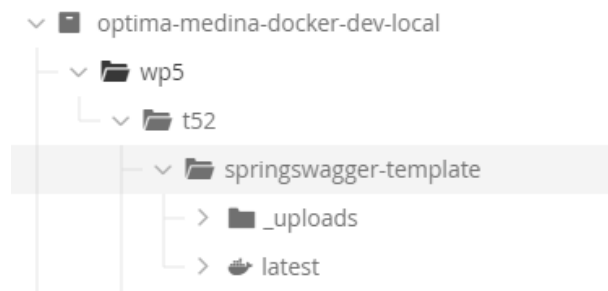


Figure 62. Excerpt of MEDINA’s Docker registry

The REST API exposed by each micro-service is reachable from the Internet using the “*.k8s.medina.esilab.org” URL, corresponding to the static public IP 172.26.124.120. In particular, on the Kubernetes cluster a nginx [29] service is configured as a proxy to redirect all the requests to the correct micro-service component. The binding between the nginx service and the public IP is setup with MetalB. MetalB [32] is a network load-balancer implementation that associates the public IP to the nginx service and uses standard routing protocols to make available (part of) the network behind the Kubernetes cluster. It is essential for the MEDINA cluster because, unlike a public cloud provider cluster, this one has no load balancer and Kubernetes does not provide it by itself.

The user can address the environment s/he wants using this URL naming convention (see Figure 63):

```
<component_name>-<environment [test or dev]>.k8s.medina.esilab.org
```

For example, if the user needs to refer to the API exposed by the “api-swagger” component running on the Kubernetes “test” environment, s/he will address it as:

```
api-swagger-test.k8s.medina.esilab.org
```

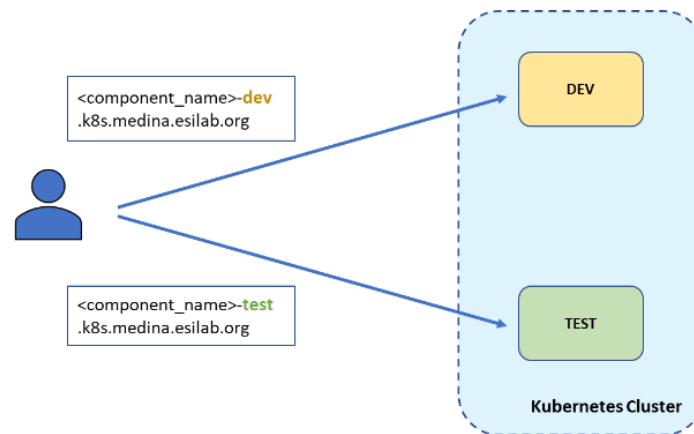


Figure 63. URL naming convention for dev/test environments

8.2 Kubernetes Dashboard

The Kubernetes Dashboard is a web-based User Interface for the Kubernetes cluster (see Figure 65). It is helpful to deploy containerized applications to a Kubernetes cluster, troubleshoot them, and manage the cluster resources. We installed K8s Dashboard using the Helm package manager [34].

To have access to the Dashboard it is needed to generate a Service Account token by creating a service account (see Figure 64). We have two service accounts with different permissions: one is “dashboard-admin” that has access to all cluster resources and the other is “partner-user” for the partners’ access that has restricted permissions only to “dev” and “test” namespaces. We must copy the token to sign into the Dashboard.

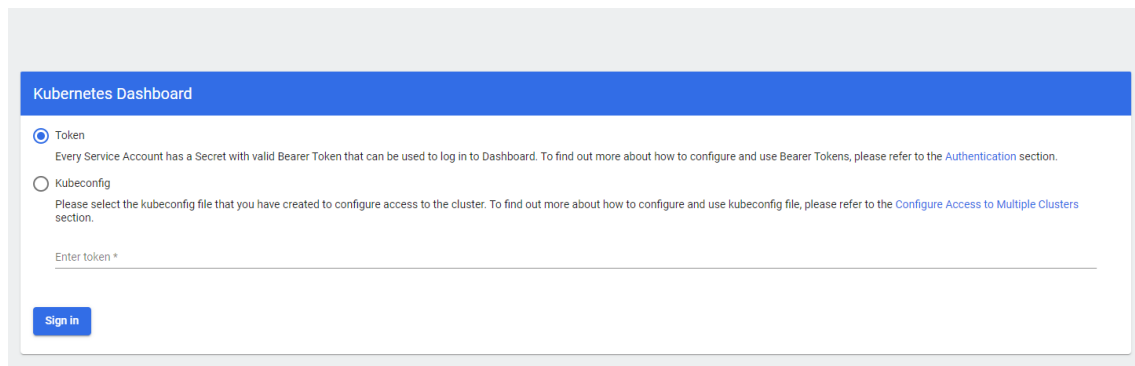


Figure 64. Service Account type used for the Kubernetes Dashboard

The Dashboard is exposed over HTTPS (see Figure 65) at <https://dashboard.k8s.medina.esilab.org/#/login> [internal use only – authentication required].

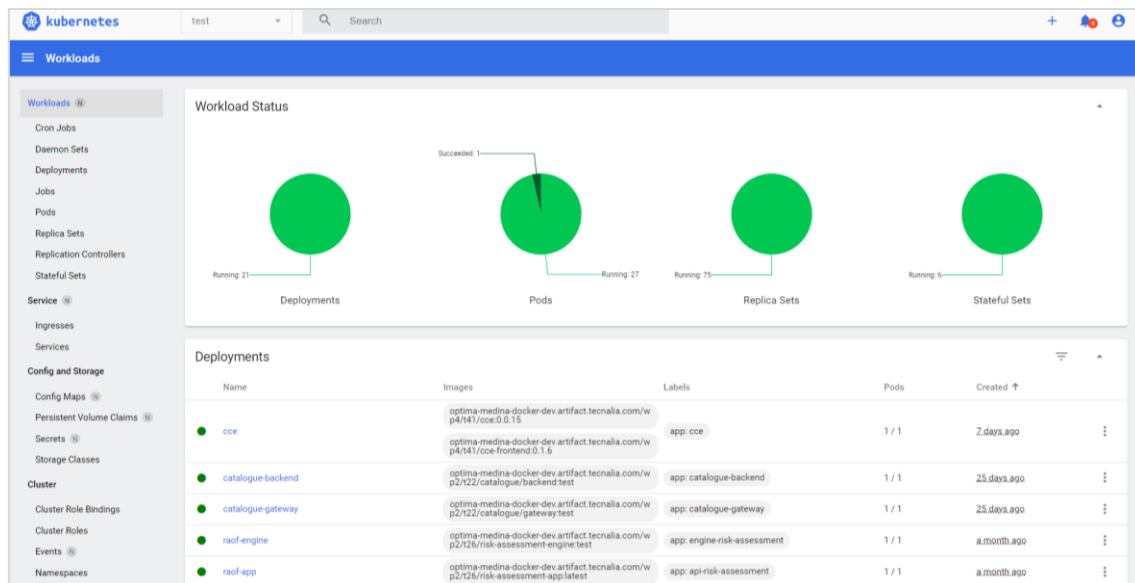


Figure 65. Kubernetes Dashboard

We have a secure Dashboard since certificates are used to expose it over HTTPS. These certificates are installed using Cert-Manager [35]. Cert-Manager automates the provisioning of certificates and provides a set of custom resources to issue certificates and attach them to services.

One of the most common use cases is securing web apps and APIs with SSL certificates from Let's Encrypt⁴⁹. Basically, we have installed Cert-Manager using the manifest file, created an issuer that uses the Let's Encrypt API for the specific domain "dashboard.k8s.medina.esilab.org" and exposed the Dashboard over HTTPS.

8.3 Hardware Infrastructure

This section describes the list of the hardware equipment used to setup the "dev" and "test" environments and the CI/CD Server Automation tool. These environments run on Virtual Machines (VM) hosted by TECNALIA and based on Ubuntu OS 20.04. The domain for all the machines is **medina.esilab.org**. The access to the virtual machines is provided via SSH protocol, using digital certificates.

The "dev" and "test" Environments are implemented on a 3-node Kubernetes cluster that virtualizes both environments, making them independent and isolated (see Figure 66). These environments run the MEDINA micro-services in containers.

⁴⁹ <https://letsencrypt.org>

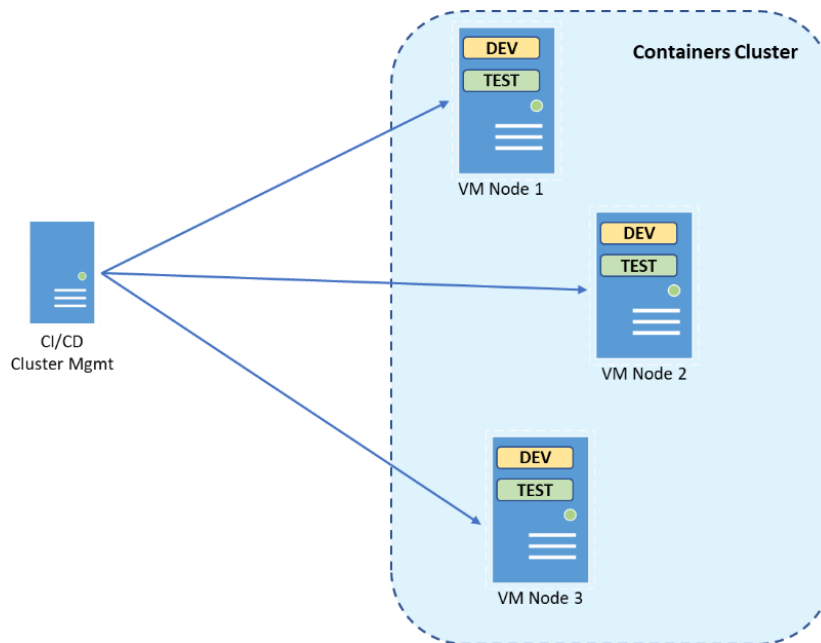


Figure 66. Kubernetes cluster on the MEDINA infrastructure

A dedicated VM hosts the CI/CD orchestration engine, the tools that support the CI/CD processes, and the Kubernetes cluster management. Its current resource status is as follows:

- RAM: 16 GB
- Cores: 4
- Hard Disk: 400 GB

The CI/CD is reachable at: ***cicd.medina.esilab.org***.

The three nodes for the Kubernetes cluster (**k8s00**, **k8s01**, **k8s02.medina.esilab.org**) share the same specifications:

- RAM: 16 GB
- Cores: 8
- Hard Disk: 200 GB + 200 GB

The 200 GB of storage of each node are organized as a distributed filesystem for data persistent layer. The Kubernetes cluster offers 200 GB of storage, and the data is duplicated among the three nodes.

An additional VM is provided for the Wazuh and VAT tools, in order to produce fake data for the MEDINA framework. The specifications are:

- RAM: 8 GB
- Cores: 4
- Hard Disk: 60 GB
- OS: Ubuntu 20.04

9 APPENDIX B: Webinars

This appendix includes a description of all the webinars that have been organized in the context of T5.3.

9.1 Docker and Kubernetes Webinar with Sample Component Integration example

In the first round, all partners manually integrated the component cluster, to be automated in the following versions of the MEDINA framework. To support all partners with this first integration, a webinar was organized in which an example project was presented.

The webinar included a part dedicated to the explanation of the main aspects and operations of Docker and Kubernetes and another part for the demonstration of all needed steps to deploy a sample project in the MEDINA environment.

The sample project, which is a spring swagger application, is available on the project's private GitLab located at TECNALIA. As shown in Figure 67, it exposes a REST API and stores data on PostgreSQL database while the Dockerfile, the Kubernetes manifests files and the README instructions are available on the repository.








Name	Last commit	Last update
 kubernetes	Added kubernetes yaml configuration files	1 month ago
 src	Initial commit	8 months ago
 .gitignore	Initial commit	8 months ago
 Dockerfile	Initial commit	8 months ago
 LICENSE	Initial commit	8 months ago
 README.md	Updated readme	1 month ago
 pom.xml	Initial commit	8 months ago

Figure 67. Spring Swagger Template on GitLab

The demo of the sample project illustrated step by step all the actions to do for the correct configuration and deployment of it, starting from the build and up to its release in the k8s cluster (see Figure 68).

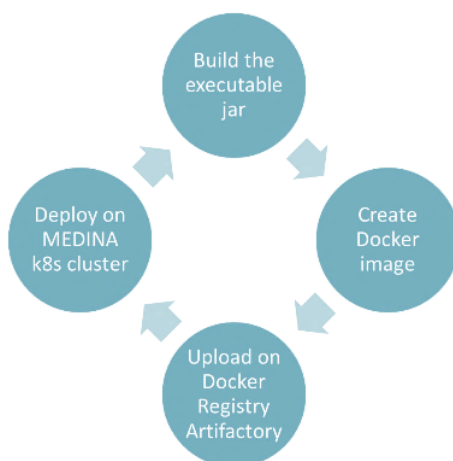


Figure 68. Sample project deployment steps

First of all, the project was packaged with Maven [36] and an executable jar is created. This jar is included in the Dockerfile for the docker image creation. Then, after the login on the private Docker Registry Artifactory, the docker image was pushed following the path convention at:

optima-medina-docker-dev.artifact.tecnalia.com/wp5/t52/springswagger-template:latest

The final step was the deployment of the docker image in the k8s cluster through the Kubernetes Dashboard. Once applied the Kubernetes manifests, the application was reachable from the internet (see Figure 69) according to this URL convention:

`<component_name>-<namespace {dev, test}>.k8s.medina.esilab.org`

For example, the access to the application in the “dev” environment is at:

<http://api-swagger-dev.k8s.medina.esilab.org/swagger-ui/index.html#/>

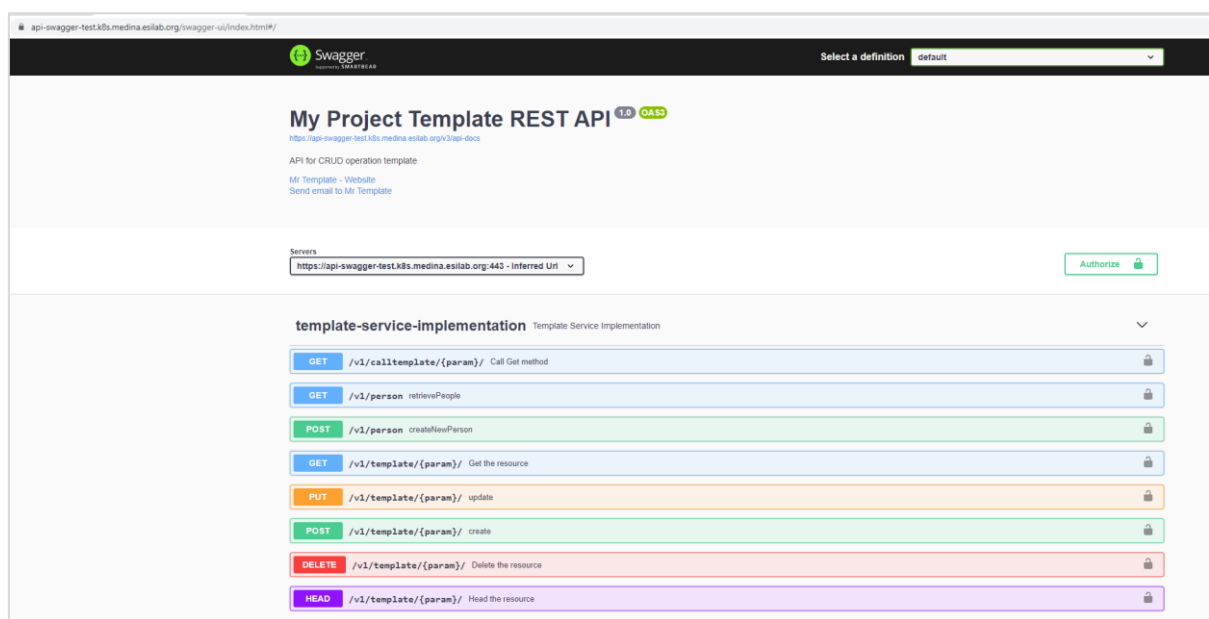


Figure 69. Demo project in the test environment

9.2 Keycloak Webinar

Keycloak [26] is an open-source identity and access management tool. It supports multiple standards, the one used in MEDINA is OpenID. Its role in MEDINA is to act as source of truth for identity and to provide login UI. The Keycloak server is reachable, for example, for the “dev” environment at this URL: <https://catalogue-keycloak-dev.k8s.medina.esilab.org/auth>. Every microservice client uses a Keycloak adapter in order to communicate with the Keycloak server.

The Keycloak webinar aimed to help partners with their micro-frontend configuration. It was divided in two parts. The first one described theoretically how Keycloak works and the flow it covers when a user initiates a request: the result is the token containing the user’s information for authentication and authorization. The second part showed a demo with a SpringBoot application for the configuration of a Keycloak adapter and the configuration on Keycloak server.

9.3 Authorization and Filtering Webinar

This webinar consisted of a demonstration about the topics of Authorization and Filtering in Keycloak for MEDINA. The first topic dealt with the configuration in Keycloak of the Composite Roles used by each component to give access permissions to endpoints within a component. In

Keycloak it is possible to manage users and roles. For example, a user without any role assigned cannot see anything in the UI, to grant permission it is necessary to define roles. These roles are defined within the Client (micro-frontend) and are only available to this Client.

The second topic was addressed by using the user-related properties obtained from the token used for authentication. These properties correspond to the token fields “cloudserviceproviderid” and “cloudserviceid” which are used to restrict the visibility of the provider (Fabasoft or Bosch) and the resources the user is interested in.

9.4 CI/CD Webinar

This webinar was focused on Continuous Integration and Continuous Delivery (CI/CD). It could be considered a second part of the first webinar dedicated to the integration with Kubernetes cluster (see section 9.1). The webinar was structured by presenting first the CI/CD environment already setup for MEDINA, then the ad-hoc pipelines developed and finally a live demo with a sample project called “springswagger-template” was shown. This example provided guidelines for partners to create their own pipelines.

9.5 Codyze Webinar

A webinar on *Codyze* is being prepared at the time of writing. The focus of the webinar is an introduction into the usage of *Codyze* to assess cloud service components in relation to the EUCS. As part of the webinar, the necessary configuration of *Codyze* is being presented. Moreover, the current integration into the MEDINA Jenkins environment is presented. This integration runs as part of the security task. It reports findings to DefectDojo which in turn provides a summary report as artefact of the security task. Partners are encouraged to include a project specific configuration file for *Codyze* into their MEDINA projects. The format and configuration options are going to be presented. In addition, the webinar will include a primer on how to write custom rulesets for *Codyze*. This primer should enable partners to have specific checks tailored towards their specific implementations.

10 APPENDIX C: Component Integration Rounds

This Appendix describes the workshops held to complete the first and the second releases of the MEDINA framework in “dev” and “test” and the status of component integration achieved.

10.1 First Round - First integration workshop

The aim of the workshop for the first round was to release the first version of the MEDINA framework in the “dev” environment of the cluster. The integration and release of components was done manually by the partners which, however, would be automated through the CI/CD pipelines in the next rounds. To carry out the integration of the components, partners were provided with access credentials to GitLab, Docker Registry Artifactory and the Kubernetes Dashboard.

During the workshop the first five actions foreseen by the defined methodology were successfully completed by all partners: first of all, each project had been uploaded to GitLab, then the Docker images had been pushed on the Artifactory registry and finally the Kubernetes manifest files had been created and applied to the “dev” environment via the Kubernetes Dashboard. At the end of the workshop, all components planned for this round were successfully released in the “dev” environment (see Figure 70).

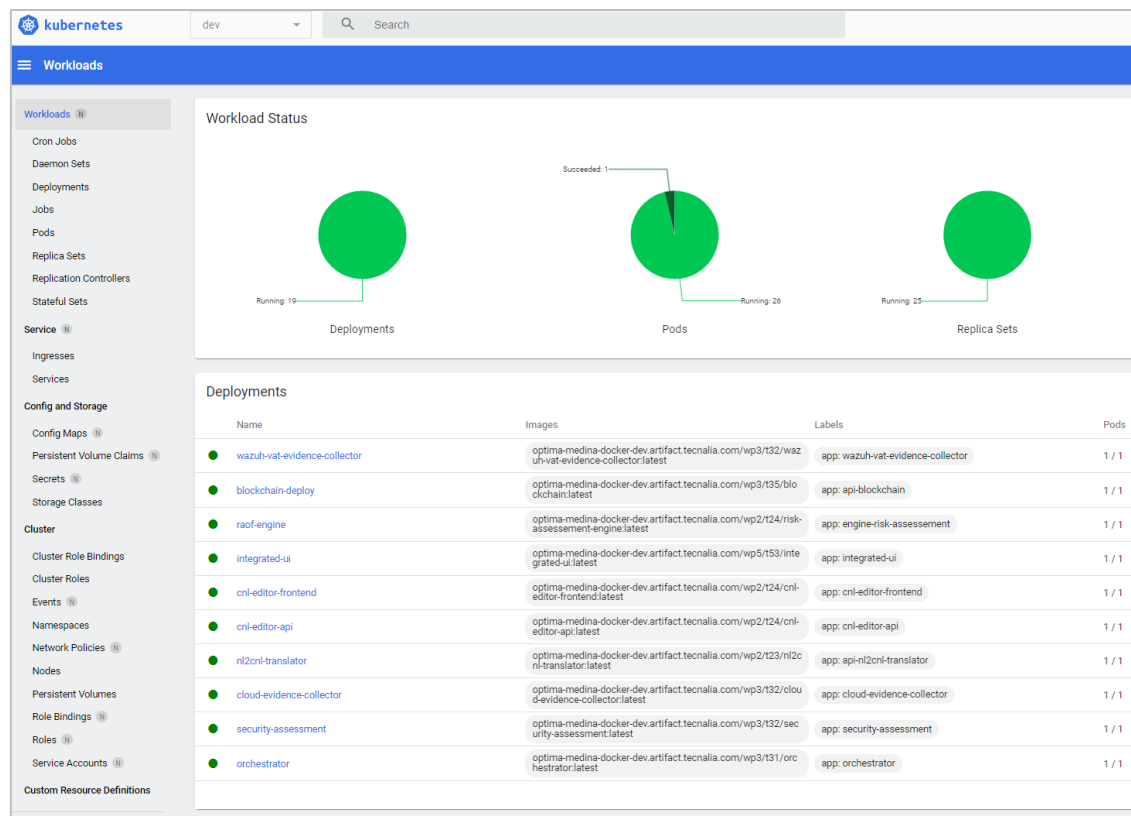


Figure 70. K8s Dashboard: Components deployed in dev environment

Figure 71 lists all the components of the MEDINA framework at the end of the first round: the green ones were released on the “dev” environment, the yellow one was deployed in the final round, and the blue ones would not be released in the Kubernetes cluster. In particular, the *Codyze* component has been integrated in the MEDINA Security pipeline and *Wazuh* and *VAT* run on a dedicated standalone VM provided by TECNALIA.

Integration Components Status										
Integration Steps										
Component	Owner (Partner)	Work Package	Task	TECNALIA GitLab	Containerization	K8s file	OpenAPI specs	Push to Docker Registry	Deploy Dev	Deploy Test
CNL Editor	HPE	WP2	T2.4	yes	yes	yes	yes	yes	yes	no
Metrics and measures catalogue	TECNALIA	WP2	T2.2	yes	yes	yes	yes	yes	yes	no
NL2CNL Translator	CNR/Fabasoft	WP2	T2.3	yes	yes	yes	yes	yes	yes	no
DSL Mapper	CNR/Fabasoft	WP2	T2.5	yes	yes	yes	yes	yes	yes	no
Cloud Evidence Collector (Clouditor)	FhG	WP3	T3.2	yes	yes	yes	yes	yes	yes	no
Security Assessment (Clouditor)	FhG	WP3	T3.2	yes	yes	yes	yes	yes	yes	no
Orchestrator (Clouditor)	FhG	WP3	T3.1	yes	yes	yes	yes	yes	yes	no
Codyze	FhG	WP3	T3.3	yes (partly)	yes	\	no	yes	no (integrated Jenkins)	no
Blockchain Monitoring Tool	TECNALIA	WP3	T3.5	no (proprietary component)	yes	yes	yes (partially)	yes	yes	no
Static Risk Assessment and Optimisation Framework	CNR	WP2	T2.4	yes	yes	yes	yes	yes	yes	no
Dynamic Risk Assessment and Optimisation Framework	CNR	WP4	T4.4	no	\	\	\	\	\	\
Wazuh + VAT evidence collector (interface to sec.ass.)	XLAB	WP3	T3.2	yes	yes	no	no	yes	no	no
Wazuh & VAT proprietary	XLAB	WP3	T3.2	no (proprietary component)	no	\	no	no	no (standalone VM)	no
Continuous Certification Evaluation	XLAB	WP4	T4.1	yes	yes	yes	no	yes	yes	no
Life Cycle Manager	FhG	WP4	T4.3	yes	yes	yes	no	yes	no	no
Organisational evidence management tool	Fabasoft	WP3	T3.4	no	no	no	no	no	no	no
Integration UI	HPE	WP5	T5.3	yes	yes	yes	no	yes	yes	no

Figure 71. Status of the first integration of the MEDINA components

Furthermore, partners performed point-to-point tests to verify the communication in pairs of the released components. Table 37 shows the working ones in green.

Table 37. Status of point-to-point connections during the first round

Component Name	Component Name	Status
Orchestrator	Continuous Certification Evaluation	CONNECTED
Orchestrator	Trustworthiness System	CONNECTED
Orchestrator	Security Assessment	CONNECTED
Orchestrator	Catalogue of Controls and Metrics	NEXT ROUND
Cloud Evidence Collector	Security Assessment	CONNECTED
Security Assessment	Wazuh + VAT Evidence Collector	CONNECTED
DSL Mapper	Orchestrator	NEXT ROUND
DSL Mapper	Catalogue of Controls and Metrics	NEXT ROUND
NL2CNL Translator	Catalogue of Controls and Metrics	NEXT ROUND
CNL Editor	DSL Mapper	NEXT ROUND
CNL Editor	NL2CNL Translator	NEXT ROUND
CNL Editor	Catalogue of Controls and Metrics	NEXT ROUND
AMOE	Catalogue of Controls and Metrics	NEXT ROUND
Static Risk Assessment and Optimisation Framework	Catalogue of Controls and Metrics	NEXT ROUND
Continuous Certification Evaluation	Catalogue of Controls and Metrics	NEXT ROUND
Continuous Certification Evaluation	Dynamic Risk Assessment and Optimisation Framework	NEXT ROUND
Dynamic Risk Assessment and Optimisation Framework	Life Cycle Manager	NEXT ROUND
Integrated UI	Catalogue of Controls and Metrics Keycloak	CONNECTED
Integrated UI	Catalogue of Controls and Metrics	CONNECTED
Integrated UI	NL2CNL Translator	CONNECTED
Integrated UI	Orchestrator	NEXT ROUND
AMOE	Orchestrator	NEXT ROUND
Integrated UI	AMOE	NEXT ROUND

10.2 Second Round – Continuous Integration

During the second round, we did not have a dedicated workshop session but the partners continuously integrated and updated their components. Thus, all component owners implemented the CI/CD pipelines, which allowed the partners to automatically release them in the Kubernetes cluster. Section 2.2 describes the strategy and implementation of the CI/CD pipelines.

One of the goals we reached during the second round is the integration of the MEDINA components into the Kubernetes cluster. In fact, the AMOE component was not integrated during the first workshop session and the SSI Framework component was introduced in recent months. AMOE and SSI are now deployed in the “dev” and “test” environments.

Figure 72 lists all the components of the MEDINA framework at the end of the second round: the green ones were released in the Development and Test environments and the blue ones would not be released in the Kubernetes cluster. The *Codyze* component has been integrated in the MEDINA Security pipeline and *Wazuh* and *VAT* run in a dedicated standalone VM provided

by TECNALIA. Interested readers can see the progress of the integration of the MEDINA components by comparing with the previous status of integration in M15, that is shown in Figure 71 in the Appendix *First Round - First integration workshop*.

INTEGRATION COMPONENTS STATUS										
Component Name	Owner (Partner)	Work Package	Task	TECNALIA Private GitLab	Containerization	K8s file	OpenAPI specs	Integration Steps		
								Push to Docker Registry	Deploy Dev	Deploy Test
CNL Editor	HPE	WP2	T2.4	yes	yes	yes	yes	yes	yes	yes
Metrics and measures catalogue	TECNALIA	WP2	T2.2	yes	yes	yes	yes	yes	yes	yes
NL2CNL Translator	CNR/Fabasoft	WP2	T2.3	yes	yes	yes	yes	yes	yes	yes
DSL Mapper	CNR/Fabasoft	WP2	T2.5	yes	yes	yes	yes	yes	yes	yes
Cloud Evidence Collector (Clouditor)	FhG	WP3	T3.2	yes	yes	yes	yes	yes	yes	yes
Security Assessment (Clouditor)	FhG	WP3	T3.2	yes	yes	yes	yes	yes	yes	yes
Orchestrator (Clouditor)	FhG	WP3	T3.1	yes	yes	yes	yes	yes	yes	yes
Codyze	FhG	WP3	T3.3	yes (partly)	yes	\	\	yes	no (integrated Jenkins)	no (integrated Jenkir)
Blockchain Monitoring Tool	TECNALIA	WP3	T3.5	no (proprietary component)	yes	yes	yes (partially)	yes	yes	yes
Static Risk Assessment and Optimisation Framework	CNR	WP2	T2.4	yes	yes	yes	yes	yes	yes	yes
Dynamic Risk Assessment and Optimisation Framework	CNR	WP4	T4.4	yes	\	\	\	no	\	\
Wazuh + VAT evidence collector (interface to sec.ass.)	XLAB	WP3	T3.2	yes	yes	no	no (uses clouditor's API)	yes	no	no
Wazuh & VAT proprietary	XLAB	WP3	T3.2	no (proprietary component)	no	\	no (uses clouditor's API)	yes	no (dedicated VM)	no (dedicated VM)
Continuous Certification Evaluation	XLAB	WP4	T4.1	yes	yes	yes	yes	yes	yes	yes
Life Cycle Manager	FhG	WP4	T4.3	yes	yes	yes	yes	yes	yes	yes
Assessment and management of organisational evidences (AMOE)	Fabasoft	WP3	T3.4	yes	yes	yes	partially	yes	yes	yes
Integration UI	HPE	WP5	T5.3	yes	yes	yes	no apis	yes	yes	yes
Self-Sovereign Identity (SSI)	TECNALIA	WP4	T4.3	yes	yes	no	yes	yes	yes	yes

Figure 72. Status of the second integration of the MEDINA components

The last four actions foreseen by the defined methodology in section 2.1.1 were successfully completed by all partners: first of all, each project was released in the Kubernetes “test” environment and the standalone and point-to-point tests were performed, finally the Use Cases tested the end to end scenarios verifying that the workflows described in section 3 were working properly in their own “Validation” environment. Further details on the validation of the workflows can be found in D5.2 [3] and in D6.3 [9].

During the regular bi-weekly WP5 meetings we checked the status of the components and the updates of the point-to-point connections. Table 38 shows the status of these connections as follows:

- Light green: the connection was implemented during the first round
- Dark green: the connection was successfully implemented during the second round
- Orange: the connection was in progress
- Grey: the connection was no longer needed

Comparing the contents of Table 38 with the previous status shown in Table 37, we can see that most of the point-to-point connections were completed: 20 connections were implemented in addition to the previous 6, 3 connections were discarded and 3 connections were still in progress.

Table 38. Status of point-to-point connections during the second round

Component Name A	Component Name B	Status
Orchestrator	Continuous Certification Evaluation	CONNECTED
Orchestrator	Trustworthiness System	CONNECTED
Orchestrator	Security Assessment	CONNECTED
Orchestrator	Catalogue of Controls and Metrics	CONNECTED
Orchestrator	NL2CNL Translator	CONNECTED
Codyze	Orchestrator	CONNECTED
Cloud Evidence Collector	Security Assessment	CONNECTED
Security Assessment	Evidence Collection from VAT	IN PROGRESS
Security Assessment	Evidence Collection from Wazuh	CONNECTED
DSL Mapper	Orchestrator	CONNECTED
DSL Mapper	Catalogue of Controls and Metrics	DISCARDED
NL2CNL Translator	Catalogue of Controls and Metrics	CONNECTED
NL2CNL Translator	CNL Editor	CONNECTED
CNL Editor	DSL Mapper	CONNECTED
CNL Editor	Catalogue of Controls & Metrics	DISCARDED
AMOE	Catalogue of Controls and Metrics	CONNECTED
AMOE	Orchestrator	CONNECTED
Catalogue of Controls and Metrics	Static Risk Assessment and Optimisation Framework	IN PROGRESS
Continuous Certification Evaluation	Catalogue of Controls and Metrics	CONNECTED
Continuous Certification Evaluation	Dynamic Risk Assessment and Optimisation Framework	CONNECTED
Continuous Certification Evaluation	Life Cycle Manager	CONNECTED
Dynamic Risk Assessment and Optimisation Framework	Life Cycle Manager	CONNECTED

Component Name A	Component Name B	Status
AMOE	Orchestrator	CONNECTED
SSI Framework	Life Cycle Manager	CONNECTED
Integrated UI	Catalogue of Controls and Metrics	CONNECTED
Integrated UI	NL2CNL Translator	DISCARDED
Integrated UI	Orchestrator	CONNECTED
Integrated UI	CNL Editor	CONNECTED
Integrated UI	SSI Framework	IN PROGRESS
Integrated UI	Static Risk Assessment and Optimization Framework	CONNECTED
Integrated UI	Continuous Certification Evaluation	CONNECTED
Integrated UI	AMOE	CONNECTED

11 APPENDIX D: Pipelines

This appendix describes the implementation of the CI/CD solution that is put in place for supporting the MEDINA framework through the pipelines' schema.

The implemented pipelines at M15 were three, named **Build** pipeline, **Deploy** pipeline and **Security** pipeline. These pipelines are called following a hierarchy: the Build pipeline is triggered automatically at every push of a project in the MEDINA public GitLab and automatizes the build of the project, the creation of the Docker image and its push on the TECNALIA Artifactory. Then, if the previous pipeline succeeds, without any errors, the second Deploy pipeline is triggered that will automatically deploy the component to the “dev” environment by default. Finally, the Security pipeline starts automatically if the Build and the Deploy pipelines succeed.

As described in D5.2 [3], to automate the deployment process we make use of the Jenkins Seed Job that automatically creates the pipelines for each component of the MEDINA framework. This is a plugin that consists in filling a form by entering parameters such the software repository URL where to retrieve the source code, the container file descriptor (in Docker format), the generated container image for publishing to an internal private registry and a list of one or more Kubernetes deployment manifest files.

This procedure is quite the same for all components because all the CI/CD tools involved are organized to simplify the deployment with a convention agreed by the consortium. The GitLab repository is divided into groups that are folders which contain the projects. The structure reflects the Work Package and Tasks division of the MEDINA project. Also, Jenkins and Artifactory are organized following this convention.

All these concepts and steps were described during the CI/CD Webinar (see section 9.4) using a Demo with the sample project “springswagger-template”. First of all, a new project named “springswagger-template” was created in GitLab. The Jenkins Seed Job could then be run by filling it with the parameters customized for the project.

Following there is the description of these parameters and an example of how to compile the form to create the specific pipelines for the project “springswagger-template”. Figure 73 shows these parameters:

- **Work Packages/Task folder**, where the Jenkins Jobs will be created. We can choose the correct path from the picklist that is previous created in Jenkins. Select “wp5/task_5.2”.
- **Job basename**, i.e., the component name: for example, springswagger-template.
- **GitLab URL**, retrieved from the TECNALIA GitLab web interface, is the source code repository for the project.
- **GitLab branch**, is the default “master”.
- **Build template**, chosen from a preconfigured template, can be empty or customized with a build automation tool like Maven. Select “Maven”.
- **Docker file**, the name of the dockerfile that contains the instructions to build the container image. In this case the folder in which is the file is “docker” and the name of the file is “Dockerfile”.
- **Image**, the name of the container image pushed to the private registry, which is the Artifactory owned by TECNALIA. The image will have the absolute path, for example: “wp5/t52/springswagger-template”.
- **Kubernetes manifests**, the yaml files used for the deployment in the Kubernetes cluster, which are contained in GitLab folder “113ubernetes”.

Once these details are provided, the Seed Job automatically creates the three pipelines (Build, Deploy and Security) for the “springswagger-template” in the selected folder (see Figure 74).

Project HPE-MEDINA-seed-job

This build requires parameters:

WPT_FOLDER

wp5/task_5.2

Please specify the Work Package/Task folder where the Jenkins job(s) will be created.

JOB_BASENAME

springswagger-template

Please specify the name of the job, typically the component name, e.g. springswagger-template
Other jobs will be automatically created, e.g. use spring-swagger to create springswagger-template-[build,deploy,security] jobs.

GITLAB_URL

git@git.code.tecnalia.com:medina/wp5/task_5.2/springswagger-template.git

Please specify the git repository. Just copy the git url from GitLab web interface (Clone with SSH).
E.g. git@git.code.tecnalia.com:medina/wp5/task_5.2/springswagger-template.git

GITLAB_BRANCH

master

Please specify the git branch if not the default 'master'.
E.g. main

BUILD_TEMPLATE

maven

Please specify a build template. Select 'empty' if not other choice apply and you will have to customize manually the build job.
E.g., 'maven' will setup stages for mvn compile / test / package

DOCKER_FILE

Dockerfile

Please specify the name of the dockerfile to build your container image, e.g. Dockerfile.

IMAGE

wp5/t52/springswagger-templat

Please specify the name of the container image (w/o tag), e.g. wp5/t52/springswagger-template
The image will be pushed to the private registry at job build time.
The tag 'latest' tag is always used, but you can set another tag when you manually run the generated build job.

YAML_FILES

kubernetes/api-swagger-deployment.yaml
kubernetes/api-swagger-ingress.yaml
kubernetes/api-swagger-svc.yaml<

Please specify the list of yaml files to deploy the build in a multi-line format. Files are relative to source code directory and path can be specified. E.g.:
kubernetes/api-swagger-deployment.yaml
kubernetes/api-swagger-ingress.yaml
kubernetes/api-swagger-svc.yaml

Build

Figure 73. Jenkins Seed Job

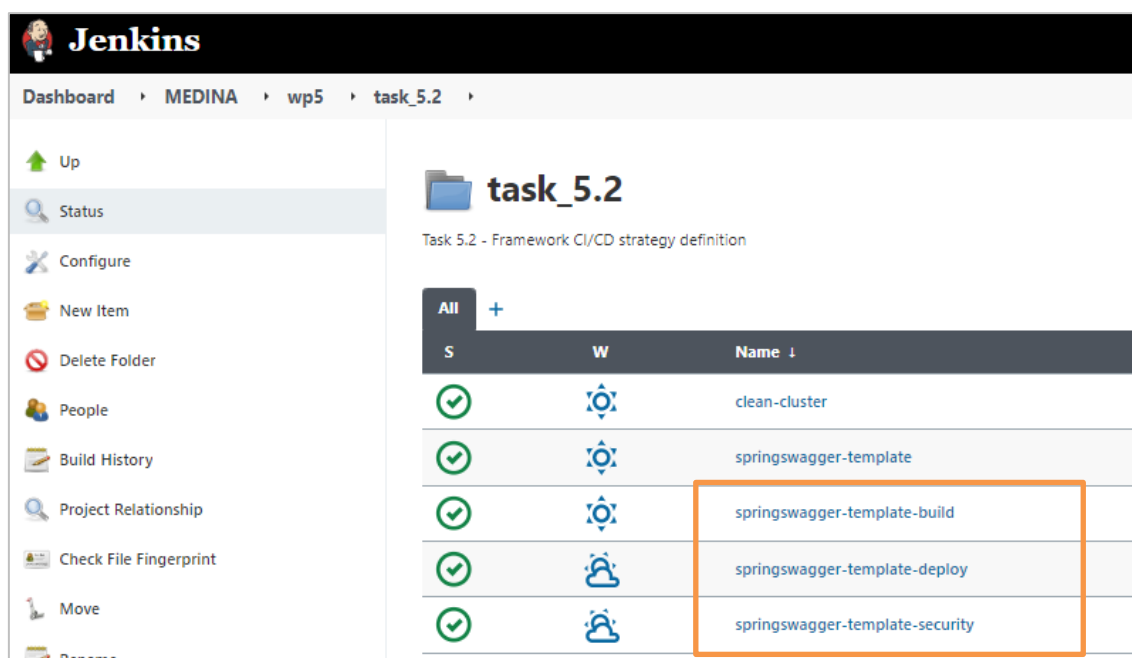


Figure 74. Pipelines

During the CI/CD Webinar demo (see section 9.4) it was shown how the creation of the pipelines flows through their stages after configuring and building the Jenkins Seed Job. Every pipeline has several stages, with a name describing what they have done.

As described theoretically in the CI/CD strategy in D5.2 [3], the Build pipeline foresees stages where the code is checked out from GitLab and the docker container is setup to execute the other build stages (see Figure 75). These stages are Compile, Testing and Package, and they can be different depending on the Build template field selected in the Seed Job before running it. In this case we have selected Maven, so “*mvn*” commands are executed. The next three stages are referred to the Docker image building and pushing to the Artifactory repository. By default, the image is pushed with the “latest” tag but there is an optional phase to tag it differently. At last, if no errors occur the Deploy Job is automatically called.

Stage View													
	Checkout Code	Setup Build Container	Compile	Testing	Package	Manage Container	Build Container Image	Push Container Latest Image	Optional Tag and Push Container	Clean-up Built Container Image	Call Deploy Job	Archive Artifacts	Declarative: Post Actions
Average stage times (Average full run time: ~3min 56s)	679ms	1s	5s	5s	3s	404ms	2s	3min 15s	0ms	597ms	13s	371ms	365ms
Oct 14 16:39 1s commit	737ms	1s	8s	5s	4s	435ms	3s	7s		426ms	17s	489ms	430ms

Figure 75. Build pipeline

The Deploy pipeline deals with the release of the components in the Kubernetes cluster. As described in Section 2.1, the Kubernetes cluster is divided in two isolated and virtual environments, “dev” and “test”. The stages of this pipeline (see Figure 76) include first the step where Jenkins accesses to the Kubernetes cluster with exchanged credentials, and then the step in which the Kubernetes manifests files are applied to release the configuration to the environment. By default, the Deploy pipeline releases the component on the “dev” environment.

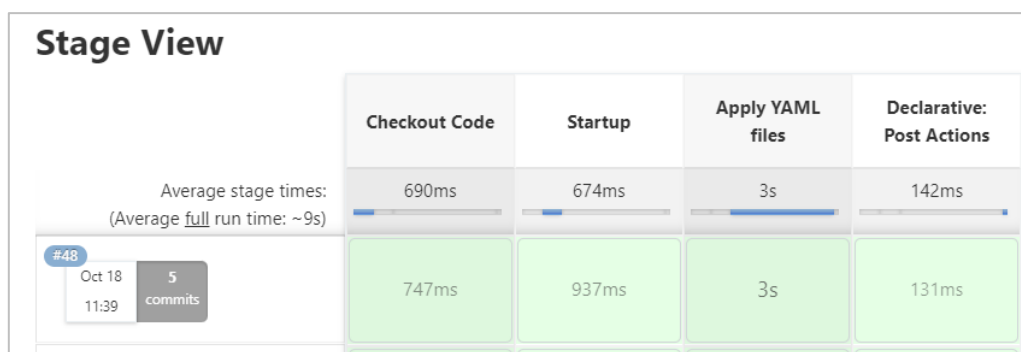


Figure 76. Deploy pipeline

Partners can also use this pipeline to manually release the component on the “test” environment changing it with one click from the Deploy pipeline, rebuilding the pipeline and choosing among the available environments, as shown in Figure 77.

Pipeline integrated-ui-deploy

This build requires parameters:

PRJ_ENV
 dev (selected)
 test
 Environment for deployment: dev - Development, test - Test

PRJ_IMAGE_TAG
 latest
 Specify the tag for the component docker image, e.g. latest, 1.0.0, etc.

YAMLS_OVERRIDE

(optional) Please specify a list of yaml files to deploy the build in a multi-line format. If the list is empty, it will use the default files you set in the seed job. If you specify a value in this field, it will not consider any default yaml file you specified in the seed job. If present, the placeholder {time} inside the yaml files will always be replaced with a timestamp. If there is a 'host' field in the yaml files and the hostname part follows the format my-hostname-dev.domain.org, the 'dev' part will be replaced by the PRJ_ENV (e.g., my-hostname-test.domain.org)
 Files are relative to source code directory and path can be specified. E.g.:
 kubernetes/api-swagger-deployment-test.yaml
 kubernetes/api-swagger-ingress-test.yaml
 kubernetes/api-swagger-svc-test.yaml

Build

Figure 77. Deploy pipeline with available ENV

The Security pipeline is automatically triggered upon a successful Build and Deploy. This pipeline includes various steps (shown in Figure 78) representing the different types of security analysis performed: Static Code analysis for checking the source code, Container security for scanning vulnerabilities into the container packages, and Software Composition Analysis (SCA) for spotting security issues in third party libraries.

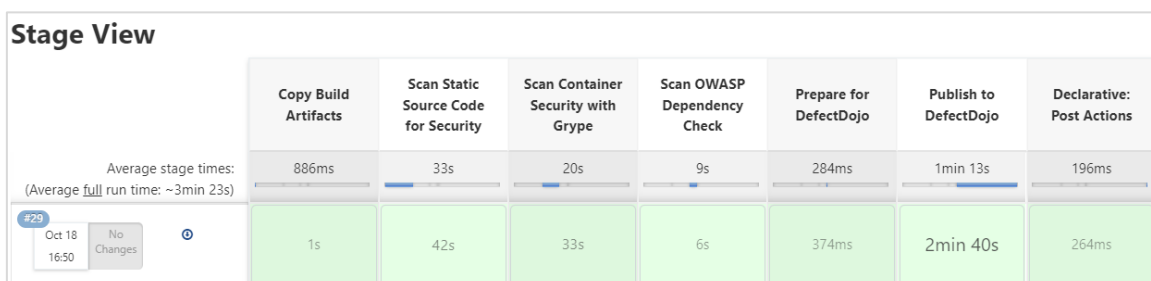


Figure 78. Security pipeline

The two first security controls are performed respectively by Semgrep and Anchore. These tools are running into containers called in the security pipeline. Once the scanning is done, these containers, in which the tools are installed, are destroyed but the output file of the analysis

persists. The advantage of this choice to use the container lives in the fact that it is possible to fast and easily update the tool to the latest version, forcing the download of the latest tag of the container images.

Regarding the third security control, SCA, the tool that performs this analysis is OWASP Dependency Check, installed via command line. In the latest stages of this Security pipeline a report is prepared, that collects all the analysis outputs of the previous stages, and finally is published to DefectDojo, the vulnerability report aggregator tool adopted to make possible to see all the analysis results in a unique view. The report is visible directly inside Jenkins, but DefectDojo provides a graphical interface with several metrics and dashboards to analyse the results using different parameters, such as the time or the severity of the vulnerabilities.

In addition to the three pipelines available in M15, a new pipeline called “clean-cluster” was added in M27 to deal with dangling docker images that caused no disk space to be left. Figure 79 shows the stages that compose this pipeline. Basically, the dangling docker images are listed, then removed, and finally the disk space is shown.

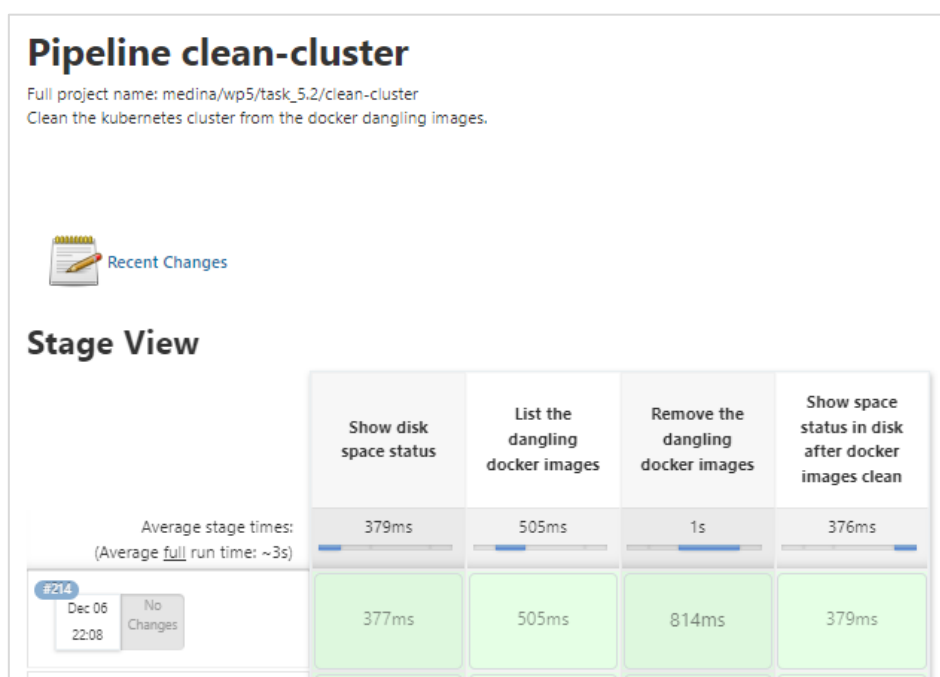


Figure 79. Clean-cluster pipeline

All these steps provide an example of how to use the CI/CD tools to adopt the SecDevOps approach in MEDINA. The aim was to give guidelines to partners to enable a conventional way of using the overall infrastructure that is setup.

12 APPENDIX E: Generic Architectural Workflows

This Appendix revisits and updates the details related to the generic architectural workflows as presented in D5.4 [2], and related to the same roles which are being presented in section 3 of the present deliverable.

12.1 WF1 - Preparation of Target of Certification (ToC)

This initial workflow, despite not invoking any of the MEDINA components, is an evident prerequisite for the CSP to fulfil before the certification process starts. Its main goal is for the CSP to prepare the Target of Certification (ToC), both from a technical (e.g., deploying the actual cloud service in the hyperscaler) and organizational (e.g., gather the operational manuals in electronic format) perspectives.

12.1.1 Related Architectural Components

As mentioned above, this workflow does not involve any of the MEDINA components. However, it setups the ToC elements in building blocks 5 and 7 from Figure 16, namely:

- ToC's organizational evidence (electronic format)
- Cloud services comprising the ToC (e.g., IaaS/PaaS/SaaS), which can be deployed in one or more hyperscalers.

12.1.2 Workflow

Table 39 describes the steps associated to this workflow. Please notice that in this case, the role "CSP" refers to any of the internal ones defined in section 3.3.1, namely:

- IT Security Governance
- Security Analyst
- Domain Governance
- Product and Service Owner
- Product (Security) Engineer
- Chief Information Security Office (CISO)
- Auditor (only internal)

Table 39. WF1 description

Step	Description	Role	Comments
1	Documentation related to organizational measures implemented by the Cloud Service is gathered and made available in electronic format.	CSP	The documentation can be made available in portable formats like PDF.
2	All Resources that comprise the Cloud Service/ToC (VMs, SQL, Web Apps, SaaS, etc.) are deployed in the hyperscaler.	CSP	Using the hyperscaler's native interfaces, the corresponding resources (i.e., the ones comprising the ToC) are deployed.

12.2 WF2 - Preparation of MEDINA Components

The second generic workflow of the architecture (WF2) refers to the actual configuration and deployment of those MEDINA components which are needed for certifying the Cloud Service. This WF2 does not perform any actual assessment, but it is a required set of deploying actions before the certification process is triggered by WF3.

12.2.1 Related Architectural Components

This workflow involves the components in building blocks 1, 2, 7 and 8 from Figure 16, namely:

- Catalogue of Controls and Metrics
- Organizational Evidence Gathering and Processing
- Security Assessment (CS Level and OS) – Clouditor Assessment
- Evidence Collection / Security Assessment CS level and CSP Native (Azure Policies)
- Evidence Collection / Security Assessment Application Level (Codyze)
- Evidence Collection Wazuh
- Evidence Collection VAT
- Company Compliance Dashboard / Integrated UI

12.2.2 Workflow

Table 40 describes the steps associated to this workflow. Once again, the CSP role relates to any of the following internal ones defined in section 3.3.2, namely:

- IT Security Governance
- Security Analyst
- Domain Governance
- Product and Service Owner
- Product (Security) Engineer
- Chief Information Security Office (CISO)
- Auditor (only internal)

Table 40. WF2 description

Step	Description	Role	Comments
1	Configuring the following settings in the Company Compliance Dashboard / Integrated UI: <ol style="list-style-type: none"> SSO integration Setup users and roles Add cloud service IDs to the respective users. 	CSP	The Integrated UI provides the entry point to the MEDINA framework, and as such it needs to become integral part of the CSP's systems. Therefore, actions like SSO integration are needed. A role-based authorization model allows MEDINA users to only perform specific actions.
2	Setting up the Catalogue of Controls and Metrics: <ol style="list-style-type: none"> Configure the EUCS catalogue with all assurance levels, and including corresponding controls/requirements/metrics. Check that there is a list of controls that are similar in other security frameworks (e.g., C5:2020, SecNumCloud, ISO/IEC 27002, ISO/IEC 27017 and Cisco CCF) Check that there exist an Implementation Guidance for each of the EUCS high level requirements related to continuous monitoring. 	MEDINA ⁵⁰	The Catalogue of Controls and Metrics is prefilled with EUCS and related information (including mappings and guidelines), so it comes out-of-the-box for the CSP (see WF3).

⁵⁰ This role means the actual MEDINA framework (non-human role).

Step	Description	Role	Comments
3	Configuring the Clouditor Evidence Collector (IaaS, PaaS): a. Clouditor's Service Principal / Technical User is configured and deployed on the ToC	CSP	Please refer to D6.3 Appendix B [9].
4	Configuration of CSP Native Evidence Collector (IaaS, PaaS): a. CSP-Native is part of the default Clouditor configuration for Azure	CSP	In analogy to the collector described in Step 3, this CSP-Native one is used to gather evidence from technical measures.
5	Configuration of Security Assessment Application Level (Codyze) Evidence Collector (SaaS): a. Codyze is configured	CSP	Used to gather evidence from technical measures (code-level).
6	Configuration of Wazuh and VAT Evidence Collectors (IaaS): a. Wazuh is configured b. VAT is configured	CSP	Used to gather evidence from technical measures.

12.3 WF3 - EUCS deployment on ToC

After the ToC has been deployed on the hyperscaler (WF1) and the corresponding MEDINA components were configured/deployed by the CSP (WF2), then it is possible to use the later for certifying the Cloud Service. That is the goal of this WF3.

12.3.1 Related Architectural Components

This workflow involves the components in building blocks 1, 2, 5 and 6 from Figure 16, namely:

- Catalogue of Controls and Metrics
- CNL Editor
- Organizational Evidence Gathering and Processing
- Orchestrator / Clouditor Orchestrator

12.3.2 Workflow

Table 41 describes the steps associated to this workflow. As in previous WFs, the CSP role refers to the following internal ones defined in section 3.3.3, namely:

- IT Security Governance
- Security Analyst
- Domain Governance
- Product and Service Owner
- Product (Security) Engineer
- Chief Information Security Office (CISO)
- Auditor (only internal)

Please notice that in the case of Step 3, only the Product (Security) Engineer has the permissions to manage the organizational evidence in MEDINA's AMOE component (except evaluation, which will be discussed afterwards).

Table 41. WF3 description

Step	Description	Role	Comments
1	<p>The Orchestrator UI is used to perform the following actions:</p> <ul style="list-style-type: none"> a. In the “Cloud Services” tab the corresponding cloud service is selected b. General information is checked on the “Overview” tab c. Both General and Target of Evaluation⁵¹ information is configured for the cloud service (“Configuration” tab) d. If the Target Values of a specific set of Metrics need to be customized, then the corresponding Requirements are selected from the “Configuration” tab (Target of Evaluation -> Configure Controls in Scope). 	CSP	<p>Each Resource is automatically discovered and assigned to its corresponding cloud service.</p> <p>In the current version of the framework, the assignment of Resources into Cloud Services is manually performed and hardcoded in the Cloudditor.</p>
2	<p>The UI from the CNL Editor (Customization of Requirements) is used to customize any Requirement selected in the Orchestrator (previous step):</p> <ul style="list-style-type: none"> e. From the List of displayed REOs, select the Requirement which Obligations need to be customized. f. On the Edit window, modify the target value(s) of the metric(s) to customize. g. Finalize the REO creation process (Edit → Complete). 	CSP	<p>Once the corresponding Obligations have been selected and configured with a Target Value, then they are ready to be stored along with the ToC information in MEDINA’s Orchestrator (in the form of a REO).</p>
3	<p>The Organizational Evidence Assessment is used to upload the collected documentation of the ToC (see WF1)</p>	CSP	<p>These documents (PDF) are stored directly on the database of the component, and not on the Orchestrator’s.</p> <p>Assessment results and associated evidence can be forwarded to the Orchestrator on demand (see WF5).</p>

12.4 WF4 - EUCS Preparedness – ToC Self-Assessment

This workflow relates to the components in charge of performing the static risk management (SATRA) and the EUCS self-assessment (Catalogue of Controls and Metrics - Questionnaire) as documented by D2.8 [12] and D2.2 [13] respectively. Although SATRA implements a “stand alone functionality”, which does not need to be technically deployed in the Cloud Service (cf. WF3), it is integrated into the whole MEDINA framework thanks to the Integrated UI.

12.4.1 Related Architectural Components

This workflow involves the components in building blocks 1 and 3 from Figure 16, namely:

- Risk Assessment and Optimization Framework

⁵¹ As mentioned in WF2, the MEDINA framework support EUCS as default certification scheme.

- Catalogue of Controls and Metrics - Questionnaire

12.4.2 Workflow

The related activities in WF4 are described in Table 42. In this case besides the internal CSP roles defined in section 3.3.4, namely:

- IT Security Governance
- Security Analyst
- Domain Governance
- Product and Service Owner
- Product (Security) Engineer
- Chief Information Security Office (CISO)
- Auditor (only internal)

we also consider external auditor roles or CABs (Conformance Assessment Bodies) which can take over the final assessment of evidence using the corresponding MEDINA tools.

Table 42. WF4 description

Step	Description	Role	Comments
1	<p>In the Catalogue of Controls and Metrics UI:</p> <ul style="list-style-type: none"> a. Select the “Questionnaires” menu option, then either fill in the details of a New questionnaire (cloud service name, assurance level), load an existing one, or proceed to generate an audit report⁵². b. Provide answers to the questions for each requirement, based on any of the following potential answers: <ul style="list-style-type: none"> • Fully supported • Partially supported • Not supported at all • Not applicable c. Provide some evidence to support the answers to the questionnaire d. Include some comments e. Save the questionnaire f. Generate the report 	CSP	<p>The tool is based on a questionnaire interface containing requirements from EUCS, just as described in the referenced D2.2 [13].</p> <p>A closed set of possible answers guarantees the computation of a degree of compliance, which represents the CSP’s level of preparedness for obtaining an EUCS certificate.</p> <p>The format of the audit report is presented in D2.2 [13].</p>
2	<p>In the Catalogue of Controls and Metrics UI:</p> <ul style="list-style-type: none"> a. Assess the service to identify the level of conformity b. Include non-conformities, if needed c. Save the questionnaire d. Generate the report 	CSP (internal auditor) or CAB	<p>The format of the audit report is presented in D2.2 [13].</p>

⁵² Corresponding functionality depends on the user’s assigned MEDINA role, which will be described in Deliverable D6.4 [10]

Step	Description	Role	Comments
3	In the Catalogue of Controls and Metrics UI: a. The compliance result for each requirement is calculated based on the answers provided for its related questions. b. The compliance results are sent to the SATRA end point.	CSP	The compliance results are sent to SATRA each time the user exits the questionnaire.
4	In the Risk Assessment and Optimization Framework UI: a. A Target of Certification is defined, by selecting the cloud service layer, certification scheme and assurance level. b. The “Conduct static risk assessment” option is selected. c. The SATRA questionnaire is automatically filled in with the answers provided in the Catalogue - Questionnaire ⁵³ . d. Asset information is entered (identification, number of units, C/I/A-impact levels). e. ToC information and Impact level (per-Resource type) are entered into the tool.	CSP	The ToC information required for the static risk assessment is manually entered into the tool (contrary to the automated discovery of Resources in WF3), mostly because less granular details are needed for the preparedness assessment. For example, details about the actual Resources’ configuration are not needed for this static assessment.
5	The Risk Assessment and Optimization Framework UI computes and reports: a. The degree of compliance for each requirement, based on the CSP’s answers to the questionnaire. b. The Suggested Optimization functionality.	MEDINA	The preparedness report includes the identification of major and minor non-conformities, and comparison between the ideal conformity case and the provided CSP answers. More details are presented in D2.8 [12].

12.5 WF5 - EUCS Compliance Assessment

MEDINA proposes the notion of “continuous audit-based certification”, which departs from the EUCS definition of “continuous (automated) monitoring” referring to **periodically assessing the ToC**. This WF5 describes **discrete compliance assessments**, which should then be periodically executed for the MEDINA framework to start the certification lifecycle (cf. WF6).

Further information about the underlying evidence collection mechanisms can be found in D3.3 [19].

12.5.1 Related Architectural Components

This workflow involves the components shown in building blocks 5 and 7 from Figure 16, namely:

- Organizational Evidence Gathering and Processing
- Security Assessment (CS Level and OS) – Clouditor Assessment
- Evidence Collection / Security Assessment CS level and CSP Native (Azure Policies)
- Orchestrator / Clouditor Orchestrator

⁵³ Alternatively, the SATRA questionnaire can be also manually answered by the user (i.e., in case the Catalogue – Questionnaire has not been used).

- Evidence Collection / Security Assessment Application Level (Codyze)
- Evidence Collection / Clouditor Discovery
- Evidence Collection Wazuh
- Evidence Collection

12.5.2 Workflow

The different interactions corresponding to this WF5 are shown in Table 43. Notice that in this workflow the “non-human” MEDINA role is in charge of performing most of the related actions, with exception of Step 2 which is executed by the CAB role. All other internal CSP roles have limited accesses as presented in section 3.3.5.

Table 43. WF5 description

Step	Description	Role	Comments
1	The Organizational Evidence Assessment UI: a. Automatically <i>assesses</i> the uploaded organizational documentation from the ToC based on the selected Metrics.	MEDINA	MEDINA supports EUCS auditors in their currently manual/time-consuming activity of assessing organizational evidence of the CSP (e.g., operation manuals). The automated assessment of such organizational evidence is expected to release auditors from most of this time-consuming activity, although a minimum level of human interaction is still expected (e.g., to confirm the assessment results of the tool, or to provide training data which is CSP-specific).
2	The Organizational Evidence Assessment UI: a. Automated assessments (based on organizational metrics) confirmed by the human-operator either with a Compliant or Not Compliant status. b. For each Metric, the confirmed assessment) is sent to Orchestrator by the human-operator.	CAB ⁵⁴	A human-in-the-loop (CAB) approach is followed to take final action on the NLP-based assessments.
3	Evidence Collection Codyze: a. Assesses code-level Resources from the ToC based on selected Metrics. b. Assessments results are automatically sent to the Orchestrator.	MEDINA	D3.3 [19] includes an analysis of the high assurance level requirements covered by the MEDINA tools.

⁵⁴ In this case we refer to the external Auditor

Step	Description	Role	Comments
4	Evidence Collection Clouditor: a. Assesses cloud service-level Resources from the ToC based on selected Metrics. b. Assessments results are automatically sent to the Orchestrator.	MEDINA	Please refer to D3.3 [19] for further details on metrics' coverage.
5	Evidence Collection Wazuh: a. Assesses cloud service-level Resources from the ToC based on selected Metrics. b. The status of the Wazuh vulnerability assessment is sent to the Orchestrator using VAT.	MEDINA	Please refer to D3.3 [19] for further details on metrics' coverage.
6	Evidence Collection CSP Native (Azure Policies): a. Assesses cloud service-level Resources from the ToC based on selected Metrics.	MEDINA	Please refer to D3.3 [19] for further details on metrics' coverage.
7	Orchestrator: a. Assessment Results from organizational and technical assessments are stored. b. Evidence from organizational and technical assessments is stored. c. Assessment Results are sent to Continuous Certification Evaluation (see WF7). d. Evidence and Assessment Results are hashed and sent to the MEDINA Evidence Trustworthiness Management System.	MEDINA	Organizational and technical evidence are managed by MEDINA in the same manner, so they can be postprocessed homogeneously by the rest of components (cf. WF6 WF7 and WF9).

12.6 WF6 - EUCS – Maintenance of ToC certificate

This WF6 departs from the current definition of certificate maintenance in the EUCS core document (see Figure 80) and, for the purposes of MEDINA, also adds an initial stage of “certificate issuance”. The main objective of WF6 is to take the “discrete/point in time” assessments from WF5 in order to trigger the different statuses of the corresponding EUCS certificate.

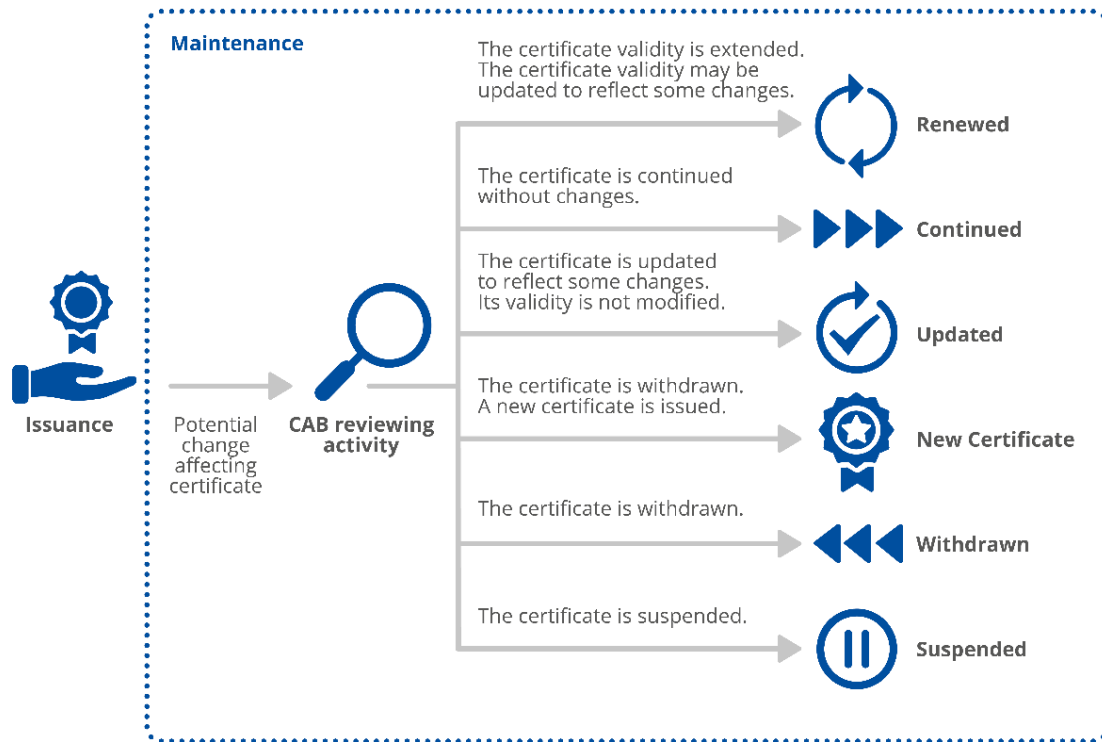


Figure 80. Certificate maintenance (source: EUCS [11])

12.6.1 Related Architectural Components

This workflow involves the components shown in building blocks 3 and 4 from Figure 16, namely:

- Continuous Certification Evaluation
- Risk Assessment and Optimization Framework
- Automated Certificate Lifecycle Management

12.6.2 Workflow

The different interactions corresponding to this WF6 are shown in Table 44.

Table 44. WF6 description

Step	Description	Role	Comments
1	<p>Continuous Certification Evaluation:</p> <p>a. Assessment Results (point-in-time assessment) are received from Orchestrator / Clouditor Orchestrator (push-mode).</p> <p>b. Tree-based evaluation is performed with received Assessment Results (which are received per-Resource).</p> <p>c. Tree-based evaluation results are stored in Certification Evaluation Storage.</p>	MEDINA	This component automatizes the currently manual audit process for analysing a set of evidence (in particular when operational efficiency is in scope, like in the case of EUCS High).

	d. If a non-compliance is found ⁵⁵ , then the Risk Assessment and Optimization Framework is invoked (RAOF, see Step 2 below)		
2	<p>Risk Assessment and Optimization Framework (RAOF):</p> <p>a. In analogy to WF4, the degree of non-compliance is computed based on the (point-in-time) assessments obtained from the Continuous Certification Evaluation.</p> <p>b. The degree of non-compliance is communicated to the Certificate Lifecycle Manager (see Step 4 below).</p>	MEDINA	As mentioned in WF4, the “degree on non-compliance” is computed comparing the real (e.g., based on monitored/declared status of requirements) risk level and ideal one (i.e., with all requirements satisfied). A threshold is to be set which determines if the difference is higher (major non-conformity) or lower (minor non-conformity). See D2.8 [12] for more details.
3	<p>Automated Certificate Lifecycle Manager:</p> <p>a. Based on the <i>Operational Effectiveness Criteria</i> defined by EUCS, the certificate maintenance lifecycle is triggered.</p> <p>b. The status of the certificate can be updated to any of New Certificate, Renewal, Continuation, Update, Withdraw, or Suspension.</p>	MEDINA	The core EUCS document defines the basis for MEDINA to implement the automation of the certificate lifecycle management.
4	<p>Automated Certificate Lifecycle Manager:</p> <p>a. Certificate status is published/updated on the MEDINA’s Public Registry.</p>	MEDINA	This is a required step in EUCS to provide transparency to the certification process.

12.7 WF7 - EUCS –Report on ToC Certificate

The goal of this WF7 is to report about the status of an EUCS certificate corresponding to the ToC and at different levels of detail, depending on the targeted audience (CAB, CSP, etc.). This WF7 considers for example, the case where a CAB needs to verify the technical/organizational evidence which resulted on the suspension of a certificate.

12.7.1 Related Architectural Components

This workflow involves the components shown in building block 4 from Figure 16, namely:

- Orchestrator
- Automated Certificate Lifecycle Management
- Continuous Certification Evaluation

⁵⁵ Compliances are not reported to the Risk Assessment and Optimization Framework

12.7.2 Workflow

The different interactions corresponding to this WF7 are shown in Table 45. Please notice that in this WF7, the CSP role refers to the following ones defined in section 3.3.7, namely:

- IT Security Governance
- Security Analyst
- Domain Governance
- Product and Service Owner
- Product (Security) Engineer
- Chief Information Security Office (CISO)
- Customer
- Auditor (only internal)

Table 45. WF7 description

Step	Description	Role	Comments
1	The Orchestrator UI is used to: a. Visualize the security assessment results (cf. "Assessment" tab).	CAB CSP NCCA ⁵⁶	The Orchestrator UI shows the assessment result for each selected Metric, adding additional information as timestamp, ResourceID, ResourceType and comment (if any).
2	The Automated Certificate Lifecycle Management UI: a. A lookup on the Public Registry(-ies) is performed to search for a specific criterion (e.g., Certificate_ID, ToC, CSP, period of time, etc.). b. If found on the Public Registry, the corresponding certificate is shown.	CAB CSP NCCA Customer	Details to display include certificate's history, ToC, degree of non-compliance, etc.
3	The Continuous Certificate Evaluation UI: a. The CloudServiceID to visualize in the tree is selected from the corresponding drop-down menu. b. The point-in-time assessment is selected from the corresponding drop-down. c. The tree is navigated to visualize the compliance status of different EUCS Requirements and associated Resources.	CAB CSP NCCA	The CCE UI implements a tree-like view to visualize the (aggregated) compliance status at different levels of abstraction.

12.8 WF8 – Auditor- Verifiable credentials for certificates (NEW)

The goal of this WF8 is to issue verifiable credentials to the CSPs and generate verifiable proofs to CSP customers. For this purpose, a Self-Sovereign Identity (SSI) Framework is considered. The SSI Framework provides CSPs with the capability to manage their own security certificates as part of their identity through verifiable credentials. "To manage their own identity" ultimately

⁵⁶ The NCCA can also obtain non-compliance information from these MEDINA components.

means that they store their identity on their own “user space” without intervention of a third-party.

12.8.1 Related Architectural Components

This workflow involves the components shown in building block 4 from Figure 16, namely:

- Self-Sovereign Identity (SSI)

12.8.2 Workflow

The different interactions corresponding to this WF8 are shown in Table 28. For this WF8, the CSP role refers exclusively to the internal auditor.

Table 46. WF9 description

Step	Description	Role	Comments
1	<p>Simulated issuer UI:</p> <ol style="list-style-type: none"> The simulated CAB SSI graphical interface (issuer) is at: https://medina-webapp.cybersec.digital.tecnalia.dev/. No authentication is needed. A connection screen will be automatically shown to the user. A connection with the issuer must be established. Once the issuer connection is established, the “MEDINA Wallet” logo (upper left side of the screen) must be pushed to start the operation. The user, as issuer, can issue new credentials through the “Create Credential” button on the Credentials section. For this purpose, the user needs to complete the required form and press the “Accept” button. <ul style="list-style-type: none"> Connection: to whom the credential will be issued. By default, “SSI Tecnalia Holder TEST” should be selected. Owned schema: the specific format considered for the credential. By default, “medina-ssi (version 1.0)”. Attributes (id, status): these will be automatically completed every time new information is received from the LCM (every time the LCM detects a change on the certificate status). Once issued, the new credential will be automatically shown in the credentials list on the CSP (see step 2). 	CAB	More details are provided on the SSI Framework User manual in <i>APPENDIX G: User Manuals</i> .
2	<p>The SSI Framework UI for automatic validation is available on the MEDINA IUI.</p> <p>Once authenticated, the user can check existing credentials and proofs:</p> <ol style="list-style-type: none"> The user will be able to see received credentials, and can copy or remove them. The user will be able to see emitted proofs and can copy or remove them. 	CSP	<p>The CSP can only execute “Read actions”.</p> <ul style="list-style-type: none"> • CSP cannot issue credentials, just list the existing ones. • CSP cannot emit proofs on demand; the proofs are automatically generated when a

			verifier makes a proof request.
3	<p>Simulated verifier UI:</p> <ol style="list-style-type: none"> The simulated customer SSI graphical interface (issuer) is at: https://medina-webapp.cybersec.digital.tecnalia.dev/. No authentication is needed. A connection screen will be automatically shown to the user. A connection with the verifier must be established. Once the verifier connection is established, the “MEDINA Wallet” logo (upper left side of the screen) must be pushed to start the operation. On the Proofs section, a list of previous proofs requests (and their status) is shown. The user, as verifier, can ask the CSP for proofs about the certificate status through the “Request Proofs” button on the Proofs section. For this purpose, the user needs to complete the required form and press the “Accept” button. <ul style="list-style-type: none"> Connection: to whom the proofs will be requested. By default, “SSI Tecnalia Holder TEST” should be selected. Comment: this is a comment for the proof (the reason could be provided). This is a text parameter. Attributes: this is a list of the attributes the verifier wants to know about the CSP. In MEDINA there are only two attributes (ID and status). Any of them (or both of them) should be included. If a different attribute is provided, the process works but the proof will be finally abandoned as the CSP cannot probe the required information. Conditions: this is an optional parameter related to the ZKP concept. This is not really applicable in MEDINA although it can be verified with the ID attribute. Once requested, the new request will be automatically shown in the proofs list on the CSP (see step 2) and will be automatically included in the proofs list on the verifier, obtaining the requested values and indicating the “done” state (shown in green). 	Cus- tomer ⁵⁷	More details are provided on the SSI Framework User manual in <i>APPENDIX G: User Manuals</i> .

12.9 WF9 - Auditor- Integrity verification (NEW)

The goal of this WF9 is to validate the integrity of both evidence and assessment results. For this purpose, the information currently available on the Orchestrator needs to be compared with the information recorded on the MEDINA Evidence Trustworthiness Management System. Thanks for this, integrity is verified.

⁵⁷ This entity represents a non-authenticated user of the framework

There are two modes of operation: automatic (though the MEDINA *Integrated UI*) and manual.

12.9.1 Related Architectural Components

This workflow involves the following components shown in building block 6 from Figure 16, namely:

- MEDINA Evidence Trustworthiness Management System (DLT)
- Orchestrator (only for the manual operation)

12.9.2 Workflow

The different interactions corresponding to this WF9 are shown in Table 47. For this WF9, the CSP role refers exclusively to the internal auditor.

Table 47. WF8 description

Step	Description	Role	Comments
1	The Trustworthiness system UI for automatic validation is available in the MEDINA IUI.	CAB CSP NCCA	Any authenticated user can access the system.
2	In the Trustworthiness system UI: a. The user selects if evidence or assessment results integrity are to be validated (on the header; evidence list is shown by default).	CAB CSP NCCA	Options available at the header are: <ul style="list-style-type: none"> • List of evidence • Evidence • List of assessment results • Assessment result
3	In the Trustworthiness system UI (List of Evidence): a. In the “List of Evidence” section (header), for evidence, different optional filters (cloud service ID and/or tool ID) can be applied to limit the evidence to be validated. b. Once filters are included (if needed), the “Submit” button needs to be pressed. c. The current integrity validation status is shown for each piece of evidence recorded on the Orchestrator (ID, integrity check).	CAB CSP NCCA	The validation is automatically executed for the evidence recorded on the Orchestrator.
4	In the Trustworthiness system UI (Evidence): a. For each evidence in the list obtained in step 3, the ID can be obtained. b. In the “Evidence” section (header), the evidence ID can be provided to better analyse the hash values of the evidence recorded in the Orchestrator and in the Trustworthiness system. By this way, an integrity problem can be identified.	CAB	This is especially useful for obtaining more details of the integrity validation for evidence with wrong results in the list from Step 3 as the information from Orchestrator and Blockchain is shown.
5	In the Trustworthiness system UI (List of Assessment Results): a. In the “List of Assessment Results” section (header), for assessment results, different optional filters (cloud service ID and/or	CAB CSP NCCA	The validation is automatically executed for the assessment results recorded in the Orchestrator.

	<p>metric ID and/or only compliant assessment results) can be applied to limit the assessment results to be validated.</p> <p>b. Once filters are included (if needed), the “Submit” button needs to be pressed.</p> <p>c. The current integrity validation status is shown for each assessment result recorded in the Orchestrator (ID, integrity check).</p>		
6	<p>In the Trustworthiness system UI (Assessment Result):</p> <p>a. For each assessment result in the list obtained in step 5, the ID can be obtained.</p> <p>b. In the “Assessment Result” section (header), the assessment result ID can be provided to better analyse the hash values of the assessment result recorded in the Orchestrator and in the Trustworthiness system. By this way, the integrity problem can be identified.</p>	<p>CAB</p> <p>CSP</p> <p>NCCA</p>	<p>This is especially useful for obtaining more details of the integrity validation for assessment results with wrong results in the list from Step 5 as the information from Orchestrator and Blockchain is shown.</p>
7	<p>The Trustworthiness system also provides a dashboard for manual integrity check. It is available at:</p> <p>https://kibana.medina.bclab.dev/.</p>	<p>CAB</p> <p>CSP</p> <p>NCCA</p>	<p>Authentication is needed (considering the same users as in MEDINA).</p>
8	<p>In the Trustworthiness system manual dashboard:</p> <p>a. The authenticated user will see the complete list of recorded evidence and assessment results associated to a specific cloud service provider.</p> <p>b. The user can apply different filters to look for specific evidence or assessment result.</p> <p>c. Once the specific evidence or assessment result is found, the user can obtain the recorded hash on the Blockchain.</p>	<p>CAB</p> <p>CSP</p> <p>NCCA</p>	<p>For evidence, the following information is shown:</p> <ul style="list-style-type: none"> • Evidence ID • Evidence Hash • Resource • Evidence Collector • CSP • Orchestrator timestamp (when evidence was received in the orchestrator). • Blockchain timestamp (when evidence was recorded on the Blockchain). <p>For assessment results, the following information is shown:</p> <ul style="list-style-type: none"> • Assessment Result ID • Assessment Result Hash • Metric • Associated evidence • Orchestrator timestamp (when assessment result was received in the orchestrator). • Blockchain timestamp (when assessment result was recorded on the Blockchain).
9	<p>In the Orchestrator UI:</p> <p>a. From the Orchestrator, the user should obtain the evidence or assessment result</p>	<p>CAB</p>	<p>This is the manual validation.</p>

	<p>value. For this purpose, in the Orchestrator, the “cloud service” section (header) is needed. Here, the “Show More Info” in the Assessment tag needs to be consulted.</p> <p>b. The user manually calculates the hash of the obtained information (evidence or assessment result) using SHA-256 algorithm. For example, the user can use: https://emn178.github.io/online-tools/sha256.html</p> <p>c. Both hashes (the one recorded on the Blockchain from Step 8 and the one just calculated) can be compared to identify if the obtained evidence or assessment result value has been tampered or modified.</p>	<p>CSP</p> <p>NCCA</p>	
--	--	------------------------	--

13 APPENDIX F: Published APIs

13.1 Component: Catalogue of Controls and Metrics

The following screenshot series show the list of available APIs that can be used by the components interacting with the *Catalogue of Controls and Metrics*.

cloud-service-provider-resource Cloud Service Provider Resource ^		
GET	/api/cloud-service-providers	getAllCloudServiceProviders
POST	/api/cloud-service-providers	createCloudServiceProvider
GET	/api/cloud-service-providers/count	countCloudServiceProviders
GET	/api/cloud-service-providers/{id}	getCloudServiceProvider
PUT	/api/cloud-service-providers/{id}	updateCloudServiceProvider
DELETE	/api/cloud-service-providers/{id}	deleteCloudServiceProvider
PATCH	/api/cloud-service-providers/{id}	partialUpdateCloudServiceProvider
cloud-service-resource Cloud Service Resource ^		
GET	/api/cloud-services	getAllCloudServices
POST	/api/cloud-services	createCloudService
GET	/api/cloud-services/count	countCloudServices
GET	/api/cloud-services/{id}	getCloudService
PUT	/api/cloud-services/{id}	updateCloudService
DELETE	/api/cloud-services/{id}	deleteCloudService
PATCH	/api/cloud-services/{id}	partialUpdateCloudService
question-answer-resource Question Answer Resource ^		
GET	/api/question-answers	getAllQuestionAnswers
GET	/api/question-answers/count	countQuestionAnswers
GET	/api/question-answers/{id}	getQuestionAnswer
question-assurance-level-resource Question Assurance Level Resource ^		
GET	/api/question-assurance-levels	getAllQuestionAssuranceLevels
GET	/api/question-assurance-levels/count	countQuestionAssuranceLevels
GET	/api/question-assurance-levels/{id}	getQuestionAssuranceLevel
question-resource Question Resource ^		
GET	/api/questions	getAllQuestions
GET	/api/questions/count	countQuestions
GET	/api/questions/count-extended	countQuestionsExtended

questionnaire-non-conformity-resource Questionnaire Non Conformity Resource ^

GET	/api/questionnaire-non-conformities	getAllQuestionnaireNonConformities	▼
GET	/api/questionnaire-non-conformities/count	countQuestionnaireNonConformities	▼
POST	/api/questionnaire-non-conformities/create	createQuestionnaireNonConformity	▼
POST	/api/questionnaire-non-conformities/save	saveQuestionnaireNonConformity	▼
GET	/api/questionnaire-non-conformities/{questionnaireName}	getQuestionnaireNonConformitiesByQuestionnaireName	▼

questionnaire-resource Questionnaire Resource ^

GET	/api/questionnaires	getAllQuestionnaires	▼
GET	/api/questionnaires/count	countQuestionnaires	▼
GET	/api/questionnaires/count-questions	countQuestions	▼
POST	/api/questionnaires/create	createQuestionnaire	▼
POST	/api/questionnaires/deleteByName	deleteQuestionnaireByName	▼
POST	/api/questionnaires/report-pdf	saveQuestionnaireReportAsPDF	▼
POST	/api/questionnaires/save	saveQuestionnaire	▼
GET	/api/questionnaires/{id}	getQuestionnaire	▼

reference-tom-resource Reference Tom Resource ^

GET	/api/reference-toms	getAllReferenceToms	▼
GET	/api/reference-toms/count	countReferenceToms	▼
GET	/api/reference-toms/{id}	getReferenceTom	▼
PUT	/api/reference-toms/{id}	updateReferenceTom	▼
PATCH	/api/reference-toms/{id}	partialUpdateReferenceTom	▼

resource-resource Resource Resource ^

GET	/api/resources	getAllResources	▼
POST	/api/resources	createResource	▼
GET	/api/resources/count	countResources	▼
GET	/api/resources/{id}	getResource	▼
PUT	/api/resources/{id}	updateResource	▼
DELETE	/api/resources/{id}	deleteResource	▼
PATCH	/api/resources/{id}	partialUpdateResource	▼

resource-type-resource Resource Type Resource			^
GET	/api/resource-types	getAllResourceTypes	▼
POST	/api/resource-types	createResourceType	▼
GET	/api/resource-types/count	countResourceTypes	▼
GET	/api/resource-types/{id}	getResourceType	▼
PUT	/api/resource-types/{id}	updateResourceType	▼
DELETE	/api/resource-types/{id}	deleteResourceType	▼
PATCH	/api/resource-types/{id}	partialUpdateResourceType	▼
security-control-category-resource Security Control Category Resource			^
GET	/api/security-control-categories	getAllSecurityControlCategories	▼
GET	/api/security-control-categories/count	countSecurityControlCategories	▼
GET	/api/security-control-categories/{id}	getSecurityControlCategory	▼
PUT	/api/security-control-categories/{id}	updateSecurityControlCategory	▼
PATCH	/api/security-control-categories/{id}	partialUpdateSecurityControlCategory	▼
security-control-framework-resource Security Control Framework Resource			^
GET	/api/security-control-frameworks	getAllSecurityControlFrameworks	▼
GET	/api/security-control-frameworks-full	getAllSecurityControlFullFrameworks	▼
GET	/api/security-control-frameworks/checkHasRequirements/{name}	checkHasRequirements	▼
GET	/api/security-control-frameworks/count	countSecurityControlFrameworks	▼
GET	/api/security-control-frameworks/{id}	getSecurityControlFramework	▼
PUT	/api/security-control-frameworks/{id}	updateSecurityControlFramework	▼
PATCH	/api/security-control-frameworks/{id}	partialUpdateSecurityControlFramework	▼
security-control-resource Security Control Resource			^
GET	/api/security-controls	getAllSecurityControls	▼
GET	/api/security-controls/count	countSecurityControls	▼
GET	/api/security-controls/{id}	getSecurityControl	▼
PUT	/api/security-controls/{id}	updateSecurityControl	▼
PATCH	/api/security-controls/{id}	partialUpdateSecurityControl	▼

security-metric-resource Security Metric Resource ^

GET	/api/security-metrics	getAllSecurityMetrics	✓
POST	/api/security-metrics	createSecurityMetric	✓
GET	/api/security-metrics/count	countSecurityMetrics	✓
GET	/api/security-metrics/{id}	getSecurityMetric	✓
PUT	/api/security-metrics/{id}	updateSecurityMetric	✓
DELETE	/api/security-metrics/{id}	deleteSecurityMetric	✓
PATCH	/api/security-metrics/{id}	partialUpdateSecurityMetric	✓

similar-control-resource Similar Control Resource ^

GET	/api/similar-controls	getAllSimilarControls	✓
GET	/api/similar-controls/count	countSimilarControls	✓
GET	/api/similar-controls/{id}	getSimilarControl	✓
PUT	/api/similar-controls/{id}	updateSimilarControl	✓
PATCH	/api/similar-controls/{id}	partialUpdateSimilarControl	✓

target-value-resource Target Value Resource ^

GET	/api/target-values	getAllTargetValues	✓
POST	/api/target-values	createTargetValue	✓
GET	/api/target-values/count	countTargetValues	✓
GET	/api/target-values/{id}	getTargetValue	✓
PUT	/api/target-values/{id}	updateTargetValue	✓
DELETE	/api/target-values/{id}	deleteTargetValue	✓
PATCH	/api/target-values/{id}	partialUpdateTargetValue	✓

tom-resource Tom Resource ^

GET	/api/toms	getAllToms	✓
GET	/api/toms/count	countToms	✓
GET	/api/toms/framework-assurance/{frameworkName}	getTomsByFrameworkName	✓
GET	/api/toms/framework-assurance/{frameworkName}/{assuranceLevel}	getTomsByFrameworkNameAndAssuranceLevel	✓
GET	/api/toms/{id}	getTom	✓
PUT	/api/toms/{id}	updateTom	✓
PATCH	/api/toms/{id}	partialUpdateTom	✓

user-resource User Resource ^

GET	/api/admin/users	getAllUsers	✓
GET	/api/admin/users/{login}	getUser	✓

13.2 Component: NL2CNL Translator and DSL Mapper

The following screenshots show available APIs that can be used by the other components to interact with the *NL2CNL Translator* and the *DSL Mapper*, respectively.

GET	/livez	Liveness Check
GET	/readyz	Readiness Check
POST	/create_reo_for_requirement/{username}	Get Reo For Tom

GET	/livez	Liveness Check
GET	/readyz	Readiness Check
POST	/map_obligations_to_rego/{reoid}	Map Obl2Rego

13.3 Component: CNL Editor

The following screenshot shows the list of available APIs that can be used by the components interacting with the *CNL Editor*.

reo-operations-controller REO Operations Controller		
POST	/reo/create/{username}	Creates new REO
GET	/reo/delete/{reoid}	Delete REO
GET	/reo/filterby/cloudservice	Fetch the details of the REO filter by cloud service ID
GET	/reo/get/{reoid}	Retrieve the REO file
GET	/reo/map/{reoid}	Send REO to Mapper
POST	/reo/update/{reoid}	Update the REO file

13.4 Component: Risk Assessment and Optimisation Framework

The following screenshots shows the list of available APIs that can be used by the components interacting with *SATRA/RAOF*. The API can be used for two purposes:

- 1) Operate the risk and non-conformity assessment process through a custom-built dashboard.
- 2) Use of the dynamic risk assessment functionality during the continuous certification monitoring phase by other MEDINA components.

SATRA - Self-Assessment Tool for Risk Analysis ^{0.1}

[Base URL: /api/v1]
/api/v1/swagger.json

Manage interaction with the SATRA engine

registration Register a new practice for that user

POST /registration/ToE/ Create a new practice

GET /registration/access_resp/{username}/{password} Get the access token from keycloak

DELETE /registration/delete_contract/{UUID} Delete a ToE

PUT /registration/update_contract/{UUID} Update ToE

practice Interact with the survey, update question/answers and get risk...

POST /practice/analysis/{UUID} Send information on a test result

POST /practice/answer/{UUID}/{question_id}/{answer_id} Send (eventually, update) an answer for a specific question

GET /practice/answer/{UUID}/{type_id} Get all the possible answers by type_id

GET /practice/answers/{UUID} Get the question and the answers chosen by the user for the ToE

POST /practice/asset_answers/{UUID} Send (eventually, update) an asset answers

GET /practice/assets/{UUID} Get all assets type

GET /practice/assets_answers/{UUID} Get all assets answer

DELETE /practice/assets_answers/{UUID}/{asset_id} Delete a specific asset throughout the name

GET /practice/assets_dynamic_answers/{UUID} Get all dynamic assets answer

GET /practice/assurance/{UUID} Get all possible assurance levels

GET /practice/certification/{UUID} Get all possible certification schemes

GET /practice/csp_market/{UUID} Get all possible CSP's markets

POST	/practice/dynamic_evaluated_risk/{UUID}	Dynamic evaluated risk computation
POST	/practice/map/{UUID}	Map an external questionnaire
GET	/practice/non_conformity_gap/{UUID}	Get all possible non conformity gap
GET	/practice/question/{UUID}	Get all the questions
GET	/practice/question/{UUID}/{question_id}	Get one question and its possible answers
GET	/practice/risk/{UUID}	Get the updated risk
GET	/practice/threats/{UUID}	Get the updated threat

13.5 Component: Continuous Certification Evaluation

The following screenshots shows the list of available APIs that can be used by the components interacting with *CCE*.

cce-api-controller Continuous Certification Evaluation REST API			^
GET	/toms	Returns a list of all TOMs	▼
GET	/toes/{targetOfEvaluationId}	Returns the current tree state for the chosen ToE.	▼
GET	/toes/{targetOfEvaluationId}/statistics	Returns the statistics (operational effectiveness values) for the specified ToE and the time period between start and end times. End time parameter is optional, if not specified it defaults to the current time.	▼
GET	/toes/{targetOfEvaluationId}/listHistory	Returns a list (tree state ID and timestamp) of all saved tree states for the specified ToE.	▼
GET	/toeList	Returns a list of available Targets of Evaluation (ToE) with their ID, name, and Cloud Service ID.	▼
GET	/history/{treeStateId}	Returns the specified tree state by ID.	▼
GET	/complianceReport	Returns compliance report for all user's targets of evaluation	▼

gRPC functions

- `cce.Evaluation.AddAssessmentResult(AssessmentResult)` returns `(google.protobuf.Empty)`
- `cce.Statistics.GetTreeStatistics(StatisticsQuery)` returns `(TreeStatistics)`
- `cce.Notification.TargetOfEvaluationCreated(TargetOfEvaluation)` returns `(google.protobuf.Empty)`

See `src/main/proto/` for message entities definitions.

The complete technical specification (request and response parameters and types) of the gRPC API is available in the CCE repository: <https://git.code.tecnalia.com/medina/public/continuous-certification-evaluation/-/tree/main/src/main/proto>

13.6 Component: Life Cycle Manager

The following screenshot shows the list of available APIs that can be used by the components interacting with *LCM*.

POST	/certificate	Create a new certificate	✓ ↕
PUT	/certificate	Update a certificate	✓ ↕
DELETE	/certificate	Delete a certificate	✓ ↕
POST	/evaluation	Provide a risk evaluation	✓ ↕
GET	/statechange/{certificate_id}	Get information about the state history of a certificate	✓ ↕

13.7 Component: Automated Self-Sovereign Identity-based certificates management (SSI)

The following screenshot shows the list of available APIs that can be used by the components interacting with *SSI Framework*.

DELETE	/certificate/id
GET	/certificate/id
PUT	/certificate/id
GET	/certificates
POST	/certificates

13.8 Component: Assessment and Management of Organizational Evidence – AMOE

The following screenshot shows the list of available APIs that can be used by the components interacting with *AMOE*.

GET	/api/v1/files/{cloud_service_id}	AMOE List Files Cloud Service	▼
POST	/api/v1/files/	AMOE List Files Cloud Services	▼
GET	/api/v1/file/{file_id}	AMOE Get File	▼
GET	/api/v1/file/last/{cloud_service_id}	get_amoe_last_file	▼
GET	/api/v1/evidence/list/{file_id}	AMOE Get List Evidence For File	▼
POST	/api/v1/evidence/list_per_metric_id	AMOE Get List Evidence Per Metric	▼
GET	/api/v1/evidence/{evidence_id}	AMOE Get Evidence	▼
POST	/api/v1/evidence/assessment	AMOE Set Assessment Result	▼
GET	/api/v1/evidence/send_to_orchestrator/{evidence_id}	AMOE Send Assessment Result	▼
GET	/api/v1/evidence/file/{evidence_id}	AMOE Get HTML File	▼
GET	/api/v1/file/pdf/{file_id}	AMOE Get PDF File	▼
POST	/api/v1/file/{cloud_service}	AMOE Upload PDF File	▼
GET	/api/v1/file/delete/{file_id}	AMOE Delete File And Evidence	▼

13.9 Component: Orchestrator

The following screenshots show the list of available APIs that can be used by the components interacting with the *Orchestrator*.

Orchestrator ^

GET	/v1/orchestrator/assessment_results
POST	/v1/orchestrator/assessment_results
GET	/v1/orchestrator/assessment_tools
POST	/v1/orchestrator/assessment_tools
GET	/v1/orchestrator/assessment_tools/{toolId}
PUT	/v1/orchestrator/assessment_tools/{toolId}
DELETE	/v1/orchestrator/assessment_tools/{toolId}
GET	/v1/orchestrator/catalogs
POST	/v1/orchestrator/catalogs
GET	/v1/orchestrator/catalogs/{catalogId}
PUT	/v1/orchestrator/catalogs/{catalogId}
DELETE	/v1/orchestrator/catalogs/{catalogId}
GET	/v1/orchestrator/catalogs/{catalogId}/categories/{categoryName}/controls/{controlId}
GET	/v1/orchestrator/catalogs/{catalogId}/category/{categoryName}
GET	/v1/orchestrator/certificates
POST	/v1/orchestrator/certificates
GET	/v1/orchestrator/certificates/{certificateId}
PUT	/v1/orchestrator/certificates/{certificateId}
DELETE	/v1/orchestrator/certificates/{certificateId}
GET	/v1/orchestrator/cloud_services
POST	/v1/orchestrator/cloud_services

```

GET      /v1/orchestrator/cloud_services/{cloudServiceId}
PUT      /v1/orchestrator/cloud_services/{cloudServiceId}
DELETE   /v1/orchestrator/cloud_services/{cloudServiceId}
GET      /v1/orchestrator/cloud_services/{cloudServiceId}/catalogs/{catalogId}/toes
PUT      /v1/orchestrator/cloud_services/{cloudServiceId}/catalogs/{catalogId}/toes
DELETE   /v1/orchestrator/cloud_services/{cloudServiceId}/catalogs/{catalogId}/toes
GET      /v1/orchestrator/cloud_services/{cloudServiceId}/metric_configurations
GET      /v1/orchestrator/cloud_services/{cloudServiceId}/metric_configurations/{metricId}
PUT      /v1/orchestrator/cloud_services/{cloudServiceId}/metric_configurations/{metricId}
GET      /v1/orchestrator/controls
GET      /v1/orchestrator/metrics
POST     /v1/orchestrator/metrics
GET      /v1/orchestrator/metrics/{metricId}
PUT      /v1/orchestrator/metrics/{metricId}
GET      /v1/orchestrator/metrics/{metricId}/implementation
PUT      /v1/orchestrator/metrics/{metricId}/implementation
GET      /v1/orchestrator/toes
POST     /v1/orchestrator/toes

```

13.10 Component: Trustworthiness System

The following screenshots show the list of available APIs that can be used by the components interacting with the *MEDINA Evidence Trustworthiness System*.

POST	/client/account
GET	/client/account
POST	/client/wallet
GET	/client/wallet
POST	/client/registration
GET	/client/admin
POST	/client/admin
DELETE	/client/admin

GET	/client/adminnum
GET	/client/orchestratorsnum
GET	/client/orchestrator/evidence/check
GET	/client/orchestrator/assessment/checkhash
GET	/client/orchestrator/assessment/checkcompliance
GET	/client/orchestrators
GET	/client/authorizedowner
POST	/client/authorizedowner
DELETE	/client/authorizedowner
GET	/client/authorizedownernum
POST	/client/orchestrator
POST	/client/orchestrator/evidence
POST	/client/orchestrator/assessment
GET	/client/orchestrator/evidence/{id}
GET	/client/orchestrator/assessment/{id}
GET	/client/orchestrator/evidences
GET	/client/orchestrator/assessments
GET	/client/orchestrator/owner
GET	/client/orchestrator/creationtime
GET	/client/orchestrator/id

13.11 Component: Evidence Collection (Cloud Discovery)

The following screenshot shows the list of available APIs that can be used by the components interacting with the *Evidence Collection*.

Discovery ^

POST /v1/discovery/query

POST /v1/discovery/start

13.12 Component: Security Assessment (Clouditor)

The following screenshot shows the list of available APIs that can be used by the components interacting with the *Security Assessment*.

Assessment ^

POST /v1/assessment/evidences

GET /v1/assessment/results

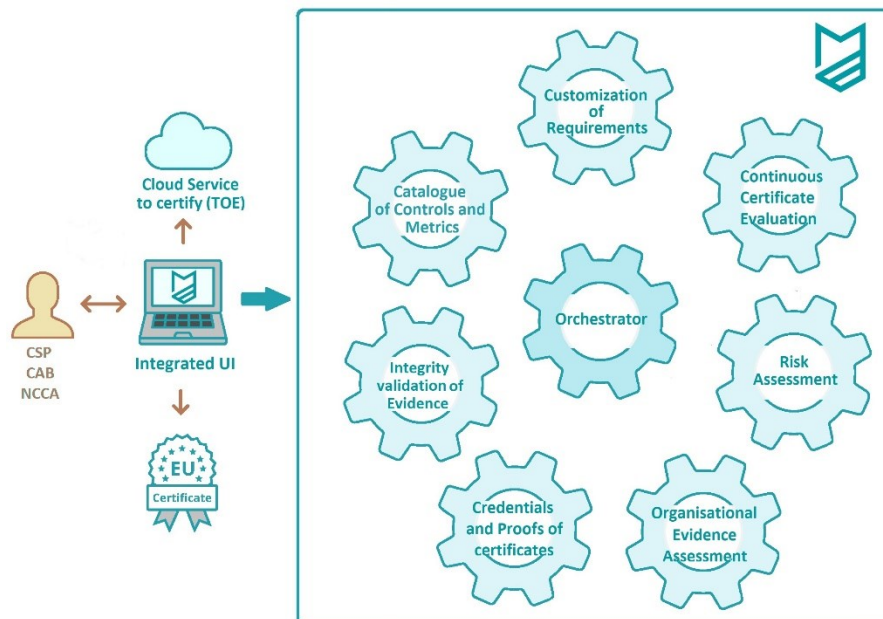
14 APPENDIX G: User Manuals

This Appendix includes the user manuals for those components of the MEDINA framework described in section 4 that have a graphical user interface, namely:

- *Catalogue of Controls and Metrics* (see section 4.1.1)
- *Orchestrator* (see section 4.6.1)
- *CNL Editor* (Customization of Requirements) (see section 4.2.2)
- *Risk Assessment and Optimization Framework* (see section 4.3.1)
- *Organizational Evidence Gathering and Processing* (see section 4.5.1)
- *Continuous Certification Evaluation* (see section 4.4.1)
- *Self-Sovereign Identity (SSI) Framework* (see section 4.4.3)
- *Evidence Trustworthiness Management System* (see section 4.6.2)

Catalogue of Controls and Metrics

- User Manual -



Project Title:	MEDINA - Security Framework to achieve a continuous audit-based certification in compliance with the EU-wide cloud security certification scheme
Project Number:	952633
Editor:	Iñaki Etxaniz (Fundación TECNALIA Research and Innovation)
Version:	v1.0
Date:	31.07.2023
Distribution level:	PU



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952633

Table of contents

1. Introduction	4
1.1. User Roles and Permissions	4
2. User Manual	5
2.1. Toolbar	5
2.2. Catalogue	5
2.2.1. Frameworks	5
2.2.2. Categories	7
2.2.3. Controls	8
2.2.4. Requirements	10
2.2.5. Metrics	12
2.2.6. Similar Controls	14
2.2.7. Implementation Guidelines	15
2.3. Questionnaires	17
2.3.1. Create a new Questionnaire	17
2.3.2. Manage Questionnaires	19
2.3.3. Generate a Report	20
2.3.4. Connection to SATRA	21
2.4. Administration	22
2.4.1. Gateway	23
2.4.2. API	23
2.4.3. Audit logs	25
3. Delivery and Usage	27
3.1. Licensing information	27
3.2. Download	27
3.3. More information	27

List of Figures

FIGURE 1. CATALOGUE TOOLBAR.....	5
FIGURE 2. LIST OF SECURITY FRAMEWORKS.....	5
FIGURE 3. EDIT THE DETAILS OF A FRAMEWORK	6
FIGURE 4. VIEW THE DETAILS OF A FRAMEWORK	6
FIGURE 5. SECURITY CATEGORIES BELONGING TO THE EUCS FRAMEWORK.....	7
FIGURE 6. LIST OF CATEGORIES	7
FIGURE 7. EDIT THE DETAILS OF A CATEGORY.....	8
FIGURE 8. CONTROLS BELONGING TO THE CATEGORY “ORGANISATION OF INFORMATION SECURITY”	8
FIGURE 9. LIST OF CONTROLS.....	9
FIGURE 10. FILTERING OF CONTROLS	9
FIGURE 11. EDIT THE DETAILS OF A CONTROL	9
FIGURE 12. REQUIREMENTS BELONGING TO THE OIS-01 CONTROL	10
FIGURE 13. LIST OF REQUIREMENTS	10
FIGURE 14. FILTERING OF REQUIREMENTS	11
FIGURE 15. EDIT THE DETAILS OF A REQUIREMENT	11
FIGURE 16. METRICS IMPLEMENTED FOR THE OPS-05.3H REQUIREMENT	12
FIGURE 17. LIST OF METRICS	12
FIGURE 18. FILTERING OF METRICS	13
FIGURE 19. VIEW THE DETAILS OF A METRIC	13
FIGURE 20. LIST OF SIMILAR CONTROLS	14
FIGURE 21. FILTERING OF SIMILAR CONTROLS	14
FIGURE 22. EDIT THE DETAILS OF A SIMILAR CONTROL	15
FIGURE 23. LIST OF IMPLEMENTATION GUIDELINES.....	15
FIGURE 24. DETAILS OF AN IMPLEMENTATION GUIDELINE.....	16
FIGURE 25. START A NEW QUESTIONNAIRE	17
FIGURE 26. NUMBER OF QUESTIONS FOR EACH LEVEL OF ASSURANCE	17
FIGURE 27. QUESTIONNAIRE STRUCTURE	18
FIGURE 28. DETAILS OF A QUESTION	18
FIGURE 29. COMPLIANCE VALUE FOR A REQUIREMENT.....	19
FIGURE 30. LOAD AN EXISTING QUESTIONNAIRE.....	19
FIGURE 31. EDIT/READ AN EXISTING QUESTIONNAIRE	20
FIGURE 32. EDIT A QUESTIONNAIRE (AUDITOR ROLE).....	20
FIGURE 33. REPORT GENERATED FOR A QUESTIONNAIRE	21
FIGURE 34. CALCULATED COMPLIANCE.....	21
FIGURE 35. ACTIONS THAT CAUSE THE SAVING OF A QUESTIONNAIRE.....	22
FIGURE 36. ADMINISTRATION MENU – GATEWAY	23
FIGURE 37. ADMINISTRATION MENU – API.....	23
FIGURE 38. LIST OF AUDIT LOGS.....	25

1. Introduction

The *Catalogue of Controls and Metrics* (a.k.a. *Catalogue*) provides the necessary technological means for the endorsement of any security scheme and their related attributes. Furthermore, it provides guidance for the implementation, as well as the (self-)assessment of security requirements.

The *Catalogue* allows the compliance manager of a CSP (Cloud Service Provider) or an auditor to select the EUCS (European Cybersecurity Certification Scheme for Cloud Services)¹ and obtain all the information and guidance related to that security scheme, namely the controls, security requirements, assurance levels, etc. In other words, everything that can be considered as "static" information that appears in the certification standard. This information has been enriched with the following facilities:

- Filtering of information based on some values for the attributes, such as the selection of requirements of a certain assurance level, the selection of requirements from a certain framework or the selection of metrics related to a requirement.
- Homogenization of different certification schemes, in the sense of showing the requirements that are equivalent in different security frameworks with reference to the EUCS.
- Consultation of implementation guidelines. An implementation guideline is an explanation of how a specific security requirement can be implemented, in a vendor and technology-agnostic way. Examples from larger CSPs are provided for inspiration.
- Finally, the Catalogue also contains a first implementation of a Questionnaire that allows a CSP to perform a self-assessment of the fulfilment degree of the EUCS scheme. It covers all the requirements of EUCS for all levels of certifications (Basic, Substantial and High).

1.1. User Roles and Permissions

Access to the *Catalogue* is managed by Keycloak². The visibility of the different components of the *Catalogue*, and the operations that are allowed to be carried out, are conditioned by the role to which each user is assigned.

The table below details which actions are allowed for each of the defined roles in MEDINA:

Roles	Allowed Actions
IT Security Governance	Read entities, Load questionnaire, Generate report
Security Analyst	Read entities, Load questionnaire, Generate report
Domain Governance	Read entities, Load questionnaire, Generate report
Product and Service Owner	Read/Write entities, Start/Edit questionnaire, Save questionnaire, Generate report, Remove questionnaire
Product (Security) Engineer	Load questionnaire, Generate report
Chief Information Security Office (CISO)	Load questionnaire, Generate report
Customer (non-authenticated user)	None
Auditor	Load questionnaire, Edit Non-conformities, Save questionnaire, Generate report

¹ EUCS – Cloud Service Scheme <https://www.enisa.europa.eu/publications/eucs-cloud-service-scheme>

² <https://www.keycloak.org>

2. User Manual

2.1. Toolbar

The Catalogue of Controls and Metrics includes a toolbar (see Figure 1), always accessible in the upper area, with all the options that are available in the tool:

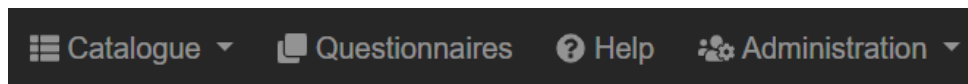


Figure 1. Catalogue toolbar

The different menu options, which are described in the following sections, are as follows:

- **Catalogue:** Provides access to information about Frameworks, Categories, Controls, Requirements, Metrics, Similar Controls, and Implementation Guidelines.
- **Questionnaires:** Provides access to the self-assessment questionnaires.
- **Help:** Provides access to the user manual.
- **Administration:** Provides access to the gateway and REST API information. This option is only available to users with administration rights.

2.2. Catalogue

The *Catalogue* menu option displays a submenu with the following options that are detailed below:

- Frameworks
- Categories
- Controls
- Requirements
- Metrics
- Similar Controls
- Implementation Guidelines

2.2.1. Frameworks

The main *Frameworks* window (see Figure 2) shows the list of all the registered frameworks in the *Catalogue*. The current version only includes the EUCS Security Framework.

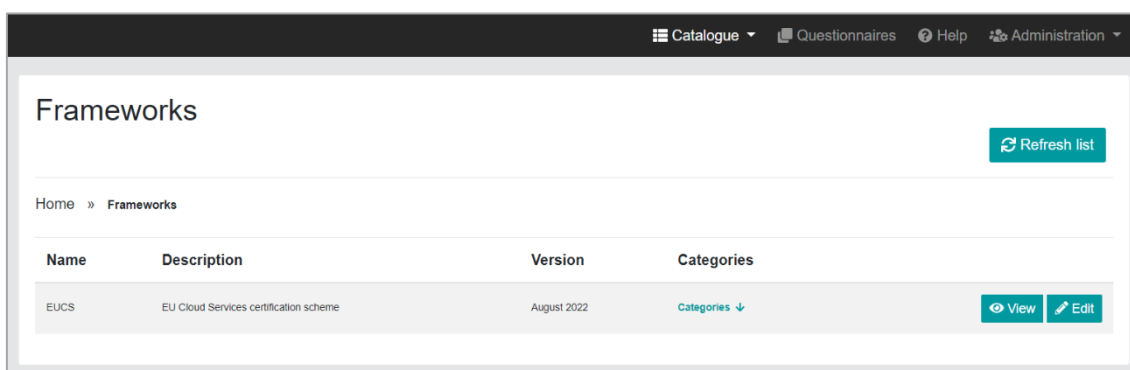


Figure 2. List of Security Frameworks

The following fields are listed for each Framework:

- Name

- Description
- Version

At the right of each Framework, two buttons allow to *Edit/View* the details of the entity, in this case the Framework. In the *Edit* window (see Figure 3), only the description and the version fields can be updated³. While clicking on the *View* button, a similar window (see Figure 4), but in this case with view-only fields, is shown⁴.

Figure 3. Edit the details of a Framework

Figure 4. View the details of a Framework

Finally, each Framework offers the possibility to access its related Categories (see Figure 8) by clicking on the *Categories* link:

Name	Description	Version	Categories	
EUCS	EU Cloud Services certification scheme	August 2022	Categories ↓	View Edit

³ The Edit options are further limited by the role-based access feature, so that some roles can use this option and others cannot.

⁴ As these *View/Edit* options are repeated in almost all entities, and as the structure of the two windows is quite similar, in the remainder of this manual we will only show one of the two windows.

Categories (Framework: EUCS)					Refresh list	
Home » Framework: EUCS » Categories						
Code	Name	Description	Framework	Controls		
A1	Organisation of Information Security	Plan, implement, maintain and continuously improve the information security framework within the organisation	EUCS ↑	Controls ↓	View	Edit
A2	Information Security Policies	Provide a global information security policy derived into policies and procedures regarding security requirements and to support business requirements	EUCS ↑	Controls ↓	View	Edit
A3	Risk Management	Provide a global information security policy, derived into policies and procedures regarding security requirements and to support business requirements	EUCS ↑	Controls ↓	View	Edit

Figure 5. Security Categories belonging to the EUCS Framework

2.2.2. Categories

The main *Categories* window (see Figure 6) lists all the Categories stored in the *Catalogue*.

Catalogue

Questionnaires

Help

Administration

Categories

Refresh list

Home » Frameworks » Categories

Code	Name	Description	Framework	Controls	
A1	Organisation of Information Security	Plan, implement, maintain and continuously improve the information security framework within the organisation	EUCS ↑	Controls ↓	<div><div>View</div><div>Edit</div></div>
A2	Information Security Policies	Provide a global information security policy derived into policies and procedures regarding security requirements and to support business requirements	EUCS ↑	Controls ↓	<div><div>View</div><div>Edit</div></div>
A3	Risk Management	Provide a global information security policy, derived into policies and procedures regarding security requirements and to support business requirements	EUCS ↑	Controls ↓	<div><div>View</div><div>Edit</div></div>
A4	Human Resources	Ensure that employees understand their responsibilities, are aware of their responsibilities with regard to information security, and that the organisation's assets are protected in the event of changes in responsibilities or termination.	EUCS ↑	Controls ↓	<div><div>View</div><div>Edit</div></div>
A5	Asset Management	Identify the organisation's own assets and ensure an appropriate level of protection throughout their lifecycle	EUCS ↑	Controls ↓	<div><div>View</div><div>Edit</div></div>

Figure 6. List of Categories

The following fields are listed for each Security Category:

- Code
- Name
- Description

As with any other entity in the *Catalogue*, each Security Category allows to view its details or to edit it. Let us look at the *Edit* window (see Figure 7), where only the description can be updated:

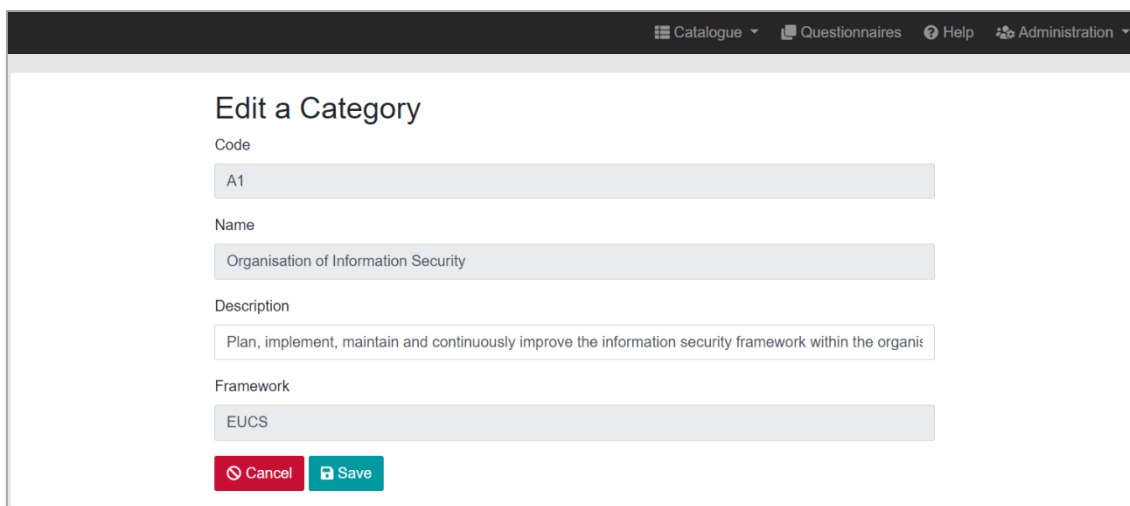


Figure 7. Edit the details of a Category

Finally, for each Category in Figure 6, the user can access its related Controls (see Figure 8) or can go back to the Framework window, by clicking on the *Controls* and *EUCS* links, respectively:

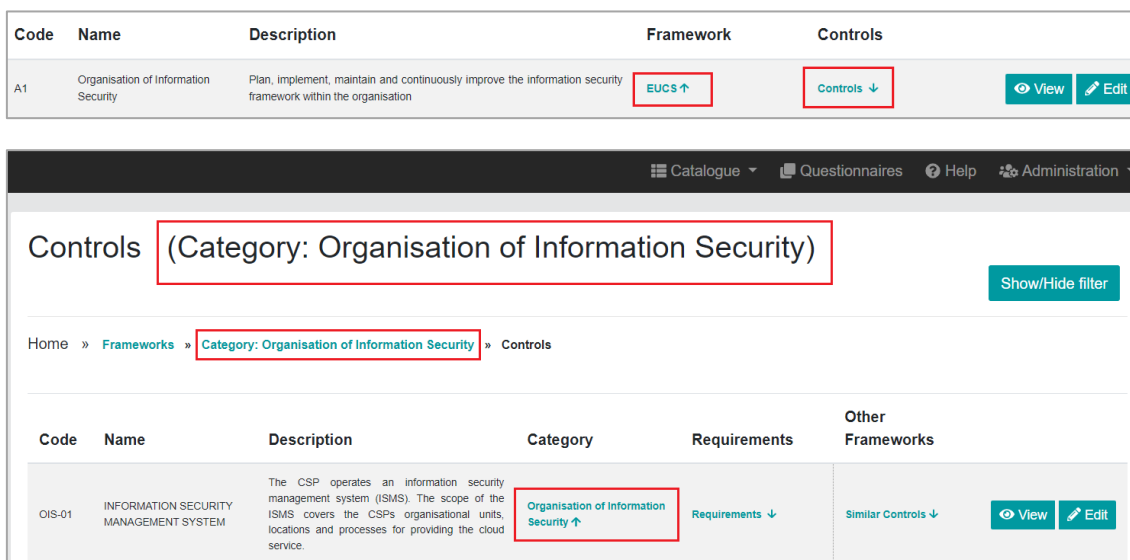


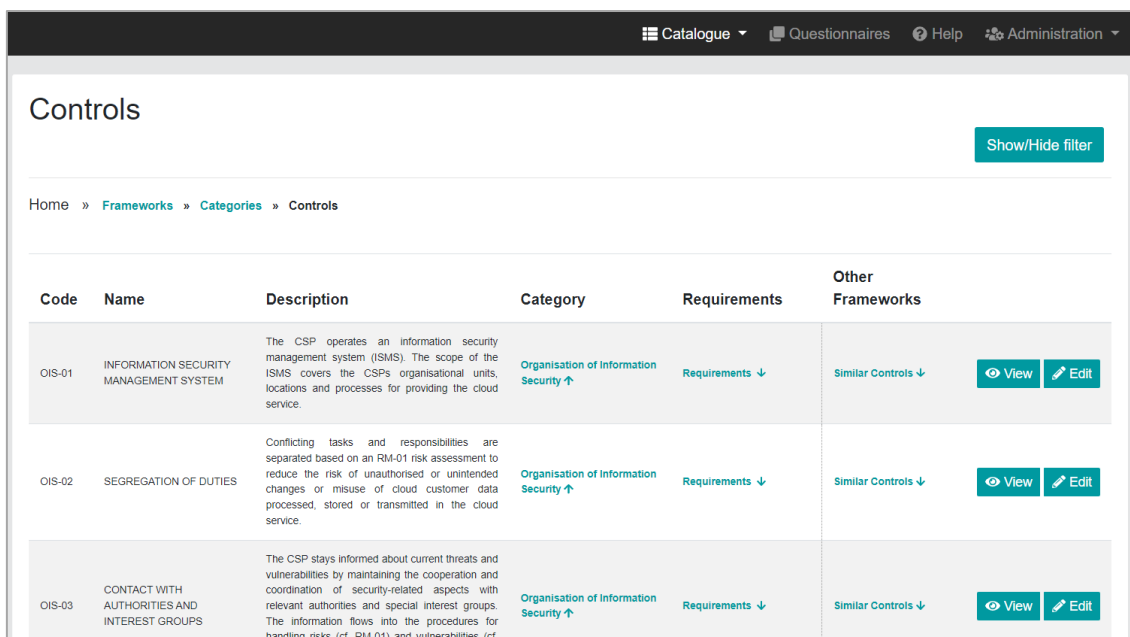
Figure 8. Controls belonging to the Category “Organisation of Information Security”

2.2.3. Controls

The main *Controls* window (see Figure 9) shows all the Security Controls registered in the *Catalogue*.

The following fields are listed for each Security Control:

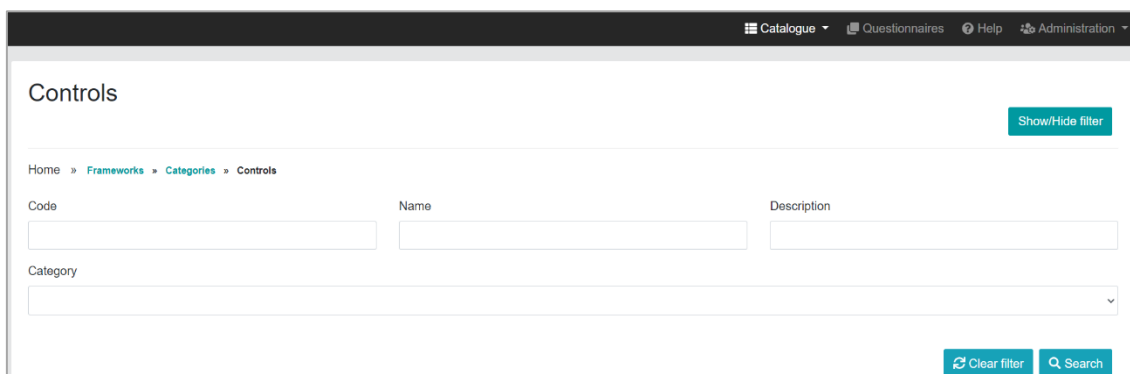
- Code
- Name
- Description



Code	Name	Description	Category	Requirements	Other Frameworks
OIS-01	INFORMATION SECURITY MANAGEMENT SYSTEM	The CSP operates an information security management system (ISMS). The scope of the ISMS covers the CSPs organisational units, locations and processes for providing the cloud service.	Organisation of Information Security ↑	Requirements ↓	Similar Controls ↓ View Edit
OIS-02	SEGREGATION OF DUTIES	Conflicting tasks and responsibilities are separated based on an RM-01 risk assessment to reduce the risk of unauthorised or unintended changes or misuse of cloud customer data processed, stored or transmitted in the cloud service.	Organisation of Information Security ↑	Requirements ↓	Similar Controls ↓ View Edit
OIS-03	CONTACT WITH AUTHORITIES AND INTEREST GROUPS	The CSP stays informed about current threats and vulnerabilities by maintaining the cooperation and coordination of security-related aspects with relevant authorities and special interest groups. The information flows into the procedures for handling risks (cf. RM-01) and vulnerabilities (cf. RM-01).	Organisation of Information Security ↑	Requirements ↓	Similar Controls ↓ View Edit

Figure 9. List of Controls

This list of Security Controls can be customized using the implemented filters (code, name, description, and category), as shown in Figure 10). Also, each Control can be edited (see Figure 11), and its details can be consulted by clicking on the *View* button.



Controls

Home » Frameworks » Categories » Controls

Code

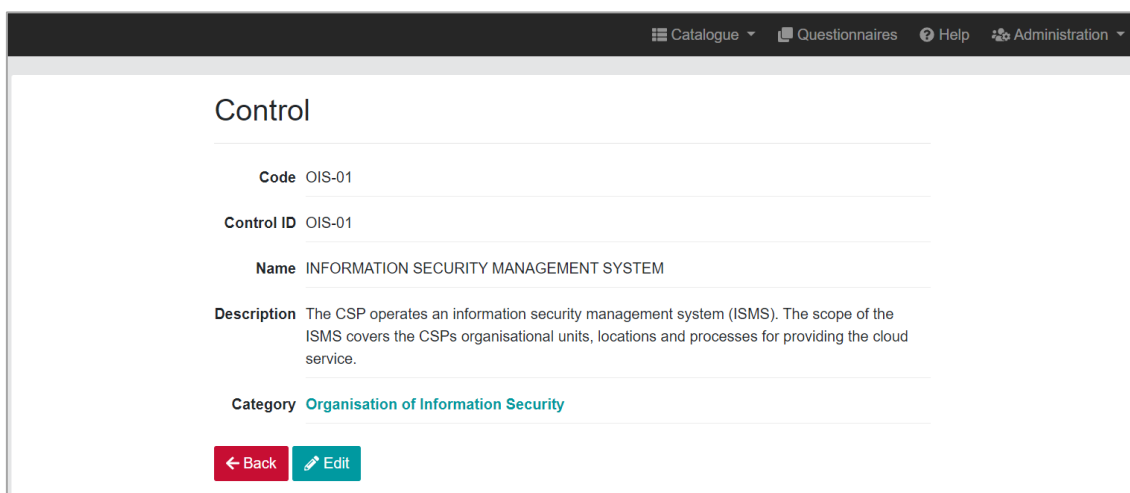
Name

Description

Category

[Clear filter](#) [Search](#)

Figure 10. Filtering of Controls



Control

Code OIS-01

Control ID OIS-01

Name INFORMATION SECURITY MANAGEMENT SYSTEM

Description The CSP operates an information security management system (ISMS). The scope of the ISMS covers the CSPs organisational units, locations and processes for providing the cloud service.

Category Organisation of Information Security

[Back](#) [Edit](#)

Figure 11. Edit the details of a Control

Finally, for each Control in Figure 9, the user can go back to its Category, can access its related Requirements (see Figure 12), or can consult Similar Controls in other frameworks, by clicking on the following links:

Code	Name	Description	Category	Requirements	Other Frameworks
OIS-01	INFORMATION SECURITY MANAGEMENT SYSTEM	The CSP operates an information security management system (ISMS). The scope of the ISMS covers the CSPs organisational units, locations and processes for providing the cloud service.	Organisation of Information Security ↑	Requirements ↓	Similar Controls ↓

Requirements (Control: OIS-01)

Home » Framework: EUCS » Category: Organisation of Information Security » Control: OIS-01 » Requirements

Code	Description	Assurance Level	Type	Control	Implementation guidelines	Metrics
OIS-01.B	The CSP shall have an information security management system (ISMS), covering at least the operational units, locations, people and processes for providing the cloud service.	translation-not-found[cocGatewayApp.AssuranceLevel.Basic]	Organizational	OIS-01 ↑	----	No Metrics
OIS-01.2B	The CSP shall provide documented information of the ISMS applied to the cloud service.	translation-not-found[cocGatewayApp.AssuranceLevel.Basic]	Organizational	OIS-01 ↑	----	No Metrics

Figure 12. Requirements belonging to the OIS-01 Control

2.2.4. Requirements

The main *Requirements* window (see Figure 13) shows the list of Requirements registered in the *Catalogue*.

Requirements

Home » Frameworks » Categories » Controls » Requirements

Code	Description	Assurance Level	Type	Control	Implementation guidelines	Metrics
OIS-01.B	The CSP shall have an information security management system (ISMS), covering at least the operational units, locations, people and processes for providing the cloud service.	translation-not-found[cocGatewayApp.AssuranceLevel.Basic]	Organizational	OIS-01 ↑	----	No Metrics
OIS-01.2B	The CSP shall provide documented information of the ISMS applied to the cloud service.	translation-not-found[cocGatewayApp.AssuranceLevel.Basic]	Organizational	OIS-01 ↑	----	No Metrics
OIS-01.1S	The CSP shall have an information security management system (ISMS), covering at least the operational units, locations, people and processes for providing the cloud service, in accordance with EN ISO/IEC 27001. Where the controls referred to in ISO/IEC 27001 6.1.3 shall be the controls in this TS on level Substantial	translation-not-found[cocGatewayApp.AssuranceLevel.Substantial]	Organizational	OIS-01 ↑	----	No Metrics

Figure 13. List of Requirements

The following fields are listed for each Requirement:

- Code
- Description
- Assurance Level
- Type

The list of requirements in Figure 13 can be customized using the implemented filters (code, description, type, and assurance level) as shown in Figure 14. In addition, a Requirement can be edited by clicking on the *Edit* button (see Figure 15).

Figure 14. Filtering of Requirements

Figure 15. Edit the details of a Requirement

For each Requirement listed in Figure 13, the user can go back to the Controls window or can access its related Metrics (if any) (see Figure 16), by clicking on the following links:

OPS-05.2H	Signature-based and behaviour-based malware protection tools shall be updated at least daily.	translation-not-found[cocGatewayApp.AssuranceLevel.High]	Organizational	OPS-05 ↑	---	No Metrics	View	Edit
OPS-05.3H	The CSP shall automatically monitor the systems covered by the malware protection and the configuration of the corresponding mechanisms to guarantee fulfillment of above requirements, and the antimalware scans to track detected malware or irregularities.	translation-not-found[cocGatewayApp.AssuranceLevel.High]	Organizational	OPS-05 ↑	Implementation guideline	Metrics ↓	View	Edit

Category	Name	Source	Description	Operator	Requirements	
Operational security	MalwareProtectionEnabled	Technical	This metric is used to assess if the antimalware solution is enabled on the respective resource.	==	OPS-05.3H ↑	View
Operational security	NumberOfThreatsFound	Technical	This metric is used to assess if the antimalware solution reports no irregularities.	==	OPS-05.3H ↑	View
Operational security	MalwareProtectionOutput	Technical	This metric states whether automatic notifications are enabled (e.g. e-mail) about malware threats. This relates to EUCS' definition of "continuous monitoring".	==	OPS-05.3H ↑	View

Figure 16. Metrics implemented for the OPS-05.3H Requirement

2.2.5. Metrics

The main *Metrics* window (see Figure 17) shows the list of all the registered metrics in the *Catalogue*.

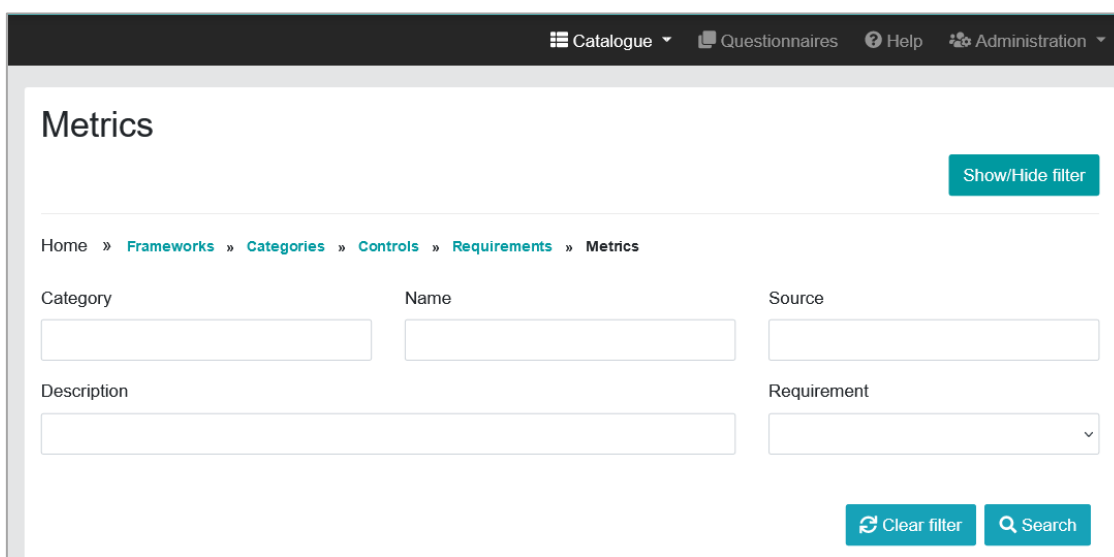
Category	Name	Source	Description	Operator	Requirements	
Operational security	MalwareProtectionEnabled	Technical	This metric is used to assess if the antimalware solution is enabled on the respective resource.	==	OPS-05.3H ↑	View
Operational security	NumberOfThreatsFound	Technical	This metric is used to assess if the antimalware solution reports no irregularities.	==	OPS-05.3H ↑	View
Operational security	BackupEnabled	Technical	This metric is used to assess if backups are enabled for a cloud service/asset	==	OPS-07.2H ↑	View
Operational security	BackupRetentionSet	Technical	This metric is used to assess the configured backup retention (days) on a cloud service/asset	>	OPS-07.2H ↑	View

Figure 17. List of Metrics

The following fields are listed for each Metric:

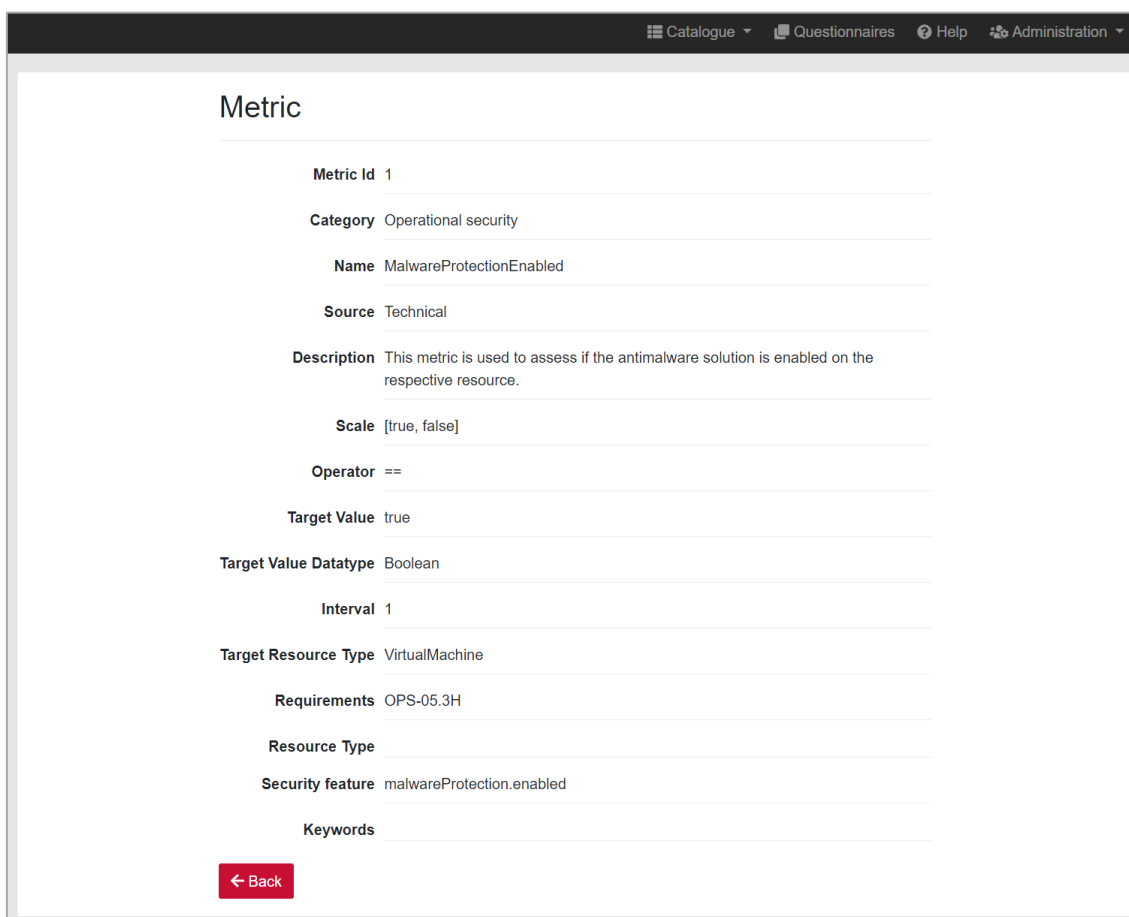
- Category
- Name
- Source
- Description
- Operator

The list of Metrics in Figure 17 can be customized using the implemented filters (category, name, source, description, and requirement), as shown in Figure 18. Also, the user can consult the details of each Metric by clicking on the *View* button (see Figure 19).



The screenshot shows the 'Metrics' page in the MEDINA Catalogue. At the top, there is a navigation bar with links for 'Catalogue', 'Questionnaires', 'Help', and 'Administration'. Below the navigation bar, the page title 'Metrics' is displayed. A 'Show/Hide filter' button is located in the top right corner. The main content area contains a breadcrumb trail: 'Home » Frameworks » Categories » Controls » Requirements » Metrics'. Below the breadcrumb, there are five input fields for filtering: 'Category', 'Name', 'Source', 'Description', and 'Requirement'. The 'Requirement' field is a dropdown menu. At the bottom right, there are two buttons: 'Clear filter' and 'Search'.

Figure 18. Filtering of Metrics



The screenshot shows the 'Metric' details page. The page title is 'Metric'. Below the title, there is a list of fields and their values:

- Metric Id: 1
- Category: Operational security
- Name: MalwareProtectionEnabled
- Source: Technical
- Description: This metric is used to assess if the antimalware solution is enabled on the respective resource.
- Scale: [true, false]
- Operator: ==
- Target Value: true
- Target Value Datatype: Boolean
- Interval: 1
- Target Resource Type: VirtualMachine
- Requirements: OPS-05.3H
- Resource Type:
- Security feature: malwareProtection.enabled
- Keywords:

At the bottom left, there is a red button labeled 'Back'.

Figure 19. View the details of a metric

For each metric listed in Figure 17, the user can go to its related requirements by clicking on the corresponding link.

2.2.6. Similar Controls

The *Similar Controls* window (see Figure 20) shows the list of Controls belonging to other Security Frameworks that are equivalent to each of the controls of the EUCS Security Framework.

EUCS Control ID	EUCS Control Name	Framework	Similar Control ID	Similar Control Name
OIS-01	INFORMATION SECURITY MANAGEMENT SYSTEM	C5.2020 GERMANY	OIS-01	Information Security Management System (ISMS)
OIS-02	SEGREGATION OF DUTIES	C5.2020 GERMANY	OIS-04	Segregation of Duties
OIS-03	CONTACT WITH AUTHORITIES AND INTEREST GROUPS	C5.2020 GERMANY	OIS-05	Contact with Relevant Government Agencies and Interest Groups
OIS-04	INFORMATION SECURITY IN PROJECT MANAGEMENT	C5.2020 GERMANY	OIS-05	Contact with Relevant Government Agencies and Interest Groups

Figure 20. List of Similar Controls

The following fields are listed:

- EUCS Control ID
- EUCS Control Name
- Framework (other than EUCS) e.g., C5-2020 GERMANY
- Similar Control ID in that framework
- Similar Control Name in that framework

The list of Similar Controls in Figure 20 can be customized using the implemented filters (EUCS Control ID, EUCS Control name and Framework), as shown in Figure 32. Also, each Similar Control can be edited by clicking on the *Edit* button (see Figure 22).

Figure 21. Filtering of Similar Controls

Edit a Similar Control

EUCS Control

OIS-01: INFORMATION SECURITY MANAGEMENT SYSTEM

Framework

C5.2020 GERMANY

Similar Control ID

OIS-01

Similar Control Name

Information Security Management System (ISMS)

[Cancel](#) [Save](#)

Figure 22. Edit the details of a Similar Control

2.2.7. Implementation Guidelines

An Implementation guideline is an explanation of how a specific security requirement can be implemented, in a vendor and technology-agnostic way. Examples from larger CSPs are provided for inspiration.

The *Implementation guidelines* window (see Figure 23) shows the list of Implementation Guidelines included in the Catalogue. The following fields are shown for each Implementation Guideline:

- Requirement Code
- Requirement Description
- Control Code
- Control Name

Implementation guidelines [Refresh list](#)

Home » Frameworks » Categories » Controls » Requirements » Implementation guidelines

Requirement Code	Requirement Description	Control Code	Control Name	
OIS-02.4H	The CSP shall automatically monitor the assignment of responsibilities and tasks to ensure that measures related to segregation of duties are enforced.	OIS-02	SEGREGATION OF DUTIES	View Edit
ISP-03.5H	The list of exceptions shall be automatically monitored to ensure that the validity of approved exceptions has not expired and that all reviews and approvals are up-to-date.	ISP-03	EXCEPTIONS	View Edit
HR-03.4H	All employees shall acknowledge in a documented form the information security policies and procedures presented to them before they are granted any access to CSC data, the production environment, or any functional component thereof, and the verification of this acknowledgement shall be automatically monitored in the processes and automated systems used to grant access rights to employees.	HR-03	EMPLOYEE TERMS AND CONDITIONS	View Edit

Figure 23. List of Implementation guidelines

Figure 24 shows the details of an Implementation guideline.

Catalogue
Questionnaires
Help

Implementation guideline

Requirement OIS-02.4H

The EUCS requirement OIS-02.4H states:

"The CSP shall automatically monitor the assignment of responsibilities and tasks to ensure that measures related to segregation of duties are enforced".

and references as "measures" the following requirement also from OIS-02 Segregation of Duties:

OIS-02.1H	<p>"The CSP shall perform a risk assessment as defined in RM-01 about the accumulation of responsibilities or tasks on roles or individuals, regarding the provision of the cloud service, covering at least the following areas, insofar as these are applicable to the provision of the cloud service and are in the area of responsibility of the CSP:</p> <p>(1) Administration of rights profiles, approval and assignment of access and access authorisations (cf. IAM-01);</p> <p>(2) Development, testing and release of changes (cf. DEV-01, CCM-01); and</p> <p>(3) Operation of the system components."</p>
-----------	--

EUCS Security Control

Code	Name	Objective
OIS-02	Segregation of Duties	"Conflicting tasks and responsibilities are separated based on an RM-01 risk assessment to reduce the risk of unauthorised or unintended changes or misuse of CSC data processed, stored or transmitted in the cloud service."

References

Internal references:

- EUCS - RM-01: Risk Management Policy
- EUCS - IAM-01: Policies for Access Control to Information
- EUCS - DEV-01: Policies for the Development and Procurement of Information Systems
- EUCS - CCM-01: Policies for Changes to Information Systems

External references:

- 2020 GERMANY - OIS-04: Segregation of Duties
- SecNumCloud FRANCE - 6.1: Functions and responsibilities linked to information security
- SecNumCloud FRANCE - 6.2: Segregation of tasks
- ISO 27002 – 5.3: Segregation of duties
- ISO 27017 - CLD 6.3.1: Shared roles and responsibilities within a cloud computing environment
- Cisco CCF - CCF 91: Roles and Responsibilities over Security and Control Environment

Key concepts

Term	Definition
Management System	Framework of policies, procedures, and processes used to ensure that an organization is operating effectively and efficiently. Management systems are used to guide the activities of an organization and to help achieve its goals and objectives.
Information Security Management system (ISMS)	An information security management system (ISMS) is a framework of policies, processes, and controls that organizations use to manage and reduce their information security risks. Generally, an ISMS is designed to protect the confidentiality, integrity, and availability of the organization's information assets, and can include both technical and non-technical measures.
Risk Management	Overall process of risk identification, risk analysis and risk evaluation. An ISMS includes a process for identifying and assessing the organization's information security risks, and for developing plans to mitigate those risks.
Segregation/ Separation of Duties	The goal of segregation of duties is to ensure that no single individual has complete control over a process or activity, which can help to prevent unauthorized actions and mistakes. Also, it allows to separate conflicting duties between different individuals. So, it is a principle that is used in ISMS to reduce among others the risk of fraud and errors.
Cloud RBAC	Cloud role-based access control is an authorization system provided by the CSP that provides fine-grained access management of Cloud resources to ensure that measures related to segregation of duties are enforced.
Role assignment	It is the process (grant, change, revoke) of attaching a role definition to a security principal at a particular scope.

Guidelines

Typically, managing access to cloud resources is a critical function and is performed by the CSP by implementing a cloud RBAC (e.g., Azure RBAC, AWS RBAC) to manage who has access to specific cloud resources, what they can do with those resources, and what areas they have access to. The assignment of tasks to roles will allow a separation of duties as part of the role management process. The role assignment is monitored by the CSP.

A defined team shall be defined that is responsible for overseeing the security and control environments at the organization. It will verify the roles of each member and validate that security and control environments are being reviewed and followed up upon. Managers will check with each member to review responsibilities and roles at least annually.

Roles and responsibilities of the users are defined and agreed on in a risk assessment performed by the CSP. The risk assessment should cover administrative and user rights, and should include definitions related to data ownership, information security accountability, access provisioning and approval responsibilities, development, testing and release of changes, data backup and recovery responsibilities, and operation of the system components. Some mitigation measures should be introduced to monitor the activities in order to detect unauthorised or unintended changes as well as misuse.

A risk assessment for administrative user rights should consider the potential risks associated with granting certain users the ability to modify or delete logs or log analysis of their actions. This could include risks such as:

- Tampering with logs to cover up malicious or inappropriate activity.
- Accidentally or intentionally deleting important logs that may be needed for later analysis or investigation.
- Disrupting the integrity and reliability of log data, which could hinder incident response and forensics efforts.

To mitigate these risks, it is important to carefully consider which users should be granted administrative rights and to establish strict policies and procedures for the use of these rights. This might include requiring users to provide a justification for modifying or deleting logs, requiring multiple approvals before such actions can be taken, and implementing strict auditing and monitoring to detect any inappropriate use of these rights.

This risk assessment should also consider that a same user could have several roles which gives him different right and duties.

Back
Edit

Figure 24. Details of an Implementation guideline

2.3. Questionnaires


The Questionnaires allow a CSP to perform a self-assessment of the fulfilment degree of the EUCS certification scheme for various levels of certifications (Basic, Substantial and High), defining one or more questions for each security requirement. The user can select the assurance level for the assessment, and then provide the answer to several questions to check the fulfilment of every requirement involved. It also allows the user to enter comments related to a question, and textual references to locate the evidence supporting the answer given. Finally, it provides a summary dashboard with quantitative values to reflect the degree of fulfilment. Auditors can also have access to the questionnaire and enter non-conformities for each requirement that is not fulfilled.

2.3.1. Create a new Questionnaire

The user can create a new questionnaire by clicking on the *Questionnaires* menu option in the application toolbar (see Figure 1). To create a new questionnaire, the user must fill in the following fields as shown in Figure 25:

- **Framework:** Current version of the Catalogue only includes EUCS
- **Cloud Service:** Name of the cloud service in the MEDINA framework to be assessed
- **Assurance level:** Basic, Substantial or High

Figure 25. Start a new questionnaire

The button  in the top right corner displays the number of questions that are asked to the user depending on the level of assurance.

Questionnaires info	
	Number of questions
Basic level of assurance:	504
Substantial level of assurance:	857
High level of assurance:	1003
<div>Close</div>	

Figure 26. Number of questions for each level of assurance

When the user clicks on the *Start Questionnaire* button in Figure 25 a new questionnaire is created, and the window in Figure 27 is displayed. The panel on the left provides a navigator through which the distinct Categories of the EUCS Framework can be accessed.

Each page corresponds to a Control. A navigator at the top of the page allows the user to show other Controls in the same Category (see Figure 27).

The screenshot displays the 'Questionnaire' interface. At the top, there's a header with 'Catalogue', 'Questionnaires', and 'Help' links. Below the header, a teal bar shows the breadcrumb '2023-05-15 >> Bosch_laaS >> EUCS >> Basic'. The main content area is divided into several sections:

- Category navigator:** A vertical list on the left showing categories from A1 to A14. A1: Organisation of Information Security is highlighted.
- Current Category:** A1: Organisation of Information Security.
- Control navigator:** A horizontal bar with buttons for OIS-01, OIS-02, OIS-03, and OIS-04. OIS-01 is selected.
- Current Control:** OIS-01: The CSP operates an information security management system (ISMS). The scope of the ISMS covers the CSPs organisational units, locations and processes for providing the cloud service.
- Requirement:** OIS-01.1B: The CSP shall have an information security management system (ISMS), covering at least the operational units, locations, people and processes for providing the cloud service.
- Questions:** A section containing three questions (Q1, Q2, Q3) with radio button options for 'Fully supported', 'Partially supported', 'Not supported at all', and 'Not applicable'. Each question has an 'Evidence' field and a 'Comments' field.

Figure 27. Questionnaire structure

The Control page shows all the Requirements belonging to it, and for each Requirement the page shows all the questions that the CSP must reply during the self-assessment process. Figure 28 shows the details of a question which includes a field for registering evidence to support the answer, and another field for entering comments.

This figure shows a detailed view of question Q1: 'Does the CSP have an information security management system (ISMS) documented?'. It includes the same four radio button options as in Figure 27. The 'Evidence' field contains the text '- Documented Information Security Management System (ISMS)'. The 'Comments' field is empty.

Figure 28. Details of a question

Each question has four possible answers:

- Fully supported
- Partially supported
- Not supported at all
- Not applicable

The degree of *Compliance* with a Requirement is calculated based on the answers provided by the CSP to the corresponding questions and is displayed at the bottom of the requirement (see Figure 29). This *Compliance* value is calculated according to the following rules:

Answer to Questions	Compliance
All "Fully supported" or "Not applicable"	YES
All "Not supported at all" or "Not applicable"	NO

All “Not applicable”	N/A
Any “Not supported at all”	PARTIAL
Any “Partially supported”	PARTIAL

OIS-01.2B: The CSP shall provide documented information of the ISMS applied to the cloud service.

Q1: Does the CSP provide documented information of the ISMS applied to the cloud service?

☒ Fully supported.
☐ Partially supported.
☐ Not supported at all.
☐ Not applicable.

Evidence: - Documented information of the ISMS applied to the cloud service

Comments:

Non-conformities of the requirement:

Compliance: YES

Figure 29. Compliance value for a requirement

Finally, at the bottom of the Control page there are buttons to *Exit* the questionnaire and to go to the *Previous/Next* Security Control.



2.3.2. Manage Questionnaires

When there is at least one questionnaire stored in the *Catalogue* the user can select a previously created questionnaire (see Figure 30) to load it for further editing or just for reading (see Figure 31), depending on the user’s role (see section 1.1).

Figure 30. Load an existing questionnaire

Users with the “Auditor” role can edit the questionnaire and fill in the *Non-conformities* field to indicate non-conformities related to the compliance of the requirement (see Figure 32).

Finally, a questionnaire can also be removed by clicking on the *Remove Questionnaire* button (see Figure 30). This option is only available for the “Product and Service Owner” role.

Questionnaire:

2023-05-15 >> Bosch_ IaaS >> EUCS >> Basic

Categories

- A1: Organisation of Information Security
- A2: Information Security Policies
- A3: Risk Management
- A4: Human Resources
- A5: Asset Management
- A6: Physical Security
- A7: Operational Security
- A8: Identity, Authentication and Access Control Management
- A9: Cryptography and Key Management
- A10: Communication Security
- A11: Portability and Interoperability

A1: Organisation of Information Security

Choose a Control: OIS-01 OIS-02 OIS-03 OIS-04

OIS-01: The CSP operates an information security management system (ISMS). The scope of the ISMS covers the CSPs organisational units, locations and processes for providing the cloud service.

OIS-01.1B: The CSP shall have an information security management system (ISMS), covering at least the operational units, locations, people and processes for providing the cloud service.

Q1: Does the CSP have an information security management system (ISMS) documented?

☒ Fully supported.
☐ Partially supported.
☐ Not supported at all.
☐ Not applicable.

Evidence: - Documented Information Security Management System (ISMS)

Comments:

Q2: Does the information security management system cover the operational units?

☒ Fully supported.
☐ Partially supported.
☐ Not supported at all.

Evidence: - ISMS scope (operational units)

Comments:

Figure 31. Edit/Read an existing questionnaire

Q7: Does the CSP cover processes for providing the cloud service?

☐ Fully supported.
☐ Partially supported.
☐ Not supported at all.
☐ Not applicable.

Evidence: - ISMS scope (processes for providing the cloud service).

Comments:

Non-conformities of the requirement:

Figure 32. Edit a questionnaire (Auditor role)

2.3.3. Generate a Report

By clicking on the *Generate report* button (see Figure 30), a report in PDF format containing the evaluation results of the questionnaire is stored in the file system. A screenshot of a section of the report is shown in Figure 33.

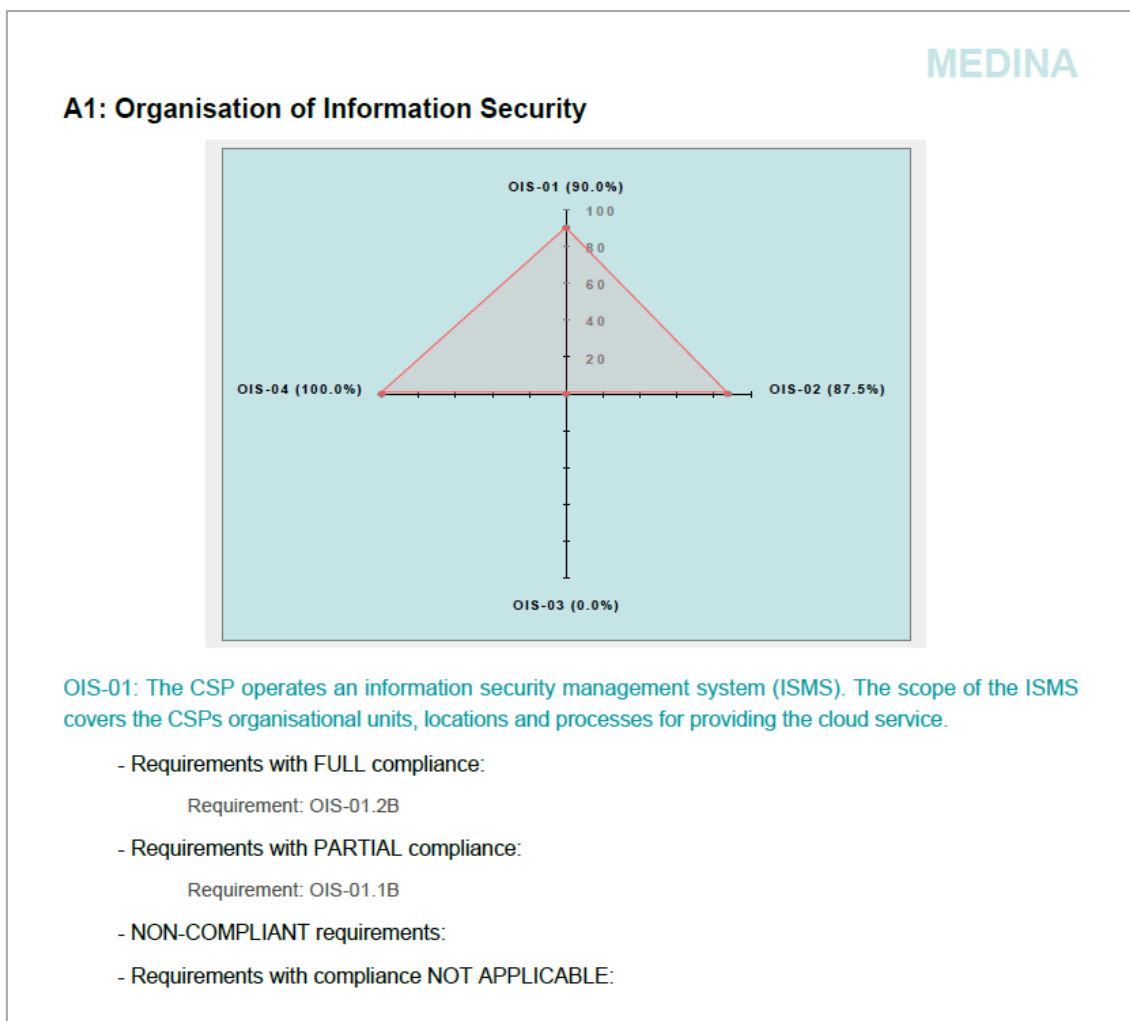


Figure 33. Report generated for a questionnaire

2.3.4. Connection to SATRA

The *Catalogue of Control and Metrics* is connected to SATRA⁵ (*Self-Assessment Tool for Risk Analysis*) through the Questionnaires. Every time a questionnaire is saved, those requirements for which compliance has been calculated are sent to SATRA (see Figure 34).

Q3: Does the information security management system (ISMS), cover locations?

☒ Fully supported.
☐ Partially supported.
☐ Not supported at all.
☐ Not applicable.

Evidence:

Comments:

Q4: Does the CSP cover processes for providing the cloud service?

☒ Fully supported.
☐ Partially supported.
☐ Not supported at all.
☐ Not applicable.

Evidence:

Comments:

Non-conformities of the requirement:

Compliance:

Figure 34. Calculated compliance

⁵ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D2.8 <https://doi.org/10.5281/zenodo.7927217>

A questionnaire is saved each time the following actions are performed, as long as the user has the role of “Product and Service Owner” or “Auditor”:

- The user clicks on the *Exit*, *Previous* or *Next* buttons
- The user changes to another Category through the Category navigator
- The user changes to another Control page through the Control navigator

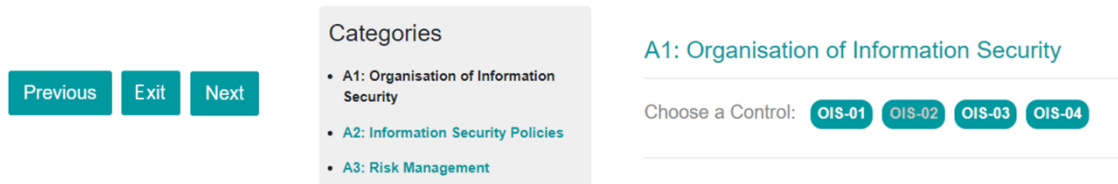


Figure 35. Actions that cause the saving of a questionnaire

Compliance values for all requirements are sent to SATRA in JSON format, similar to the following example:

```
{
  "assurance_level": 1,
  "certification_schema": 1,
  "data": "2023-05-16",
  "partner_survey": [
    {
      "question_id": "1",
      "related": "OIS-01.1B",
      "answer_value": 1
    },
    {
      "question_id": "2",
      "related": "OIS-01.2B",
      "answer_value": 3
    },
    {
      "question_id": "3",
      "related": "OIS-01.3B",
      "answer_value": 3
    }
  ]
}
```

2.4. Administration

The administrator can access the administration options by clicking on the *Administration* menu option in the application toolbar (see Figure 1). This menu option displays the following submenu options, which will be detailed below:

- Gateway
- API
- Audit Logs

2.4.1. Gateway

The *Gateway* window shows the status of all the available microservices that make up the *Catalogue* architecture, as shown in Figure 36.

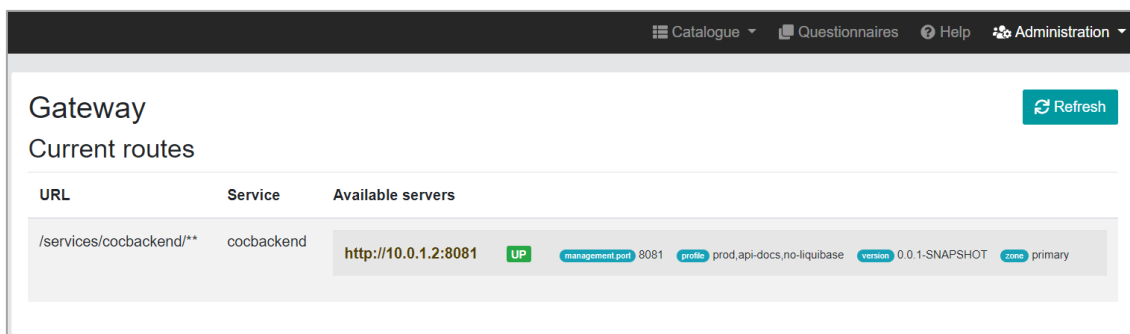


Figure 36. Administration menu – Gateway

2.4.2. API

The API menu option opens a Swagger User Interface to operate with the available REST API in the *Catalogue*, as shown in Figure 37.

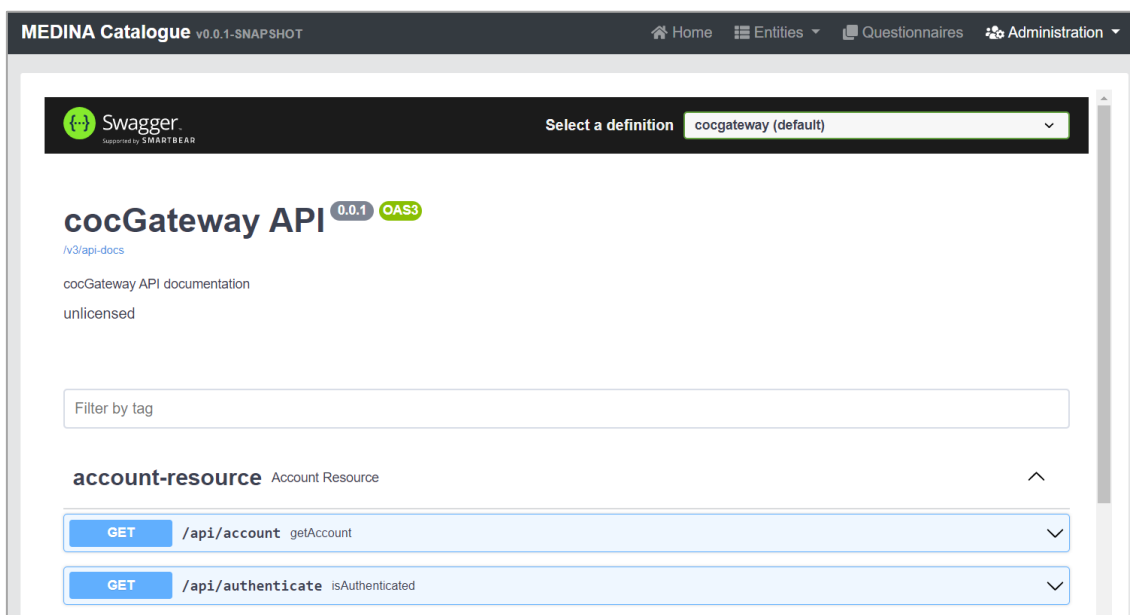
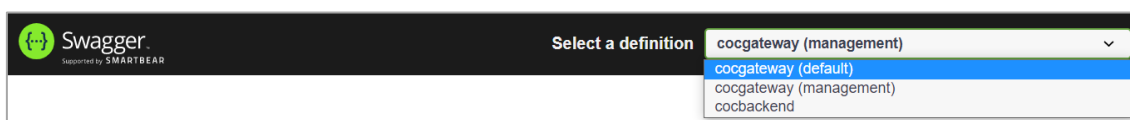


Figure 37. Administration menu – API

Both the Backend and the Frontend subcomponents of the Catalogue have their own independent REST API, that can be consulted by choosing the corresponding option in the select box:

- **Frontend:** cocgateway (default)
- **Backend:** cocbackend



Through the Backend API, operations on the Entities (Frameworks, Categories, Controls, Requirements, Metrics, Similar Controls, and Implementation Guidelines) and on the Questionnaires can be executed. For example, the available operations for Frameworks are:

security-control-framework-resource Security Control Framework Resource			^
GET	/api/security-control-frameworks	getAllSecurityControlFrameworks	✓
GET	/api/security-control-frameworks-full	getAllSecurityControlFullFrameworks	✓
GET	/api/security-control-frameworks/checkHasRequirements/{name}	checkHasRequirements	✓
GET	/api/security-control-frameworks/count	countSecurityControlFrameworks	✓
GET	/api/security-control-frameworks/{id}	getSecurityControlFramework	✓
PUT	/api/security-control-frameworks/{id}	updateSecurityControlFramework	✓
PATCH	/api/security-control-frameworks/{id}	partialUpdateSecurityControlFramework	✓

And following the same example, the operation to obtain all the information about a Framework given its ID is the following:

GET	/api/security-control-frameworks/{id}	getSecurityControlFramework	^
Parameters			
Name Description			
id * required			
integer(\$int64)id			
(path)			
id - id			
Execute			

Entering a Framework ID and clicking on *Execute*, the result obtained is the following:

```
{
  "id": 1,
  "name": "EUCS",
  "description": "EU Cloud Services certification scheme",
  "version": "December 2020"
}
```

Finally, the Frontend API is mainly used to operate on users and accounts:

account-resource	Account Resource	^
GET	/api/account getAccount	▼
GET	/api/authenticate isAuthenticated	▼
auth-info-resource	Auth Info Resource	^
GET	/api/auth-info getAuthInfo	▼
gateway-resource	Gateway Resource	^
GET	/api/gateway/routes activeRoutes	▼
logout-resource	Logout Resource	^
POST	/api/logout logout	▼

2.4.3. Audit logs

The *Audit logs* window (see Figure 38) lists all the operations performed by the users on the following entities of the Catalogue:

- Frameworks
- Categories
- Controls
- Requirements
- Metrics
- Implementation Guidelines
- Similar Controls

Catalogue Questionnaires Help Administration				
Audit Logs				
Refresh list				
Date	User	Entity	Identifier	Operation
2023-05-15 12:18:06	uct_segov	Implementation Guidelines	HR-03.4H	UPDATE
2023-05-15 12:18:00	uct_segov	Controls	OPS-02	UPDATE
2023-05-15 12:17:57	uct_segov	Controls	HR-02	UPDATE
2023-05-15 12:17:54	uct_segov	Controls	CIS-02	UPDATE
2023-05-15 12:17:50	uct_segov	Categories	Information Security Policies	UPDATE
2023-05-15 12:17:42	uct_segov	Requirements	CIS-01.2S	UPDATE
2023-05-15 12:17:38	uct_segov	Requirements	CIS-01.1B	UPDATE
2023-05-15 12:17:28	uct_segov	Categories	Information Security Policies	UPDATE

Figure 38. List of Audit logs

In this way, each time one of the aforementioned elements is updated, a record is stored in the Catalogue with the following information:

- **Date:** date and time when the operation has been performed
- **User:** user who conducted the operation
- **Entity:** Frameworks / Categories / Controls / Requirements / Implementation Guidelines / Similar Controls
- **Identifier:** ID of the updated element

- **Operation:** this current version of the Catalogue only accepts the UPDATE operation.

3. Delivery and Usage

3.1. Licensing information

This component is offered under Apache 2.0 license. The license files and more detailed information can be found in the MEDINA Public GitLab repository⁶.

3.2. Download

The code of the component is available at the public GitLab repository of the MEDINA project:

<https://git.code.tecnalia.com/medina/public/catalogue-of-controls>

3.3. More information

Interested readers can find more information about the Catalogue at this link:

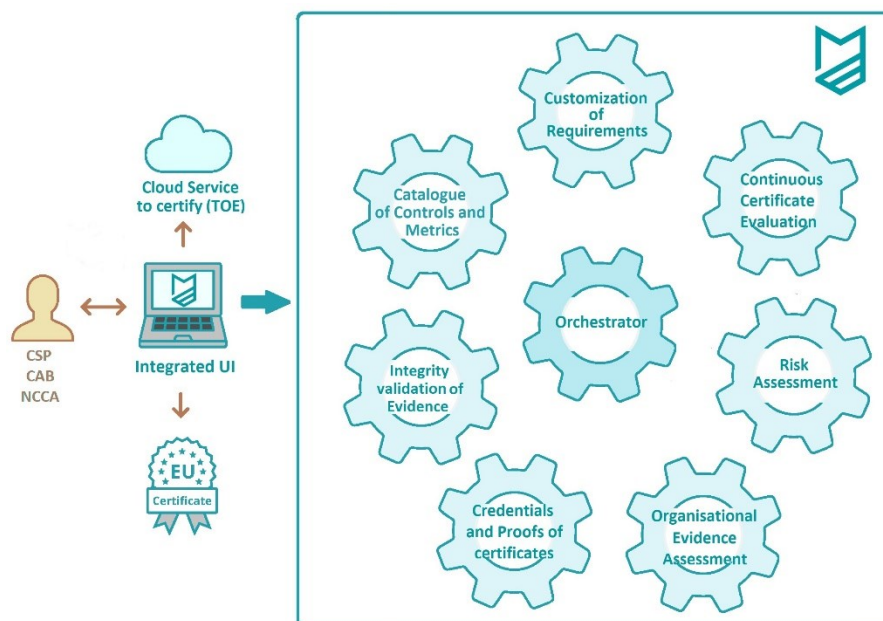
<https://doi.org/10.5281/zenodo.7794478> “D2.2 Continuously certifiable technical and organizational measures and Catalogue of cloud security metrics-v2”

The MEDINA web site (<https://medina-project.eu/>) also includes several deliverables and blog posts related to the Catalogue of Controls and Metrics.

⁶ <https://git.code.tecnalia.com/medina/public/catalogue-of-controls>

Orchestrator

– User Manual –



Project Title:	MEDINA - Security Framework to achieve a continuous audit-based certification in compliance with the EU-wide cloud security certification scheme
Project Number:	952633
Editor:	Immanuel Kunz (FhG)
Version:	v1.0
Date:	31.07.2023
Distribution level:	PU



Table of contents

- 1. Introduction 4
 - 1.1. User Roles and Permissions 4
- 2. User Manual 5
 - 2.1. Toolbar 5
 - 2.2. Cloud Services 5
 - 2.3. Metrics 6
 - 2.4. Catalogues 7
 - 2.5. Certificates 8
- 3. Delivery and usage 9
 - 3.1. Licensing information 9
 - 3.2. Download 9
 - 3.3. More information 9

List of Figures

FIGURE 1. THE FOUR MAIN VIEWS OF THE ORCHESTRATOR UI 5

FIGURE 2. OVERVIEW OF AN INDIVIDUAL CLOUD SERVICE..... 5

FIGURE 3. THE DISCOVERY OVERVIEW OF A CLOUD SERVICE WHICH SHOWS THE RESOURCES THAT HAVE BEEN
DISCOVERED. THE "SHOW MORE INFO" BUTTON REVEALS MORE DETAILED INFORMATION ABOUT THE
RESOURCE 6

FIGURE 4. THE ASSESSMENT TAB OF A CLOUD SERVICE. IT ALLOWS TO FILTER THE ASSESSMENT RESULTS FOR
VARIOUS PARAMETERS, AND ALSO REVEALS DETAILED INFORMATION ABOUT ANY ASSESSMENT RESULTS VIA
THE "SHOW MORE INFO" BUTTON 6

FIGURE 5. THE METRICS VIEW: IT SHOWS CONFIGURED METRICS WITH THEIR NAME, CATEGORY, DESCRIPTION,
AND IMPLEMENTATION 7

FIGURE 6. THE VIEW WHEN OPENING THE CODE OF A METRIC'S IMPLEMENTATION..... 7

FIGURE 7. THE OVERVIEW OF A CATALOGUE IS SHOWN WHEN OPENING THE CATALOGUES VIEW 7

FIGURE 8. THE OVERVIEW OF THE EUCS CATEGORIES 8

FIGURE 9. AN EXAMPLE OF A CERTIFICATE AS DISPLAYED IN THE CERTIFICATES VIEW IN THE ORCHESTRATOR UI . 8

1. Introduction

The *Orchestrator* is a central component of the MEDINA framework and processes and stores all evidence and assessment results. It receives them from the evidence collection and security assessment tools, and forwards them to the appropriate components, such as the *Continuous Certificated Evaluation*¹ (CCE). Furthermore, it provides a database that stores evidence and assessment results, as well as metrics, and other data.

Via its graphical user interface, the *Orchestrator* additionally provides users with multiple possibilities to review and manage cloud services, metrics, and many other information.

1.1. User Roles and Permissions

Access to *the Orchestrator* is managed by Keycloak². The operations that are allowed to be carried out are conditioned by the role to which each authenticated user is assigned. The cloud services that are shown are filtered according to the read permissions by the authenticated user. Similarly, the certificates are filtered. The Metrics and Catalogues views, however, are not filtered.

¹ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D4.3 <https://doi.org/10.5281/zenodo.7927231>

² <https://www.keycloak.org>

2. User Manual

2.1. Toolbar

The *Orchestrator* offers four main views: Cloud Services, Metrics, Catalogues, and Certificates (see Figure 1).

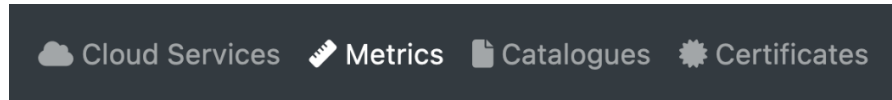


Figure 1. The four main views of the Orchestrator UI

2.2. Cloud Services

The Cloud Services view is the main view of the *Orchestrator* UI. It first presents an overview of existing cloud services and the possibility to create a new one.

- **Creating a new cloud service:** To create a new cloud service click on the *Add service* button and enter a name and description. Then click on *Save*. This will save the new service in the *Orchestrator*.
- **Creating a new Target of Evaluation:** Creating a new cloud service does not trigger its evaluation. To trigger its evaluation for a certain certification schema, a Target of Evaluation (ToE) must be created. To do so, click on a cloud service and navigate to the Configuration tab (see Figure 2). Within the Configuration tab, click on Target of Evaluation and select the desired certification schema and assurance level.
- **Deleting a cloud service or ToE:** To delete a cloud service or ToE, navigate to the respective overview and click on the red button (see Figure 2).

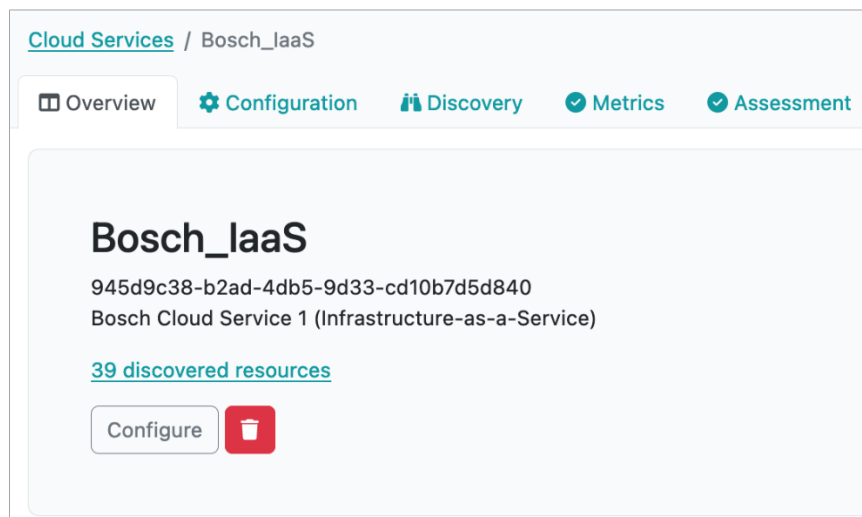


Figure 2. Overview of an individual cloud service

Additionally, to the configuration options described above, multiple tabs within a cloud service present information about resources, metrics, and assessment results. The Discovery tab shows which resources have been discovered for the service (see Figure 3). The Metrics tab presents similar information to the Metrics view described in Section 2.3, but filtered for the respective cloud service. Finally, the Assessment tab shows the assessment results that have been submitted related to that cloud service (see Figure 4).

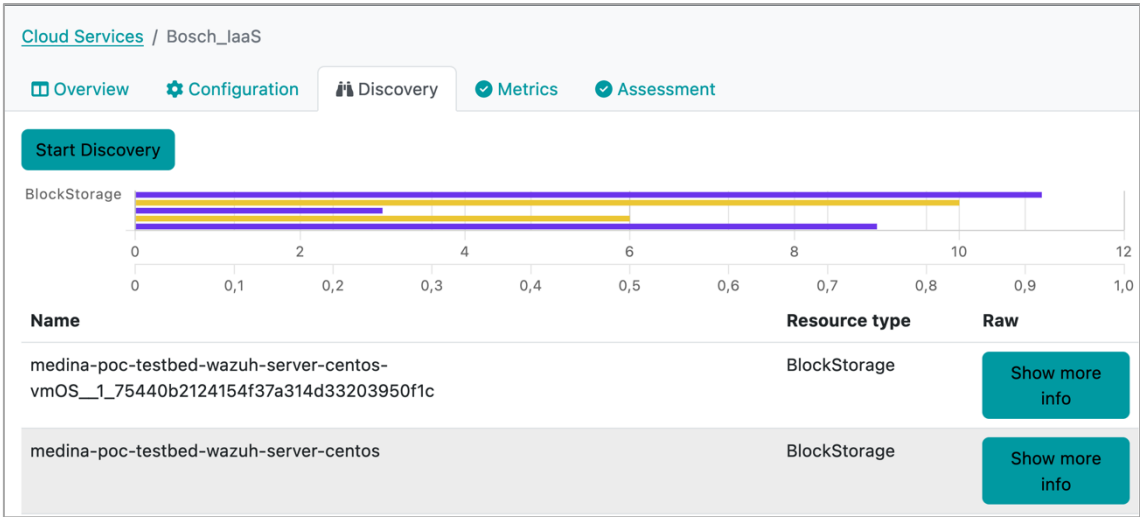


Figure 3. The Discovery overview of a cloud service which shows the resources that have been discovered. The "Show more info" button reveals more detailed information about the resource

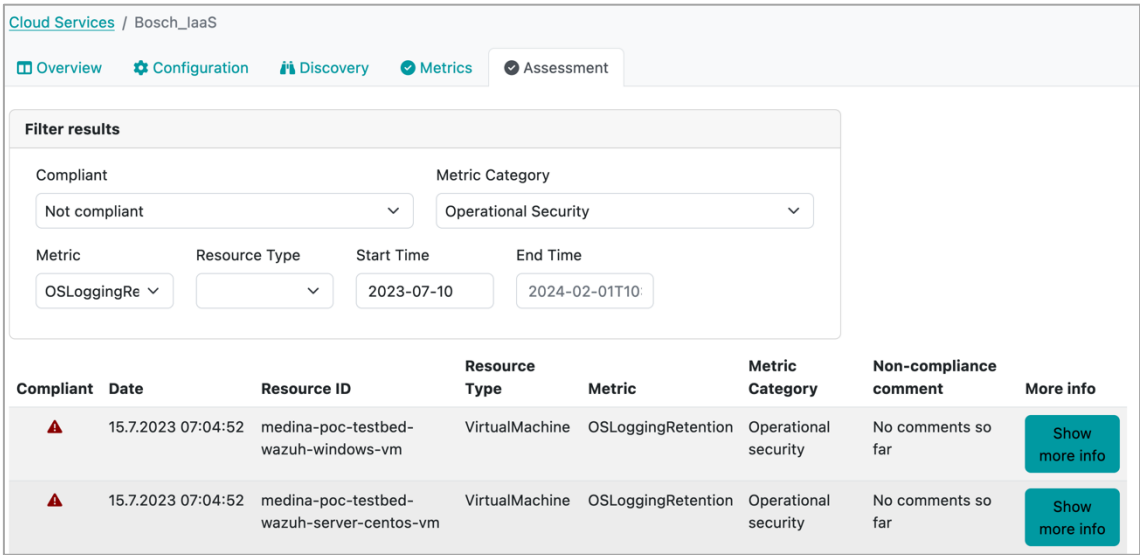


Figure 4. The Assessment tab of a cloud service. It allows to filter the assessment results for various parameters, and also reveals detailed information about any assessment results via the "Show more info" button

2.3. Metrics

The Metrics view presents information about metrics stored in the *Orchestrator's* database along with their configuration. When opening the view, the user is presented with information like shown in Figure 5.

For viewing details about a metric's concrete implementation, the user can click the *Show Code* button and is then presented with the information shown in Figure 6.

Configured Metrics

The following metrics are configured in the Clouditor orchestrator.

#	Category	Description	Implementation
ActivityLoggingEnabled	Operational security	This metric is used to assess if activity logs are enabled for the cloud service/asset.	Language: LANGUAGE_REGO Show Code
AnomalyDetectionEnabled	Operational security	This metric is used to assess if Anomaly Detection is enabled for the cloud service/asset	Language: LANGUAGE_REGO Show Code

Figure 5. The Metrics view: It shows configured metrics with their name, category, description, and implementation

```

Language: LANGUAGE_REGO

Show Code

package clouditor.metrics.activity_logging_enabled

import data.clouditor.compare

default applicable = false

default compliant = false

enabled := input.activityLogging.enabled

applicable {
    enabled != null
}

compliant {
    compare(data.operator, data.target_value, enabled)
}

```

Figure 6. The view when opening the code of a metric's implementation

2.4. Catalogues

The Catalogues view presents information about catalogues stored in the *Orchestrator*. As MEDINA focuses on the EUCS, this view presents the EUCS catalogue (see Figure 8), and shows details about its categories and controls when opening the catalogue (see Figure 9).

Catalogues

EUCS
EU Cloud Services certification scheme

Figure 7. The overview of a catalogue is shown when opening the Catalogues view

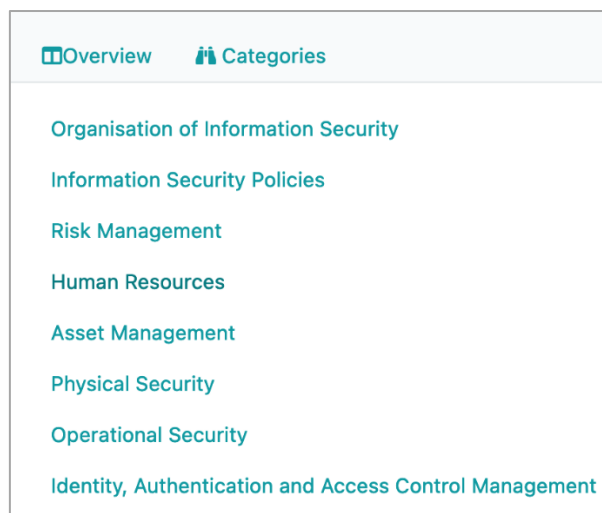


Figure 8. The overview of the EUCS categories

2.5. Certificates

The Certificates view visualizes information about the certificates that have been created via the *Automated Certificate Life Cycle Manager*³.

Figure 9 shows how certificates are displayed in the Certificates view. They include basic information, such as the name, the corresponding cloud service ID, and the certification schema, as well as information about the state history of the certificate.

"Bosch_IaaS"

ID: 2111
Name: Bosch_IaaS
Service ID: 945d9c38-b2ad-4db5-9d33-cd10b7d5d840
Issue Date: 2023-03-27T10:06:55Z
Expiration Date: 2024-03-27T10:06:54Z
Schema: EUCS
Assurance Level: high
CAB: CAB123
Description: Bosch IaaS

State History

State	Deviation	Timestamp	Tree ID
new		28 Jun 23 10:00 UTC	123456
suspended	major	30 Jun 23 08:01 UTC	223456
continued	minor	30 Jun 23 08:16 UTC	234567

Figure 9. An example of a certificate as displayed in the Certificates view in the Orchestrator UI

³ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D4.3 <https://doi.org/10.5281/zenodo.7927231>

3. Delivery and usage

3.1. Licensing information

The *Orchestrator* is offered under Apache 2.0 license. The license files and more detailed information can be found in the MEDINA Public GitLab repository⁴.

3.2. Download

The code of the component is available at the public GitLab repository of the MEDINA project: <https://git.code.tecnalia.com/medina/public/orchestrator>.

3.3. More information

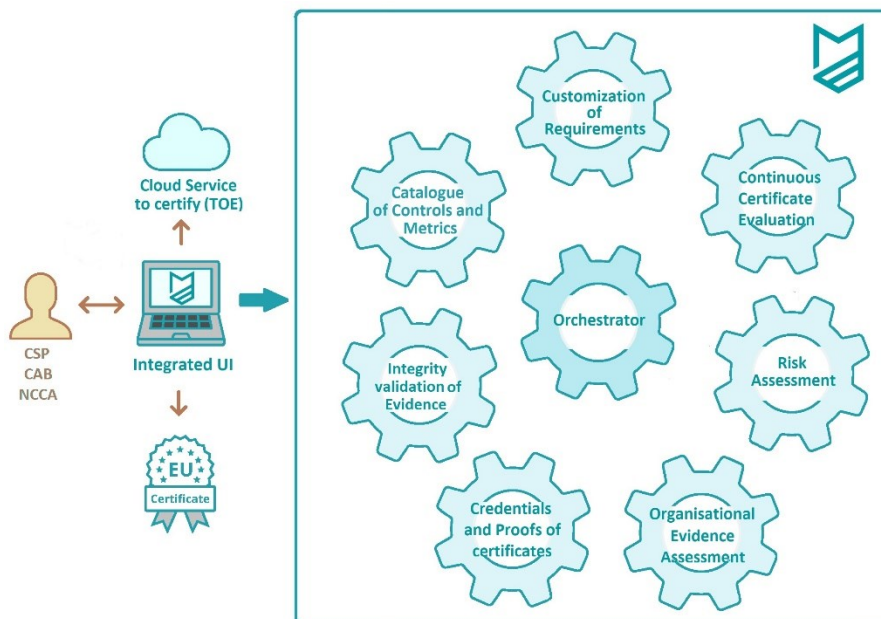
Interested readers can find more information about the Orchestrator at this link: <https://doi.org/10.5281/zenodo.7927225> “D3.6 Tools and techniques for collecting evidence of technical and organisational measures – v3”.

The MEDINA web site (<https://medina-project.eu/>) also includes several deliverables and blog posts related to the *Orchestrator*.

⁴ <https://git.code.tecnalia.com/medina/public/orchestrator>

Customization of Requirements

- User Manual -



Project Title:	MEDINA Security Framework to achieve a continuous audit-based certification in compliance with the EU-wide cloud security certification scheme
Project Number:	952633
Editor:	Patrizia Ciampoli (Hewlett Packard Italiana, SRL)
Version:	v1.0
Date:	31.07.2023
Distribution level:	PU



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952633

Table of contents

1. Introduction	4
1.1. User Roles and Permissions	4
2. User Manual	6
2.1. Toolbar	6
2.2. <i>List of REOs</i>	6
2.2.1. Show operation	8
2.2.2. Edit operation	9
2.2.3. Map operation	11
3. Delivery and usage	12
3.1. Licensing information	12
3.2. More information	12

List of Figures

FIGURE 1. CNL EDITOR TOOLBAR	6
FIGURE 2. LIST OF REOS	6
FIGURE 3. REO STATUS DIAGRAM	7
FIGURE 4. OPERATIONS AVAILABLE ON A SELECTED REO (STATUS “CUSTOMISED”).....	8
FIGURE 5. OPERATIONS AVAILABLE ON A SELECTED REO (STATUS “COMPLETED”)	8
FIGURE 6. SHOW THE DETAILS OF A REO: METADATA SECTION	9
FIGURE 7. SHOW THE DETAILS OF A REO: OBLIGATION SECTION	9
FIGURE 8. EDIT A REO: METADATA SECTION.....	10
FIGURE 9. EDIT A REO: OBLIGATIONS SECTION.....	10
FIGURE 10. EDIT A REO: CHANGING THE OPERATOR AND/OR THE TARGETVALUE FOR AN OBLIGATION	10
FIGURE 11. MAP OF A REO: FAILED OPERATION	11

1. Introduction

The Customization of Requirements functionality in MEDINA is provided by the *CNL Editor* tool. *CNL Editor* is a Web application, accessible via an Internet browser, which allows to view the Requirement and Obligations (REO) defined for Cloud Services and to make some changes to them.

A REO is an association between the Requirement, or Security Control, and a set of policies (or Obligations), where are specified the rules that must be applied to a Cloud Service *ResourceType* for the compliancy of Requirement itself.

A REO for a specific Cloud Service Id (CS_Id) is created by the *NL2CNL Translator*¹ with the aid of the *Catalogue of Controls and Metrics*² and Natural Language Processing (NLP) techniques. This process starts in the *Orchestrator*³, where the user can select a list of requirements to customize. The *Orchestrator* invokes the *NL2CNL Translator*, which aims to: associate a set of metrics to a requirement, translate metrics into obligations and store a requirement and its associated metrics into a REO object in the CNL Store.

A REO obligation is defined as follows:

ResourceType MUST MetricName TargetValueType (Operator, TargetValue)

As an example:

VirtualMachine MUST MalwareProtectionOutput Boolean (==, true)

Users can see and act on the REOs that are pertaining to those cloud services that are included in their Keycloak profile (filtering on Cloud Service Id).

A user can modify a REO by deleting Obligations, changing the operator, or updating the *TargetValue* of the metric to customize it for a specific Cloud Service Provider instead of using the default value (i.e., the value defined in the *Catalogue of Controls and Metrics*).

CNL Editor has a vocabulary containing an Ontology to control the selection of the operator and in some cases the definition of *TargetValues*. It also includes internal databases to store REO files and data.

1.1. User Roles and Permissions

Access to *CNL Editor* is managed by Keycloak⁴. Only authorized users can invoke the tool within the MEDINA IUI (Integrated User Interface). The visibility of the REO, and their eventual management, is allowed depending on Cloud Service Id, i.e., a user can show and manage a REO if its Keycloak profile contains the Cloud Service Id associated to that REO.

At the time of writing this manual, we are considering, as an enhancement to the *CNL Editor*, the implementation of the Role Authorization feature as specified in the following table:

¹ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D2.5 <https://doi.org/10.5281/zenodo.7927213>

² For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D2.2 <https://doi.org/10.5281/zenodo.7794478>

³ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D3.6 <https://doi.org/10.5281/zenodo.7927225>

⁴ <https://www.keycloak.org>

Role	UI Actions
IT Security Governance	Read
Security Analyst	Read
Domain Governance	Read
Product and Service Owner	Read
Product (Security) Engineer	Show/Edit/Complete/Map to CNL Editor UI
Chief Information Security Office (CISO)	Read
Customer	None
Auditor	Read

Thus, a user with “Read” role will only be able to “Show” REOs and a user with “Write” Role [Product (Security) Engineer] will be authorized to perform all available operations on REOs, described in this manual.

2. User Manual

2.1. Toolbar

CNL Editor includes a toolbar (see Figure 1), always accessible in the upper area. The *Help* button allows the user to get to the user manual.

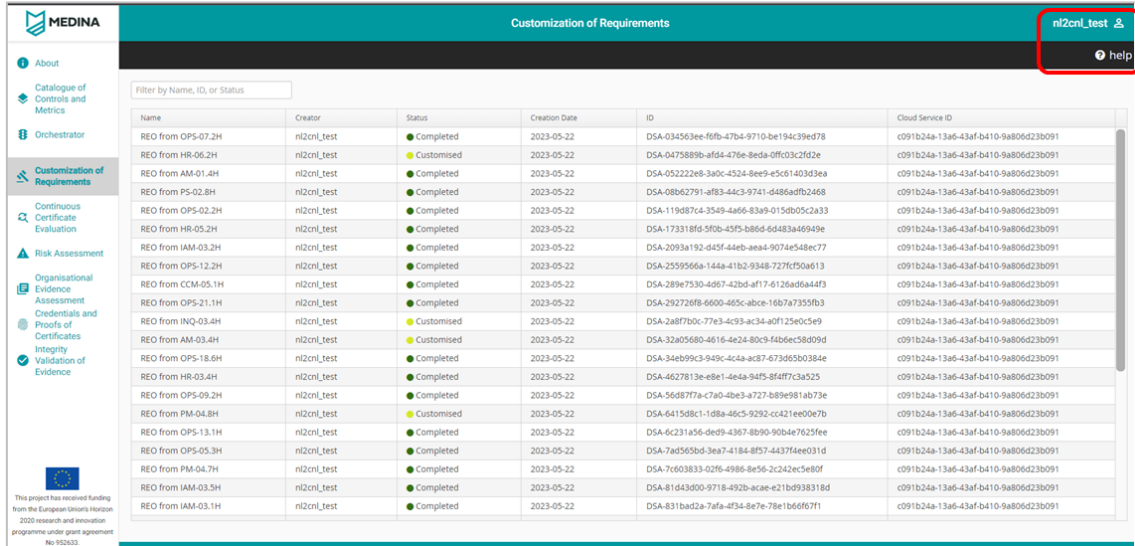


Figure 1. CNL Editor toolbar

2.2. List of REOs

The user invokes the *CNL Editor* in the MEDINA UI by selecting the “Customization of Requirements” left menu option. The web page in Figure 2 is displayed showing the list of all REOs filtered by Cloud Service Id (CS-Id).

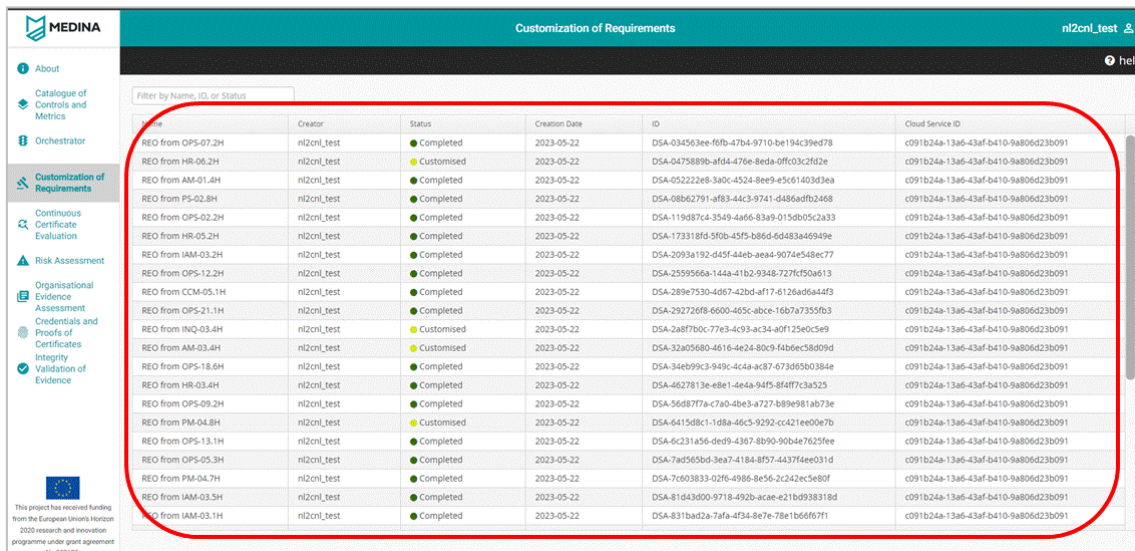


Figure 2. List of REOs

For each REO, the tool displays this metadata:

- **Name:** Description name of the REO.

- **Creator:** User that created the REO through the *Orchestrator*, with the help of *NL2CNL Translator* (to associate a set of metrics to a requirement, translate metrics into obligations and store REO).
- **Status:** Status of the REO (see Figure 3), which can have one of the following values:
 - *Customised*: the REO has been changed but has not been declared as completed by the user.
 - *Completed*: the REO was declared as completed by the user, so the *Map* operation (invocation to *DSL Mapper*⁵) can be performed on it to translate the REO into Rego code⁶.
 - *Available*: the REO was translated into Rego code after the execution of the *Map* operation.

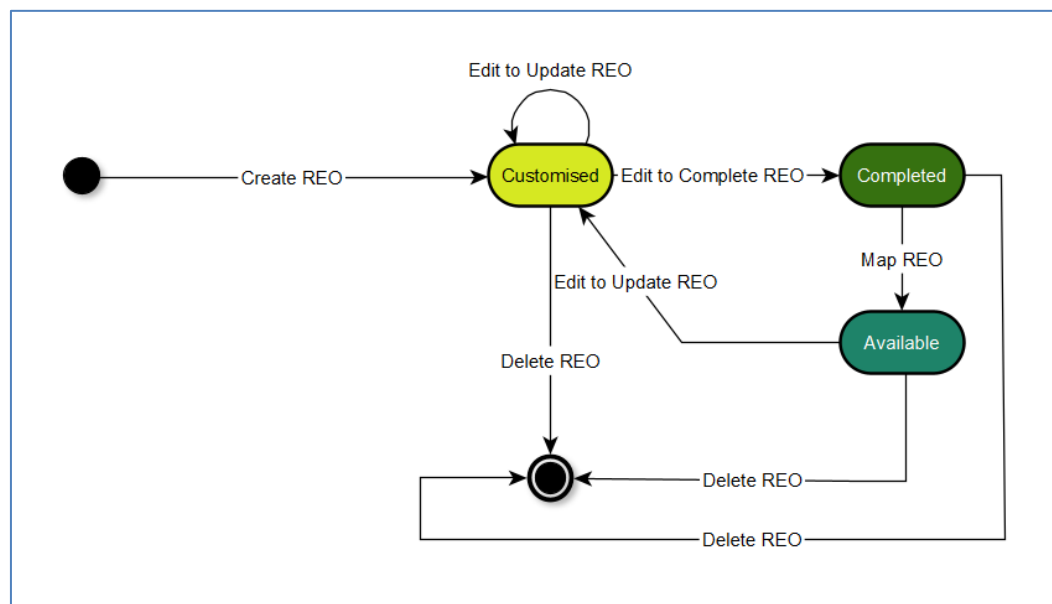


Figure 3. REO status diagram

- **Creation Date:** Date of creation of the REO.
- **ID:** Universal Unified ID of the REO.
- **Cloud Service ID:** Cloud Service Id for which the REO has been defined.

When the user selects a REO, a menu appears on the right-hand side of the web page showing the operations allowed on the REO according to its status (see Figure 4 and Figure 5). The available Operations are:

- **Show:** Visualizes the REO details.
- **Edit:** Allows editing of the REO (only available for “Customised” or “Available” REO status)
- **Map:** Invokes *DSL Mapper* that translates the Obligation into Rego code (only available for “Completed” REO status).
- **Copy:** Duplicates the REO.
- **Raw:** Shows, in .xml format, the file contained in the internal database with the REO data.
- **Delete:** Removes the REO from the internal database.

⁵ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D2.5 <https://doi.org/10.5281/zenodo.7927213>

⁶ <https://www.openpolicyagent.org/>

- **Cancel:** Makes the visualization come back to the List of REOs.

The screenshot shows the 'Customization of Requirements' interface. A table lists various REOs with columns: Name, Creator, Status, Creation Date, ID, and Cloud Service ID. The REO 'REO from INQ-03.4H' is selected. A context menu is open on the right, showing options: Show, Edit, Copy, Raw, Delete, and Cancel. The 'Delete' option is highlighted in red.

Figure 4. Operations available on a selected REO (status “Customised”)

The screenshot shows the 'Customization of Requirements' interface. A table lists various REOs with columns: Name, Creator, Status, Creation Date, ID, and Cloud Service ID. The REO 'REO from INQ-03.4H' is selected. A context menu is open on the right, showing options: Show, Map, Copy, Raw, Delete, and Cancel. The 'Map' option is highlighted in red.

Figure 5. Operations available on a selected REO (status “Completed”)

The following explains the main operations on REOs: *Show*, *Edit* and *Map*.

2.2.1. Show operation

The *Show* operation allows the user to view the details of the REO on a web page consisting of two sections: “Metadata” and “Obligations”.

The “Metadata” section shows information about the REO (see Figure 6), such as vocabulary version, Requirement, Security Control, Cloud Service, Framework (e.g., EUCS), Metric type (Technical or Organizational), and Assurance level (High, Medium, Low).

Figure 6. Show the details of a REO: Metadata section

The “Obligations” section shows all policies defined for the requirement, indicating their origin (*catalogue*⁷ or *recommender*⁸) on the right-hand side. Policies follow the statement format described in section 1.

Policies	Metric ID / Source	
PolicyDocument MUST AssetMonitoringQ1 na(na,na)	AssetMonitoringQ1 / catalogue	?
VirtualMachine MUST MalwareProtectionOutput Boolean(==,true)	MalwareProtectionOutput / recommender	?
Application MUST SecureCryptographicPrimitives Boolean(==,true)	SecureCryptographicPrimitives / recommender	?
PolicyDocument MUST AssetManagementPolicy02 String([sln,[procurement, destruction, none]])	AssetManagementPolicy02 / recommender	?
PolicyDocument MUST AssetManagementPolicy03 String([sln,[services, IP addresses, databases, VM, application, service plan instances, database instances]])	AssetManagementPolicy03 / recommender	?
PolicyDocument MUST ChangeManagementPolicy01 na(na,na)	ChangeManagementPolicy01 / recommender	?
PolicyDocument MUST DataRestoreTestFrequencyQ1 Float(<=,100)	DataRestoreTestFrequencyQ1 / recommender	?
PolicyDocument MUST PatchManagementPolicyCheckQ1 String([sln,[every, none, partial]])	PatchManagementPolicyCheckQ1 / recommender	?
PolicyDocument MUST SystemHardeningPolicyQ1 na(na,na)	SystemHardeningPolicyQ1 / recommender	?
PolicyDocument MUST EventLogMonitoringQ1 na(na,na)	EventLogMonitoringQ1 / recommender	?
PolicyDocument MUST PolicyUpToDateCheck Integer(<=,365)	PolicyUpToDateCheck / recommender	?

Figure 7. Show the details of a REO: Obligation section

Finally, the *Back* button allows the user to come back to the REO list.

2.2.2. Edit operation

When the user selects the *Edit* operation for a REO, the tool shows a web page with all REO data and the possibility to make changes to it (see Figure 4). The *Edit* operation page, as seen above for the *Show* operation page, also includes two sections: “Metadata” and “Obligations” (see Figure 8 and Figure 9).

⁷ ‘catalogue’ indicates that the metric comes from the *Catalogue of Controls and Metrics* tool.

⁸ ‘recommender’ refers that the metric comes from by the Metric recommender, which is a component of the *NL2CNL Translator* tool.

Update Complete Back

Title REO from AM-01.4H

Status CUSTOMISED

Date 2022-04-27 17:42:00

Cloud Service ID 937210b1-a9f2-4929-bbbc-5a7ecc0f089f

Additional Information

UUID DSA-5dfcbe9d-694a-4f70-90ec-8f568eb46d18.xml

Vocabulary URI https://cni-vocabulary-test.k8s.medina.es/lab.org/vocabularies/medina_vocabulary_test_v2.0.owl#

Requirement

Requirement Code AM-01.4H

Security Control AM-01

Framework EUCS

Type ORGANIZATIONAL

Description The CSP shall automatically monitor the process performing the inventory of assets to guarantee it is up-to-date.

Assurance level HIGH

Obligations

Figure 8. Edit a REO: Metadata section

Obligations

PolicyDocument	MUST	AssetMonitoringQ1	AssetMonitoringQ1
VirtualMachine	MUST	MalwareProtectionOutput	Boolean
Application	MUST	SecureCryptographicPrimitives	Boolean
PolicyDocument	MUST	AssetManagementPolicy02	String
PolicyDocument	MUST	AssetManagementPolicy03	String
PolicyDocument	MUST	ChangeManagementPolicy01	na
PolicyDocument	MUST	DataRestoreTestFrequencyQ1	Float
PolicyDocument	MUST	PatchManagementPolicyCheckQ1	String

Complete the policy

Add additional info for na#1 Add Info

Complete the policy

Add additional info for Boolean#1 Add Info

Complete the policy

Add additional info for Boolean#1 Add Info

Complete the policy

Add additional info for String#1 Add Info

Complete the policy

Add additional info for String#1 Add Info

Complete the policy

Add additional info for na#1 Add Info

Complete the policy

Add additional info for Float#1 Add Info

Complete the policy

Add additional info for String#1 Add Info

Figure 9. Edit a REO: Obligations section

The user can perform two actions on an Obligation (see Figure 10):

- delete the Obligation by clicking on the button
- change the Obligation parameters by clicking on the *Add info* button

Complete the policy

Add additional info for na#1 Add Info

Complete the policy

Add additional info for String#1 Add Info

Complete the policy

Add additional info for na#1 Add Info

Complete the policy

Add additional info for String#1 Add Info

Complete the policy

Add additional info for String#1 Add Info

Complete the policy

Add additional info for Integer#1 Add Info

Add additional info

String Param < v

String Option immediately

OK Cancel

Figure 10. Edit a REO: changing the Operator and/or the TargetValue for an Obligation

where *Param* is the Operator, and *Option* is the TargetValue.

Finally, the user can select one of the following buttons (see Figure 8):

- **Update:** Confirms changes and maintains the REO in “Customised” status.
- **Complete:** Confirms changes and pass the REO status from “Customised” to “Completed”.
- **Back:** Cancels the changes and exits from the *Edit* operation, so the visualization comes back to the List of REOs.

2.2.3. Map operation

With the selection of the *Map* operation, which is only available for REOs in the “Completed” status, the policies contained in the REO are translated into Rego code and can be used by the *Orchestrator*⁹.

If the *Map* operation is completed with success, the REO goes from “Completed” to “Available” status, otherwise the message “Failed to Map” is displayed for a few seconds in the lower right corner of the page (see Figure 11).

Customization of Requirements						nl2cnl_test
Filter by Name, ID, or Status						
Name	Creator	Status	Creation Date	ID	Cloud Service ID	
REO from OPS-07.2H	nl2cnl_test	Completed	2023-05-22	DSA-034563ee-f6fb-47b4-9710-be194c39ed78	c091b24a-13a6-43af-b410-9a806d23b091	
REO from HR-06.2H	nl2cnl_test	Customised	2023-05-22	DSA-0475889b-afd4-476e-8eda-0ffc03c2fd2e	c091b24a-13a6-43af-b410-9a806d23b091	
REO from AM-01.4H	nl2cnl_test	Completed	2023-05-22	DSA-052222e8-3a0c-4524-8ee9-e5c61403d3ea	c091b24a-13a6-43af-b410-9a806d23b091	
REO from PS-02.8H	nl2cnl_test	Completed	2023-05-22	DSA-08b62791-af83-44c3-9741-d486adfb2468	c091b24a-13a6-43af-b410-9a806d23b091	
REO from OPS-02.2H	nl2cnl_test	Completed	2023-05-22	DSA-119d87c4-3549-4a66-83a9-015d005c2a33	c091b24a-13a6-43af-b410-9a806d23b091	
REO from HR-05.2H	nl2cnl_test	Completed	2023-05-22	DSA-173318fd-5f0b-45f5-b86d-6d483a46949e	c091b24a-13a6-43af-b410-9a806d23b091	
REO from IAM-03.2H	nl2cnl_test	Completed	2023-05-22	DSA-2093a192-d45f-44eb-aea4-9074e548ec77	c091b24a-13a6-43af-b410-9a806d23b091	
REO from OPS-12.2H	nl2cnl_test	Completed	2023-05-22	DSA-2559566a-144a-41b2-9348-727fcf50a513	c091b24a-13a6-43af-b410-9a806d23b091	
REO from CCM-05.1H	nl2cnl_test	Completed	2023-05-22	DSA-289e7530-4d67-42bd-af17-6126ad6a44f3	c091b24a-13a6-43af-b410-9a806d23b091	
REO from OPS-21.1H	nl2cnl_test	Completed	2023-05-22	DSA-292726f8-6600-465c-abce-16b7a7355fb3	c091b24a-13a6-43af-b410-9a806d23b091	
REO from INQ-03.4H	nl2cnl_test	Customised	2023-05-22	DSA-2a8f7b0c-77e3-4c93-ac34-a0f125e0c5e9	c091b24a-13a6-43af-b410-9a806d23b091	
REO from AM-03.4H	nl2cnl_test	Customised	2023-05-22	DSA-32a05680-4616-4e24-80c9-44b6ec58d09d	c091b24a-13a6-43af-b410-9a806d23b091	
REO from OPS-18.6H	nl2cnl_test	Completed	2023-05-22	DSA-34eb99c3-949c-4c4a-ac87-673d65b0384e	c091b24a-13a6-43af-b410-9a806d23b091	
REO from HR-03.4H	nl2cnl_test	Completed	2023-05-22	DSA-4627813e-e8e1-4e4a-94f5-8f4f7c3a525	c091b24a-13a6-43af-b410-9a806d23b091	
REO from OPS-09.2H	nl2cnl_test	Completed	2023-05-22	DSA-56d8777a-c7a0-4be3-4727-b89e981ab73e	c091b24a-13a6-43af-b410-9a806d23b091	
REO from PM-04.8H	nl2cnl_test	Customised	2023-05-22	DSA-6415d8c1-1d8a-46c5-9292-cc421ee00e7b	c091b24a-13a6-43af-b410-9a806d23b091	
REO from OPS-13.1H	nl2cnl_test	Completed	2023-05-22	DSA-6c231a56-de09-4367-8b90-90b4e7625fee	c091b24a-13a6-43af-b410-9a806d23b091	
REO from OPS-05.3H	nl2cnl_test	Completed	2023-05-22	DSA-7ad565bd-3ea7-4184-8f57-4437f4ee031d	c091b24a-13a6-43af-b410-9a806d23b091	
REO from PM-04.7H	nl2cnl_test	Completed	2023-05-22	DSA-7c603833-02f6-4986-8e56-2c242ec5e80f	c091b24a-13a6-43af-b410-9a806d23b091	
REO from IAM-03.5H	nl2cnl_test	Completed	2023-05-22	DSA-81d43d00-9718-492b-acae-e21bd938318d	c091b24a-13a6-43af-b410-9a806d23b091	
REO from IAM-03.1H	nl2cnl_test	Completed	2023-05-22	DSA-831bad2a-7afa-4f34-8e7e-78e1b66f67f1	c091b24a-13a6-43af-b410-9a806d23b091	

Hewlett Packard Enterprise

THE WEBSITE AND ITS SERVICES ARE IN BETA AND ARE PROVIDED FOR RESEARCH PURPOSES, EXPERIMENTATION AND SCIENTIFIC PUBLICATION. COPYRIGHT © 2023 HEWLETT PACKARD ENTERPRISE DEVELOPMENT COMPANY, L.P. ALL RIGHTS RESERVED.

Failed to Map

Figure 11. Map of a REO: failed operation

⁹ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D3.6 <https://doi.org/10.5281/zenodo.7927225>

3. Delivery and usage

3.1. Licensing information

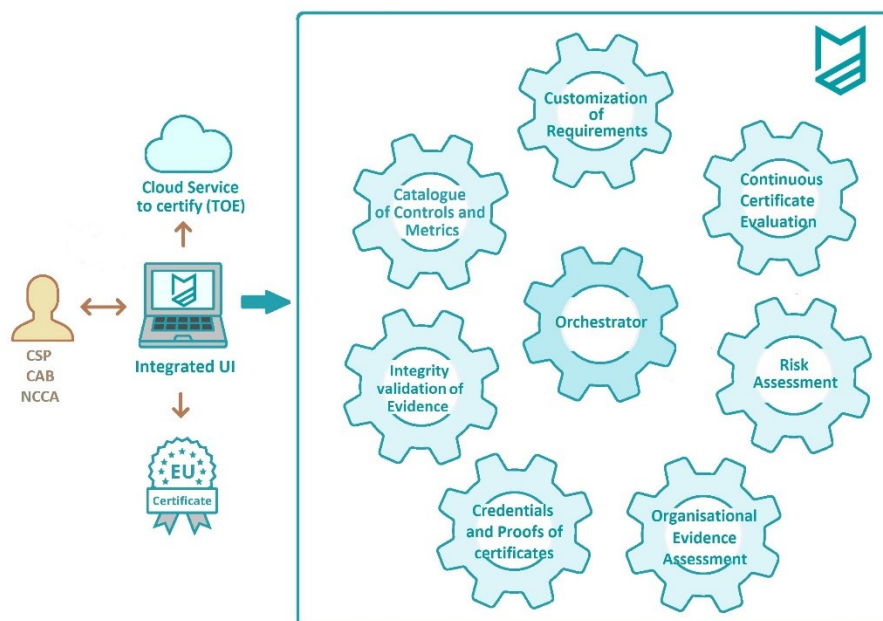
The *CNL Editor* is offered under the Apache 2.0 license and is a proprietary tool of HPE.

3.2. More information

The MEDINA web site (<https://medina-project.eu/>) also includes several deliverables and blog posts related to the *CNL Editor*.

Risk Assessment

- User Manual -



Project Title:	MEDINA - Security Framework to achieve a continuous audit-based certification in compliance with the EU-wide cloud security certification scheme
Project Number:	952633
Editor:	Artsiom Yautsiukhin (Centro Nazionale della Ricerca)
Version:	v1.0
Date:	31.07.2023
Distribution level:	PU



Table of contents

1. Introduction	4
1.1. User Roles and Permissions	4
2. User Manual	6
2.1. Toolbar	6
2.2. Select ToE	6
2.2.1. Static Risk Assessment.....	7
2.2.2. Risk Optimisation.....	12
2.2.3. Dynamic Risk Assessment.....	13
2.2.4. API.....	14
3. Delivery and usage	17
3.1. Licensing information.....	17
3.2. Download	17
3.3. More information.....	17

List of Figures

FIGURE 1. SATRA TOOLBAR	6
FIGURE 2. SELECTION OF TOES.....	6
FIGURE 3. PROVIDE INFORMATION ABOUT THE TOE	7
FIGURE 4. SELECT STATIC OR DYNAMIC RISK ASSESSMENT	7
FIGURE 5. QUESTIONNAIRE	8
FIGURE 6. QUESTIONNAIRE. BOTTOM	9
FIGURE 7. ASSETS COLLECTION.....	9
FIGURE 8. COMPLIANCE RADAR CHART	11
FIGURE 9. OVERALL RISK AND NON-CONFORMANCE VALUES	11
FIGURE 10. RISK PER THREATS	12
FIGURE 11. RISK OPTIMISATION. INPUT	13
FIGURE 12. PREPARATION FOR DYNAMIC RISK ASSESSMENT PROCESS. ASSET INFORMATION COLLECTION	14
FIGURE 13. SATRA/RAOF APIS	16

1. Introduction

Cyber security risk assessment is a high-level instrument to evaluate the cyber security of a system. It serves as a glue between the management and technical levels helping to analyse the current system state and abstract the results for the further strategic decision making. The main advantage of applying risk assessment is the focus on the concrete needs of the system owner.

In scope of MEDINA, risk assessment serves for the analysis of requirements demanded by a certification scheme and ensuring that fulfilment of these requirements is indeed relevant for the cloud service provider (CSP). Naturally, if a CSP satisfies all requirements it completely complies with the certification scheme and should obtain or maintain the certificate. But, in many real cases some requirements may be insignificant for a CSP (e.g., because they focus on protection of an asset which is not sensitive for this CSP). Such non-conformities should be evaluated, and risk assessment is used for such analysis. The analysis should tell if the detected non-conformities are major ones and the certificate should be revoked, or the deviation is minor and the certificate should be maintained (probably, under some conditions).

Risk assessment functionality is implemented with a Self-Assessment Tool for Risk Analysis (SATRA), which realises a Risk Assessment and Optimisation Framework (RAOF) component. This tool provides the following capabilities:

- Simple and fast EUCS¹-based risk assessment. The risk assessment requires filling in a questionnaire made with EUCS requirements and an asset table. The risk computation itself is performed automatically, taking into account the cloud service level in order to define the most relevant threats to analyse.
- (EUCS-based) Non-conformities assessment. The non-conformities assessment compares risks of full compliance with the risk of reported compliance with EUCS. This analysis of compliance allows to take into account the needs of the CSP.
- Optimisation support. This is an auxiliary functionality which aims to help a CSP to plan his investments in compliance with EUCS and obtain the most optimal system configuration.
- Dynamic and objective risk-based analysis of non-conformities. In contrast to the static risks assessment mentioned above, the information about the compliance with EUCS for dynamic risk assessment is collected by various assessment tools, instead of a human operator. Moreover, this information is provided and analysed in a detailed way, e.g., risk for every asset is computed separately.
- Failure prioritisation support. During the dynamic risk computation, the detected failures are estimated with respect to their contribution to the overall risk picture. These evaluations aim to help the CSP in prioritising the remediation effort.

1.1. User Roles and Permissions

Access to SATRA is managed by Keycloak². The following actions are allowed for the defined roles in the scope of the MEDINA framework.

Role	Allowed Actions
IT Security Governance	Risk Computation (Reporting)
Security Analyst	Risk Computation (Reporting)
Domain Governance	Risk Computation (Reporting)

¹ EUCS (European Cybersecurity Certification Scheme for Cloud Services)

<https://www.enisa.europa.eu/publications/eucs-cloud-service-scheme>

² <https://www.keycloak.org>

Role	Allowed Actions
Product and Service Owner	Create ToE ³ , ToE Info, Questionnaire, Asset Information, Risk Computation (Reporting)
Product (Security) Engineer	Risk Computation (Reporting)
Chief Information Security Office (CISO)	Risk Computation (Reporting)
Customer	None
Auditor	Risk Computation (Reporting)

³ ToE: Target of Evaluation

2. User Manual

2.1. Toolbar

There are only two buttons, named *Select ToE* and *Help*, in the toolbar of the SATRA tool (see Figure 1). The *Select ToE* button leads the user to the initial page of the tool, where he/she can select the ToE (Target of Evaluation) to analyse. The *Help* button opens this manual.



Figure 1. SATRA toolbar

2.2. Select ToE

First, the user should select the ToE to analyse (see Figure 2) by clicking the *Go* button next to the required ToE.

Targets of Evaluation (ToE) available for analysis

Select a Target of Evaluation for the risk-based analysis.

ToE ID	ToE Name	
0e37d39c-d3ba-4a24-aeaa-380c41cde64c	TestArt2	Go
1bca421e-c708-11ed-afa1-0242ac120002	First ToE	Go
1e67df1a-2127-421a-ba0f-c8731b5e5d3c	TEST STEFANO	Go
2a9c09a7-ebb7-4a97-835c-ea436c1b38b1		Go
2fea17b3-1298-4f8a-af6d-f80e355438d4		Go
34175106-a188-4c7f-9720-53dc7eaa490	First ToE	Go
4b60d24a-c6f4-11ed-afa1-0242ac120002	TEST ToE	Go
5b51b1d2-bb00-4512-be37-24819b5d99ab	TestHigh	Go
600e0e76-df6b-11ed-b5ea-0242ac120002	ToE TEST CCE	Go
8cd8c7d0-1446-4cac-ab96-c3f82cd91ab2		Go
90acc728-dfd0-41a7-acd6-0b86500f4568		Go

Figure 2. Selection of ToEs

Next, it is required to set up the Cloud Service Layer of the service, the certification scheme, and the assurance level to comply with (see Figure 3). In the scope of the MEDINA project, only the EUCS certification scheme is supported.

Figure 3. Provide Information about the ToE

After that the user is prompted to indicate whether he/she is interested in conducting static risk assessment or in preparing for the dynamic risk assessment (see Figure 4). The static risk assessment is performed by a user before certification (see section 2.2.1) , while the dynamic risk assessment is an integral part of the certificate monitoring process (see section 2.2.3).

The static risk assessment requires two types of information to be provided: information about compliance with the EUCS and main assets. The preparatory step for the dynamic risk assessment prompts only for the sensitivity parameters of the assets (since the information about the current status of the compliance of the service is monitored by various assessment tools without human intervention).

Figure 4. Select Static or Dynamic Risk Assessment

2.2.1. Static Risk Assessment

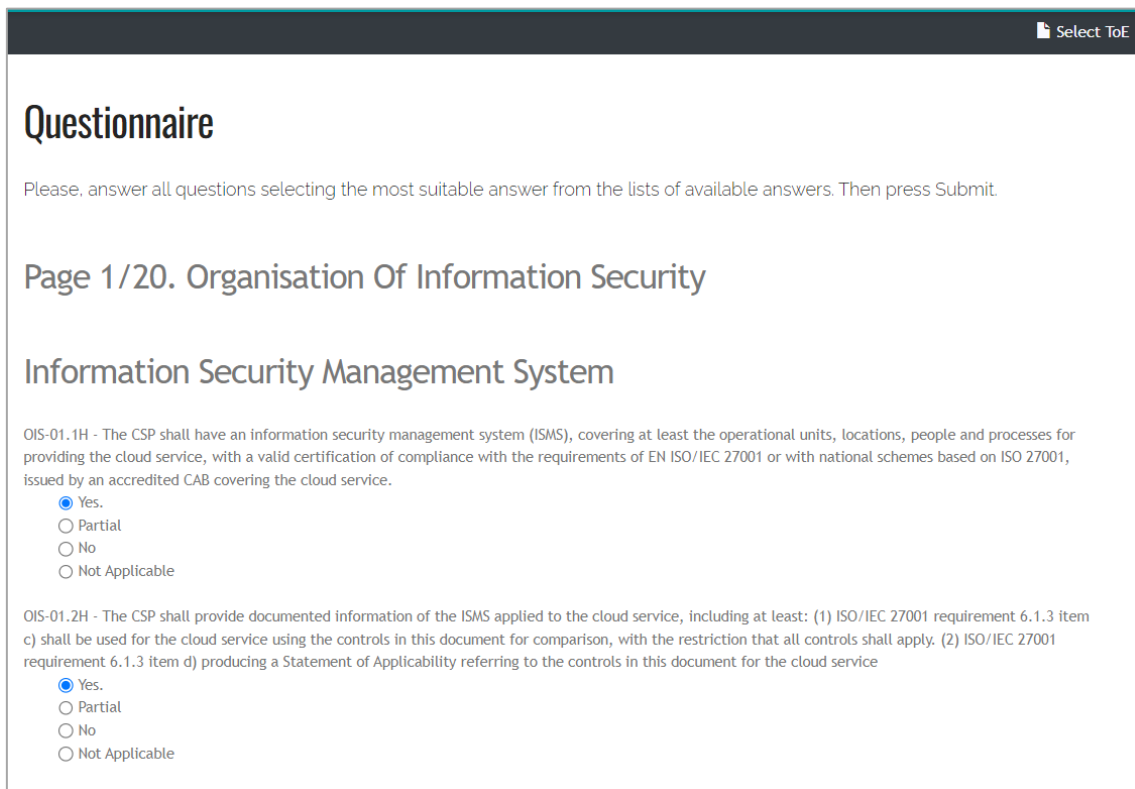
During the first step of the static risk assessment (see Figure 5), the user is asked to answer whether the corresponding EUCS requirement is:

- Fully implemented (yes)
- Partly implemented (partial)

- Not implemented (no)
- Not applicable (N/A)

The “partial” option is added for compatibility with the *Catalogue of Controls and Metrics*⁴ (a.k.a. *Catalogue*), as well as for a better representation of the status of compliance. Yet, this option is treated by the risk analysis tool (as it would be treated by the compliance verification process) as “not implemented”. The “Not applicable” option should be used with care, assuming that the CSP has enough evidence to convince the auditor that this requirement is, indeed, not applicable for the CSP.

It is worth noting that the user has the possibility to import his/her answers from the questionnaire of the *Catalogue* (by using the corresponding option). Note, that it is important to set up the same parameters (in particular, the same assurance level used in the *Catalogue*) before starting the risk assessment process. After importing the answers, the user should find the risk assessment questionnaire already prefilled (as much as it was done for the corresponding questions in the *Catalogue*).



The screenshot shows a web interface for a questionnaire. At the top right, there is a button labeled "Select ToE". The main heading is "Questionnaire". Below it, a instruction says: "Please, answer all questions selecting the most suitable answer from the lists of available answers. Then press Submit." The current page is "Page 1 / 20. Organisation Of Information Security". The specific section is "Information Security Management System".

Question 1 (OIS-01.1H): "The CSP shall have an information security management system (ISMS), covering at least the operational units, locations, people and processes for providing the cloud service, with a valid certification of compliance with the requirements of EN ISO/IEC 27001 or with national schemes based on ISO 27001, issued by an accredited CAB covering the cloud service." The options are: ☒ Yes, ☐ Partial, ☐ No, ☐ Not Applicable.

Question 2 (OIS-01.2H): "The CSP shall provide documented information of the ISMS applied to the cloud service, including at least: (1) ISO/IEC 27001 requirement 6.1.3 item c) shall be used for the cloud service using the controls in this document for comparison, with the restriction that all controls shall apply. (2) ISO/IEC 27001 requirement 6.1.3 item d) producing a Statement of Applicability referring to the controls in this document for the cloud service". The options are: ☒ Yes, ☐ Partial, ☐ No, ☐ Not Applicable.

Figure 5. Questionnaire

Requirements are organised according to the EUCS structure, i.e., grouped by controls, and controls are grouped by categories. Each of the twenty pages contains questions for a specific category defined by EUCS. Once the user finishes answering questions for a page, he/she should *Go Forward* (see Figure 6). The user can take a break pressing *Save and Leave* to return to the questionnaire later.

⁴ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D2.2 <https://zenodo.org/record/7794478>

Contact With Authorities And Interest Groups

OIS-03.1H - The CSP shall maintain regular contacts with relevant authorities in terms of information security and relevant technical groups to stay informed about current threats and vulnerabilities.

☒ Yes.
☐ Partial
☐ No
☐ Not Applicable

Information Security In Project Management

OIS-04.1H - The CSP shall perform a risk assessment according to RM-01 to assess and treat the risks on all projects that may affect the provision of the cloud service, regardless of the nature of the project.

☒ Yes.
☐ Partial
☐ No
☐ Not Applicable

Figure 6. Questionnaire. Bottom

All questions must be answered before the user can move to the second part of the input provisioning process, i.e., the description of the assets. Yet, if all questions are already answered, there is no need to pass through all pages. In this case, it is enough to press the *Go to Cloud Resource* button in Figure 6. The user should press this button at the end of the questionnaire to pass to the next step of the input provisioning process.

At the second step, the user is prompted to provide a list of Cloud Resources and assess their sensitivity (see Figure 7).

CLOUD RESOURCE IDENTIFICATION

ID	Cloud Resource	Cloud Resource Type	Number Of Unit	Confidentiality Level	Integrity Level	Availability Level
A1	<input type="text" value="Insel"/>	IoT Device Provisioning Service	<input type="text" value="1"/>	<input type="text" value="6"/>	<input type="text" value="3"/>	<input type="text" value="3"/>
A2	<input type="text" value="Insel"/>	CI CD Service	<input type="text" value="1"/>	<input type="text" value="6"/>	<input type="text" value="6"/>	<input type="text" value="3"/>
A3	<input type="text" value="Insel"/>	Function	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="4"/>	<input type="text" value="3"/>
A4	<input type="text" value="Insert"/>	Database	<input type="text" value="1"/>	<input type="text" value="7"/>	<input type="text" value="3"/>	<input type="text" value="3"/>
A5	<input type="text" value="Insert"/>	Virtual Machine	<input type="text" value="1"/>	<input type="text" value="2"/>	<input type="text" value="3"/>	<input type="text" value="5"/>
A6	<input type="text" value="Insert"/>	Client trust	<input type="text" value="1"/>	<input type="text" value="7"/>	<input type="text" value="3"/>	<input type="text" value="5"/>

Figure 7. Assets collection

Each resource should be one of the 13 supported “Cloud Resource Types”:

- 1 CI CD Service
- 2 Container
- 3 Container Image
- 4 ContainerOrchestration
- 5 ContainerRegistry

- 6 Database
- 7 Function
- 8 IoT Device Provisioning Service
- 9 IoT Messaging Hub
- 10 Local storage
- 11 Network
- 12 Virtual Machine
- 13 VM Image

Once more cloud resource type is obligatory: “Client trust”. It represents how the failure to maintain confidentiality, integrity or availability of the service impacts the future attitude of clients to the cloud service. One resource of this type is added to the list of resources by default. It cannot be removed, and no additional resources of the same type can be added.

In Figure 7, “Number of units” represents the approximate amount of resources of a certain type which belong to the service.

Also, in Figure 7, “Confidentiality, Integrity and Availability levels” indicate the approximate level of losses due to a loss of Confidentiality, Integrity and Availability of the considered resource (e.g., stolen data from a database, or unavailable virtual machine). The rules of thumbs for the levels are as follows:

- 1 Not important at all
- 2 Cause only small inconvenience (e.g., require reboot)
- 3 Systematic malfunctioning with no further consequences
- 4 Some sensitive data is lost
- 5 A portion of sensitive data is lost
- 6 Unnoticed financial abuse
- 7 Failure to function/may stop your business/large amount of data is lost (10,000 affected and more)
- 8 May cause injury/get out of business
- 9 Causing a life loss or a huge amount of sensitive data stolen (10,000,000 affected)
- 10 Catastrophic consequence with several/many lives lost.

New cloud resources can be added to the list, by clicking the *Create row* button, or deleted from the list, by clicking the *Delete row* button (requires indication of the row number).

When the setting of resources is finished, it is required to *Submit* the input. Note, that the tool will ask you to press the *Submit* button twice. After that, the tool computes the risk and analyses the detected non-conformity gap. As a result, the following information is displayed.

First, the radar chart with compliance levels per EUCS category is displayed (see Figure 8).

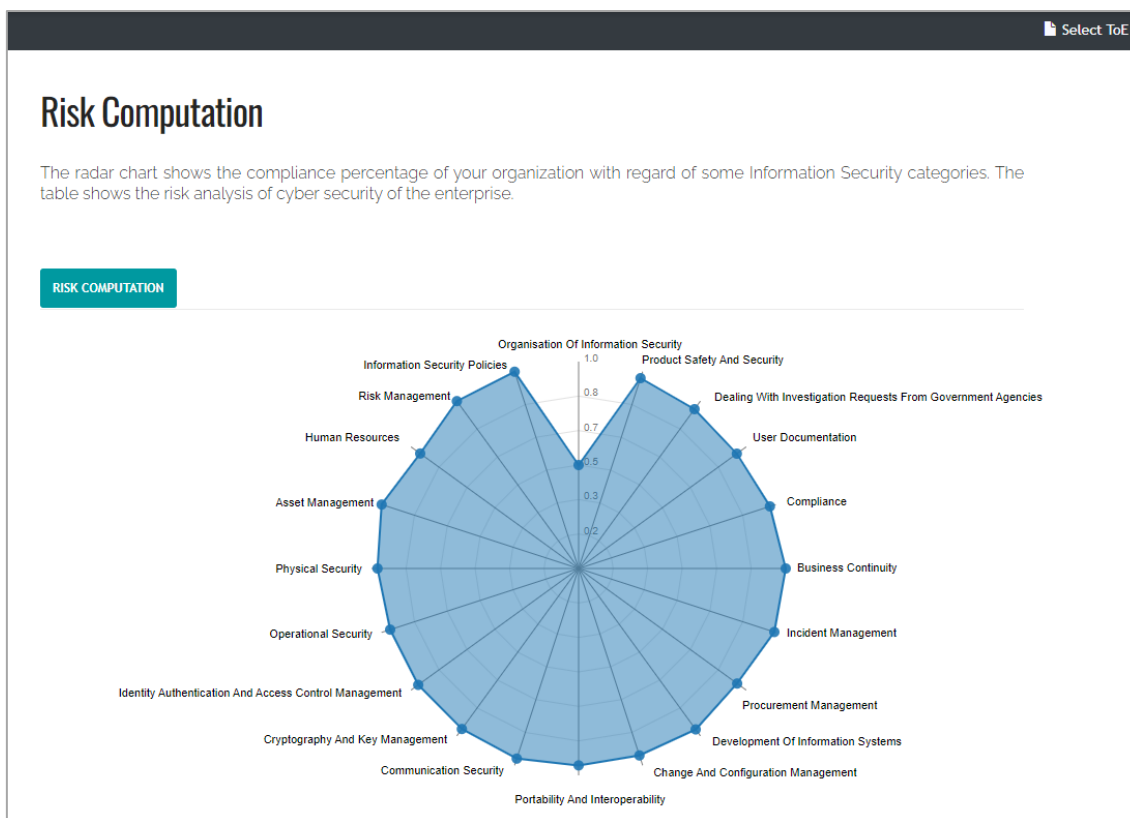


Figure 8. Compliance radar chart

Second, the overall information about the computed cyber risk for the service is displayed (see Figure 9):

- “Overall Risk” for the service
- “Best” risk level, which can be obtained by implementing all requirements
- Non-conformance gap, which is the difference between the best and overall risk values
- Non-conformance assessment result, which compares the non-conformance gap with a threshold. In case the threshold is exceeded by the non-conformance gap value, the deviation is considered Major (and Minor, otherwise).

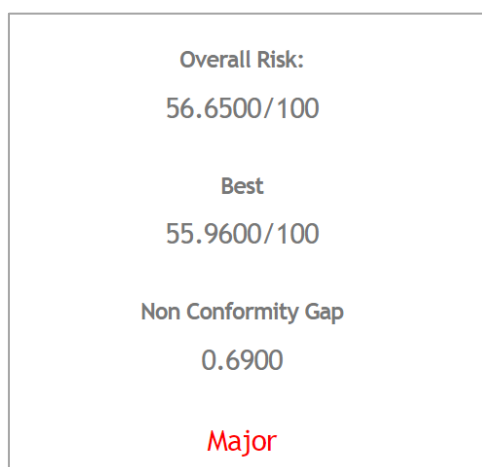


Figure 9. Overall risk and non-conformance values

Finally, the risk levels per specific threats are displayed (see Figure 10).

Threat Title	Risk
Third party problems	41.02
Account hijacking (client)	34.14
Meta- interfaces (client)	28.42
CI/CD attacks	0.0
Account hijacking (CSP)	40.19
web-application threat (API, GUI, service vulnerabilities)	0.0
Environment threat (DC)	16.44
Exhaustion of resources (client)	25.97
On-site tampering/penetration	34.75
Data location failure	45.69
Web-based attack	28.28
Unnecessary disclosure to law enforcement	33.09
DoS (client)	18.72
System glitch	42.56
Malicious client employee	24.97
Malicious client	28.31
Exploit insecure IAM (client)	36.04
Lack of support for compositional certification	51.42
Exploit misconfiguration of cloud services (CSP)	34.24
Compromised Communication	24.94
Poor IAM(client)	29.6
Hardware theft/loss (DC)	24.73
Physical threat (DC)	4.69
Exploit misconfiguration of cloud services (client)	38.57
Exploit insecure IAM (CSP)	33.48
Unlawful client	39.35
CSP's employee Negligence and mistakes	40.94
Insider hacker	29.64
Meta- interfaces (CSP)	35.95

Figure 10. Risk per threats

2.2.2. Risk Optimisation

After conducting the static risk assessment, an operator may use the Risk Optimisation mechanism. This mechanism allows an operator to analyse failed requirements and select those of them which may decrease the risk level, taking into account the budget limit for such changes.

In order to use this functionality, an operator should press the *Suggest Optimisation* button at the bottom of the risk results page. The functionality requires two types of input (see Figure 11):

- Budget limit (in euro) available for fixing failed requirements
- Cost of implementing each of the failed requirements

The Budget limit amount must be lower than the sum of all failed requirements, otherwise the selection will be trivial: all failed requirements will be selected. Another extreme is when the Budget limit is lower than the cheapest requirement; the result will be trivial as well: no requirements are to be selected. Note that SATRA counts all requirements marked as *partial*, as failed (since EUCS does not allow partial fulfilment of a requirement).

Risk Mitigation

Providing a budget, the tool will suggest security activities to invest and the optimal expense amount to minimize the overall risk calculated.

Overall Risk:

3.03€

Put the budget limit

Get Risk

€ Back to Risk Computation

Questions	Cost
PS-01.1B - The CSP shall define security perimeters in the buildings and premises related to the cloud service provided.	1000
OPS-01.1B - The CSP shall define and implement procedures to plan for capacities and resources (personnel and IT resources), which shall include forecasting future capacity requirements in order to identify usage trends and manage system overload.	1000
OPS-18.3B - The CSP shall publish and maintain a publicly and easily accessible online register of vulnerabilities that affect the cloud service and assets provided by the CSP that the CSCs have to install, provide or operate under their own responsibility.	1000
OPS-22.1B - The CSP shall segregate from other CSCs the data stored and processed on shared virtual and physical resources on behalf of a CSC to ensure the confidentiality and integrity of this data.	1000
IAM-01.1B - The CSP shall define role and rights policies and procedures for controlling access to information resources, according to ISP-02 and based on the business and security requirements of the CSP, in which at least the following aspects are covered: (1) Parameters to be considered for making access control decisions; (2) Granting and modifying access rights based on the "least-privilege" principle and on the "need-to-know" principle; (3) Segregation of duties between managing, approving and assigning access rights; (4) Dedicated rules for users with privileged access; (5) Requirements for the approval and documentation of the management of access rights.	9
CKM-01.1B - The CSP shall define and implement policies with technical and organizational safeguards for cryptography and key	

Figure 11. Risk optimisation. Input

Once the input is provided and the procedure is started, the tool will run the optimization procedure and select those requirements whose implementation sum is lower than the target budget and which maximally reduce the risk value. Moreover, the tool computes the target non-conformity gap, assuming that the selected requirements are fulfilled, and reports if the remaining non-conformity is major or minor. Having this information, the CSP may plan its future steps in improving its system and prepare for certification.

2.2.3. Dynamic Risk Assessment

In order to conduct the Dynamic Risk Assessment, the tool needs to be pre-configured. In particular, the information about the sensitivity of the assets should be provided to the system.

After clicking on the button *Prepare for Dynamic Risk Assessment* (see Figure 4), the user will be directed to the page shown in Figure 12, which is very similar to the one for the collection of the information about assets (see Figure 7). Yet, this page is slightly different from its static counterpart.

CLOUD RESOURCE IDENTIFICATION

ID	Cloud Resource	Cloud Resource Type	Confidentiality Level	Integrity Level	Availability Level	Cloud Resource ID
A1	<input type="text" value="Insert"/>	CI CD Service	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	
A2	<input type="text" value="Insert"/>	Container	<input type="text" value="6"/>	<input type="text" value="6"/>	<input type="text" value="1"/>	
A3	<input type="text" value="Insert"/>	Function	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="5"/>	
A4	<input type="text" value="Insert"/>	Virtual Machine	<input type="text" value="1"/>	<input type="text" value="5"/>	<input type="text" value="6"/>	
A5	<input type="text" value="Insert"/>	ContainerOrchestration	<input type="text" value="7"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	
A6	<input type="text" value="Insert"/>	ContainerRegistry	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	
A7	<input type="text" value="Insert"/>	Database	<input type="text" value="5"/>	<input type="text" value="5"/>	<input type="text" value="1"/>	
A8	<input type="text" value="Insert"/>	Container Image	<input type="text" value="6"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	
A9	<input type="text" value="Insert"/>	VM Image	<input type="text" value="6"/>	<input type="text" value="6"/>	<input type="text" value="4"/>	
A10	<input type="text" value="Insert"/>	IoT Device Provisioning Service	<input type="text" value="1"/>	<input type="text" value="5"/>	<input type="text" value="1"/>	
A11	<input type="text" value="Insert"/>	IoT Messaging Hub	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="7"/>	
A12	<input type="text" value="Insert"/>	Network	<input type="text" value="5"/>	<input type="text" value="3"/>	<input type="text" value="1"/>	
A13	<input type="text" value="Insert"/>	Local storage	<input type="text" value="3"/>	<input type="text" value="2"/>	<input type="text" value="2"/>	
A14	<input type="text" value="Insert"/>	Client trust	<input type="text" value="3"/>	<input type="text" value="5"/>	<input type="text" value="8"/>	
A15	<input type="text" value="Database"/>	CI CD Service	<input type="text" value="5"/>	<input type="text" value="7"/>	<input type="text" value="4"/>	Sdk3-3ddkj-0dkf

+ Add specific cloud Resources
Delete row
Submit

Figure 12. Preparation for Dynamic Risk Assessment process. Asset information collection

First, due to the dynamic nature of the cloud, the sensitivity values (i.e., the CIA triad) are assigned to resource types (i.e., these values will be applied to any resource of this type detected by the monitoring tools). Thus, it is required to provide the sensitivity values for all 14 types of Cloud Resources. Yet, if resources of a certain type are not expected to be present in the service under consideration, the values could be left as 1, and will have a negligible effect on the risk computation.

There is also the possibility to add specific resources. For these resources an ID must be provided. During the dynamic risk assessment, if the specific ID is detected, the values specified for this resource will be used (even though this resource may belong to a cloud resource type with other values). Only specific resources can be added or deleted.

The Dynamic risk assessment does not have a human readable output. Its results are used and displayed by other MEDINA components (e.g., *Automated Certificate Life-Cycle Manager*⁵ and *Continuous Certification Evaluation*⁶).

2.2.4. API

Figure 13 shows the APIs implemented by SATRA/RAOF. This API can be used for two purposes:

- 1) Operate the risk and non-conformity assessment process through a custom-built dashboard.
- 2) Use the dynamic risk assessment functionality during the continuous certification monitoring phase by other MEDINA components.

⁵ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D4.3 <https://doi.org/10.5281/zenodo.7927231>

⁶ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D4.3 <https://doi.org/10.5281/zenodo.7927231>

SATRA - Self-Assessment Tool for Risk Analysis ^{0.1}

[Base URL: /api/v1]
/api/v1/swagger.json

Manage interaction with the SATRA engine

registration Register a new practice for that user

POST /registration/toe/ Create a new practice

GET /registration/access_resp/{username}/{password} Get the access token from keycloak

DELETE /registration/delete_contract/{UUID} Delete a ToE

PUT /registration/update_contract/{UUID} Update ToE

practice Interact with the survey, update question/answers and get risk...

POST /practice/analysis/{UUID} Send information on a test result

POST /practice/answer/{UUID}/{question_id}/{answer_id} Send (eventually, update) an answer for a specific question

GET /practice/answer/{UUID}/{type_id} Get all the possible answers by type_id

GET /practice/answers/{UUID} Get the question and the answers chosen by the user for the ToE

POST /practice/asset_answers/{UUID} Send (eventually, update) an asset answers

GET /practice/assets/{UUID} Get all assets type

GET /practice/assets_answers/{UUID} Get all assets answer

DELETE /practice/assets_answers/{UUID}/{asset_id} Delete a specific asset throughout the name

GET /practice/assets_dynamic_answers/{UUID} Get all dynamic assets answer

GET /practice/assurance/{UUID} Get all possible assurance levels

GET /practice/certification/{UUID} Get all possible certification schemes

GET /practice/csp_market/{UUID} Get all possible CSP's markets

POST	/practice/dynamic_evaluated_risk/{UUID}	Dynamic evaluated risk computation
POST	/practice/map/{UUID}	Map an external questionnaire
GET	/practice/non_conformity_gap/{UUID}	Get all possible non conformity gap
GET	/practice/question/{UUID}	Get all the questions
GET	/practice/question/{UUID}/{question_id}	Get one question and its possible answers
GET	/practice/risk/{UUID}	Get the updated risk
GET	/practice/threats/{UUID}	Get the updated threat

Figure 13. SATRA/RAOF APIs

3. Delivery and usage

3.1. Licensing information

This component is offered under Apache 2.0 license. The license files and more detailed information can be found in the MEDINA Public GitLab repository⁷.

3.2. Download

The code of the component is available at the public GitLab repository of the MEDINA project:

<https://git.code.tecnalia.com/medina/public/static-risk-assessment-and-optimization-framework>

3.3. More information

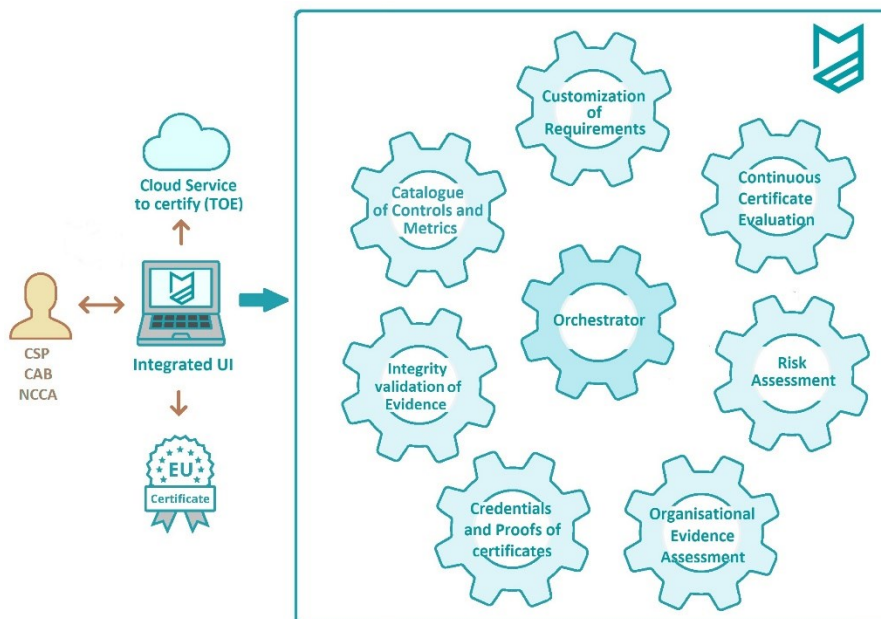
Interested readers can find more information about the *Risk Assessment and Optimisation Framework* at this link: <https://doi.org/10.5281/zenodo.7927217> D2.8 Risk-based techniques and tools for Cloud Security Certification – v3.

The MEDINA web site (<https://medina-project.eu/>) also includes several deliverables and blog posts related to the RAOF.

⁷ <https://git.code.tecnalia.com/medina/public/static-risk-assessment-and-optimization-framework>

Organisational Evidence Assessment

- User Manual -



Project Title:	MEDINA - Security Framework to achieve a continuous audit-based certification in compliance with the EU-wide cloud security certification scheme
Project Number:	952633
Editor:	Franz Deimling (Fabasoft R&D GmbH)
Version:	v1.0
Date:	31.07.2023
Distribution level:	PU



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952633

Table of contents

1. Introduction	4
1.1. User Roles and Permissions	4
2. User Manual	5
2.1. Toolbar	5
2.2. Uploaded files.....	5
2.2.1. Upload a policy file	5
2.2.2. View extracted evidence / assessment results	7
2.2.3. View compliance status / assessment result details	8
2.2.4. View extracted evidence in HTML.....	9
2.3. API	10
3. Delivery and Usage.....	12
3.1. Licensing information.....	12
3.2. Download	12
3.3. More information.....	12

List of Figures

FIGURE 1. AMOE TOOLBAR	5
FIGURE 3. AMOE LANDING PAGE.....	5
FIGURE 2. AMOE FILE UPLOAD DIALOG.....	6
FIGURE 3. AMOE LANDING PAGE AFTER FILE UPLOAD.....	6
FIGURE 4. AMOE EVIDENCE EXTRACTION PROGRESS	7
FIGURE 5. AMOE ASSESSMENT STATUS OVERVIEW PER DOCUMENT.....	7
FIGURE 6. AMOE OVERVIEW OF EXTRACTED EVIDENCE AND META DATA LINKED TO THE UPLOADED FILE	8
FIGURE 7. AMOE VIEW OF ORGANISATIONAL EVIDENCE.....	9
FIGURE 8. SHOW PROCESSED EVIDENCE (HTML VIEW)	10
FIGURE 9. AMOE API.....	11

1. Introduction

The *Assessment and Management of Organisational Evidence* component (a.k.a. *AMOE*) enables Cloud Service Providers (CSP) to generate assessment results for the continuous auditing of their security policy documents.

AMOE allows a compliance manager or an auditor to upload a policy document and view extracted evidence parts of the document relevant for MEDINA security metrics. The application can provide pre-assessment results, requiring the user only to inspect, confirm or edit and submit the final assessment result to the MEDINA framework. The main parts of *AMOE* are summarized in the following:

- Upload a policy document. The application processes the document and extracts evidence, and generates pre-assessment hints based on the metrics of the *Catalogue of Controls and Metrics*¹ and their respective configurations (e.g., Target Values) stored in the *Orchestrator*².
- View extracted assessment results.
- Set/Change assessment results.
- Submit assessment results.
- Delete all the data uploaded and stored directly in the component (except log data).

1.1. User Roles and Permissions

Access to *AMOE* is managed by Keycloak³. The visibility of the different elements of the *AMOE* UI, and the operations that are allowed to be carried out, are conditioned by the role to which each user is assigned. The main permissions are the following:

- **Read entities:** Read everything in *AMOE* that the user is allowed to (linked to a cloud service configured for the user).
- **Upload and delete entities:** Upload files and delete all data in *AMOE*.
- **Edit:** Edit assessment status of results, edit comments, and submit assessment results to the *Orchestrator*.

The table below details which actions are allowed for each of the defined roles:

Roles	Allowed Actions
IT Security Governance	Read entities
Security Analyst	Read entities
Domain Governance	Read entities
Product and Service Owner	Read entities
Product (Security) Engineer	Read entities, Upload and delete entities
Chief Information Security Office (CISO)	Read entities
Customer	None
Auditor	Read entities, edit assessment results, add comments, submit assessment results

¹ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D2.2 <https://doi.org/10.5281/zenodo.7794478>

² For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D3.6 <https://doi.org/10.5281/zenodo.7927225>

³ <https://www.keycloak.org>

2. User Manual

2.1. Toolbar

AMOE includes a toolbar (see Figure 1), always accessible in the upper area. The *Uploaded files* button provides access to the list of files that have been uploaded to AMOE. The *Help* button provides access the user manual.

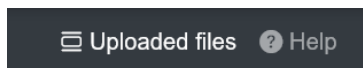


Figure 1. AMOE toolbar

2.2. Uploaded files

The *Uploaded files* view is the main view of AMOE (see Figure 2). It first presents an overview of existing uploaded policy files and the possibility to upload a new one.

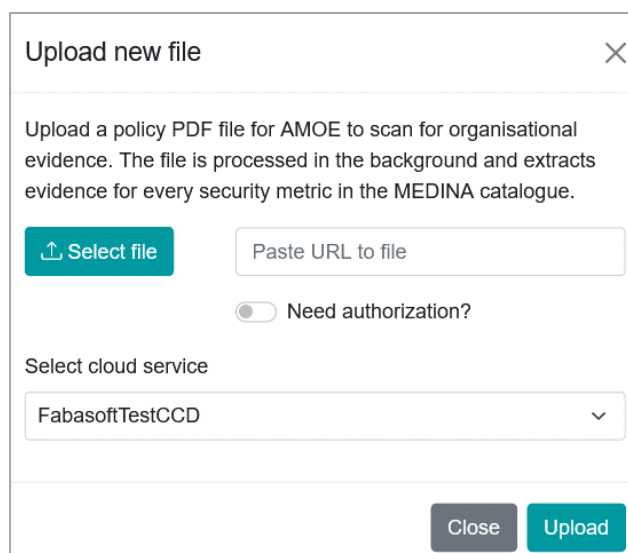
The screenshot shows the AMOE landing page. At the top right, there is a toolbar with 'Uploaded files' and 'Help' buttons. Below this, the page title is 'Process organisational evidence based on metrics'. A sub-header 'Uploaded files' is followed by a table. Above the table, there is a search bar and a button labeled 'Upload new file'. The table has columns for 'Cloud service', 'File name', 'Date', 'Progress', and 'Delete'. It lists several files, mostly from 'Fabasoft Cloud Service', with their names, upload dates, progress bars, and delete buttons. The last row is from 'Bosch_IaaS'.

Cloud service	File name	Date	Progress	Delete
Fabasoft Cloud Service	MEDINA_dummy_policies_Fabasoft_M18v5.pdf	2023-07-24 10:51:46	100.0%	Delete
Fabasoft Cloud Service	MEDINA_dummy_policies_Fabasoft_M18v5.pdf	2023-07-24 10:34:16	100.0%	Delete
Fabasoft Cloud Service	UTF-8MEDINA20dummy20policies20Fabasoft20M18.pdf	2023-05-31 10:19:27	100.0%	Delete
Fabasoft Cloud Service	MEDINA_dummy_policies_Fabasoft_M18v5.pdf	2023-05-24 05:56:04	100.0%	Delete
Fabasoft Cloud Service	MEDINA_dummy_policies_Fabasoft_M18v5.pdf	2023-05-02 10:12:47	process has been interrupted	Delete
Fabasoft Cloud Service	MEDINA_dummy_policies_Fabasoft_M18v5.pdf	2023-04-27 12:22:44	100.0%	Delete
Fabasoft Cloud Service	MEDINA_dummy_policies_Fabasoft_M18v5.pdf	2023-04-27 09:05:57	100.0%	Delete
Fabasoft Cloud Service	MEDINA_dummy_policies_Fabasoft_M18v5.pdf	2023-04-26 12:23:21	99.95%	Delete
Bosch_IaaS	today_Bosch_IoT_Cloud_Security_Concept.pdf	2023-04-20 12:43:35	99.98%	Delete

Figure 2. AMOE landing page

2.2.1. Upload a policy file

To use the AMOE GUI, start by clicking on the "Upload new file" button. A file upload dialog box will then appear, as shown in Figure 3. This dialog is only available for users with "upload and delete" permissions (see section 1.1).



Upload new file [X]

Upload a policy PDF file for AMOE to scan for organisational evidence. The file is processed in the background and extracts evidence for every security metric in the MEDINA catalogue.

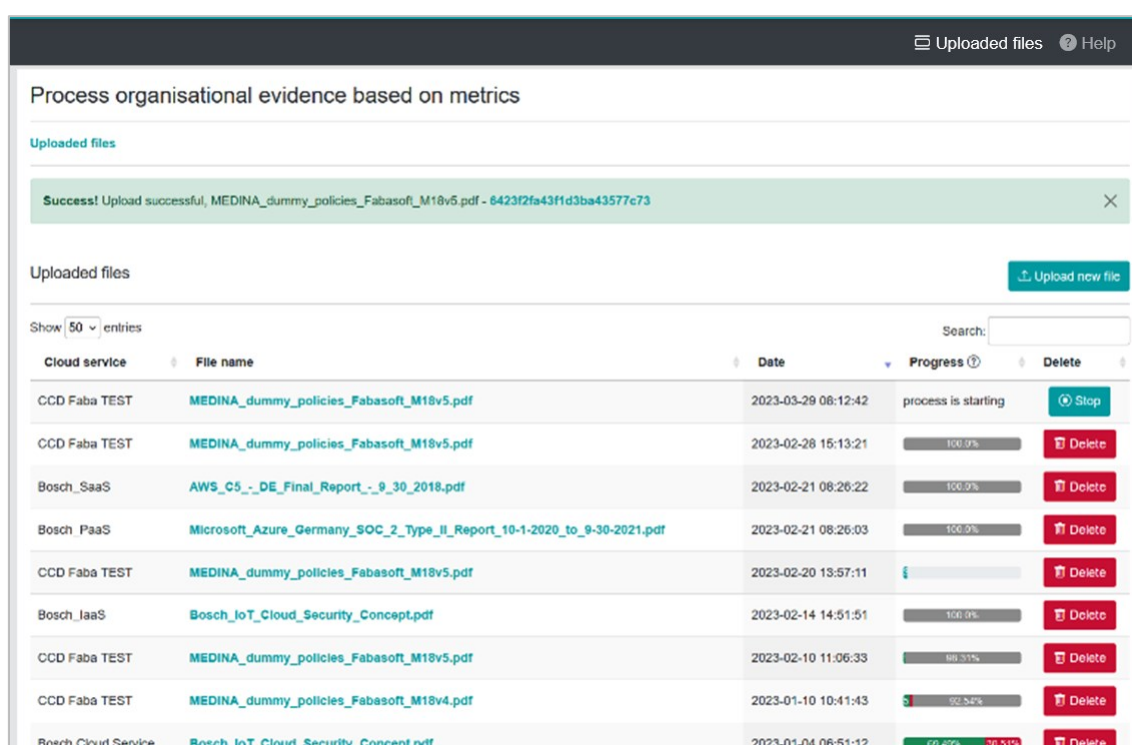
☐ Need authorization?

Select cloud service

FabasoftTestCCD [v]

Figure 3. AMOE file upload dialog

Select a policy PDF document to upload, or enter a URL from where AMOE should download the file. If the file download requires credentials, these can be provided by using the *Need authorization* functionality. Also enter the cloud service (id) to connect to. Then click on *Upload* and the page shown in Figure 4 is displayed.



Process organisational evidence based on metrics

Uploaded files [X] [Help](#)

Success! Upload successful, MEDINA_dummy_policies_Fabasoft_M18v5.pdf - 6423f2fa43f1d3ba43577c73

Uploaded files [Upload new file](#)

Show 50 entries Search: []

Cloud service	File name	Date	Progress	Delete
CCD Faba TEST	MEDINA_dummy_policies_Fabasoft_M18v5.pdf	2023-03-29 08:12:42	process is starting	Stop
CCD Faba TEST	MEDINA_dummy_policies_Fabasoft_M18v5.pdf	2023-02-28 15:13:21	100.0%	Delete
Bosch_SaaS	AWS_C5_-_DE_Final_Report_-_9_30_2018.pdf	2023-02-21 08:26:22	100.0%	Delete
Bosch_PaaS	Microsoft_Azure_Germany_SOC_2_Type_II_Report_10-1-2020_to_9-30-2021.pdf	2023-02-21 08:26:03	100.0%	Delete
CCD Faba TEST	MEDINA_dummy_policies_Fabasoft_M18v5.pdf	2023-02-20 13:57:11		Delete
Bosch_IaaS	Bosch_IoT_Cloud_Security_Concept.pdf	2023-02-14 14:51:51	100.0%	Delete
CCD Faba TEST	MEDINA_dummy_policies_Fabasoft_M18v5.pdf	2023-02-10 11:06:33	98.51%	Delete
CCD Faba TEST	MEDINA_dummy_policies_Fabasoft_M18v4.pdf	2023-01-10 10:41:43	92.52%	Delete
Bosch Cloud Service	Bosch_IoT_Cloud_Security_Concept.pdf	2023-01-04 06:51:12	69.49% 30.51%	Delete

Figure 4. AMOE landing page after file upload

The evidence extraction process is started in the background. It can take some time until every organisational metric has been processed. The process can be stopped by clicking on the turquoise *Stop* button in Figure 4. Files and their linked evidence can be deleted by clicking on the red *Delete* button.

Figure 5 depicts the progress of the background evidence extraction process. On hovering, the details are shown. Only users with “upload and delete” permission are shown the *Delete* and *Stop* buttons.

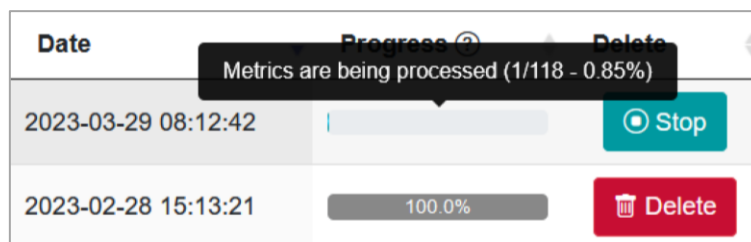


Figure 5. AMOE evidence extraction progress

Figure 6 shows the status overview. This is displayed for every file after the evidence extraction process has finished. The details are shown by hovering with the mouse. Green indicates the number of assessment results set to compliant, red the number set to not compliant, and grey marks where no status has been set (undefined).

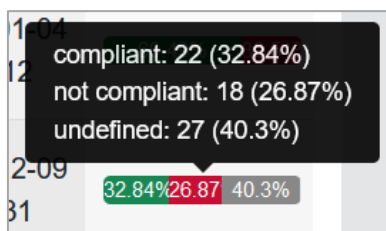


Figure 6. AMOE assessment status overview per document

2.2.2. View extracted evidence / assessment results

To view the evidence results of an uploaded file, click on a row of the respective table or the filename of the list, as depicted in Figure 4. The overview, as depicted in Figure 7, opens. This overview contains meta data of the uploaded file, filter, and search options. In case an assessment result has been set, it can be submitted to the *Orchestrator* directly from this view. Otherwise, click on a row to get to the detailed view for the extracted evidence.

Users with “edit” permission can use the *Submit* functionality from this page if the assessment status has been already set and the entry has not been submitted yet. Multiple assessment results can be sent at once if selected using the checkbox and clicking on the *Send multi assessment results* button.

View evidence

[Uploaded files](#) / MEDINA_dummy_policies_Fabasoft_M18v5.pdf

Information about the file

Cloud service

FabasoftTestCCD

File id

63e5fd22e09529afe53309c0

File name

MEDINA_dummy_policies_Fabasoft_M18v5.pdf

Uploaded on

2023-02-10 08:15:30 by ccd_admin

Extracted evidence count

118 / 118

Filter CAB assessment ?

Compliant:

10 / 11

Not compliant:

4

Undefined:

104 / 118 (88.14%)

Reset filter

Extracted evidence

Send multi assessment results

Show 50 entries

Search: passwordP

<input type="checkbox"/> MetricID	Question	Answer	AMOE assessment hint ?	CAB assessment ?	Submitted to Orchestrator ?
<input type="checkbox"/> PasswordPolicyQ1	Which parameters define the password policy?	Passwords should have at least 10 upper -and lowercase characters and contain numbers as well as special characters . Do not reuse passwords for multiple services. Passwords should not be	Undefined	✓ True	✓ Submitted
<input type="checkbox"/> PasswordPolicyQ2	What is the passwords maximum age according to the password policy?	encrypted passwords. The password needs to be changed after a maximum time duration of 60 days .	✓ True	✓ True	Submit
<input type="checkbox"/> PasswordPolicyQ3	What is the passwords rotation frequency?	Passwords should have at least 10 upper -and lowercase characters and contain numbers as well	✗ False	✗ False	Please add a comment to submit
<input type="checkbox"/> PasswordPolicyQ4	Which requirements exist for password managers?	best practice to use password managers to generate complex passwords and store the encrypted passwords . The password needs to be changed after a maximum time duration of 60	Undefined	Undefined	Please set CAB assessment status

Showing 1 to 4 of 4 entries (filtered from 118 total entries)

Previous
1
Next

Figure 7. AMOE overview of extracted evidence and meta data linked to the uploaded file

2.2.3. View compliance status / assessment result details

Figure 8 depicts the detailed view of the extracted evidence. The linked requirement is shown on the top. This is followed by the metric meta data, extracted answer, assessment hint and options to set the assessment result and comment. Only users with “edit” permission can change the assessment status and comment and submit the results to the *Orchestrator*.

Help

View compliance status

[Uploaded files](#) / [MEDINA_dummy_policies_Fabasoft_M18v5.pdf](#) / PasswordPolicyQ2

EUCS Requirement(s) linked to Metric

IAM-08 - PROTECTION AND STRENGTH OF CREDENTIALS

Requirement id

IAM-08.1H

Requirement description

The CSP shall document, communicate and make available to all users under its responsibility rules and recommendations for the management of credentials, including at least: (1) Non-reuse of credentials; (2) Trade-offs between entropy and ability to memorize; (3) Recommendations for renewal of passwords; (4) Rules on storage of passwords. (5) Recommendations on password managers (6) Recommendation to specifically address classical attacks, including phishing, social attacks, and whaling

Requirement assurance level

High

Requirement type

ORGANIZATIONAL

Metric - PasswordPolicyQ2

File

Keywords

password, age, maximum

Target value

<= 90.0 (Integer)

Question

What is the passwords maximum age according to the password policy?

Answer

encrypted passwords. The password needs to be changed after a maximum time duration of **60 days** .

File id

63e5fd22e09529afe53309c0

File name

MEDINA_dummy_policies_Fabasoft_M18v5.pdf

Extraction date

2023-02-10 08:37:07

Show processed

Show original

Assessment

Assessment hint

compliant (60 days <= 90.0 (Integer)) ?

Assessment status

☒ compliant ✓ ☐ not compliant ⚡

Last change on 2023-03-29 08:20:52 by admin

Compliance comment

Please enter a comment regarding the assessment status.

The comment has not been changed yet

Submit to orchestrator

No submission yet.

previous metric

84/118

next metric

Figure 8. AMOE view of organisational evidence

2.2.4. View extracted evidence in HTML

Figure 9 shows the processed HTML version of the document. The extracted evidence is highlighted in green.

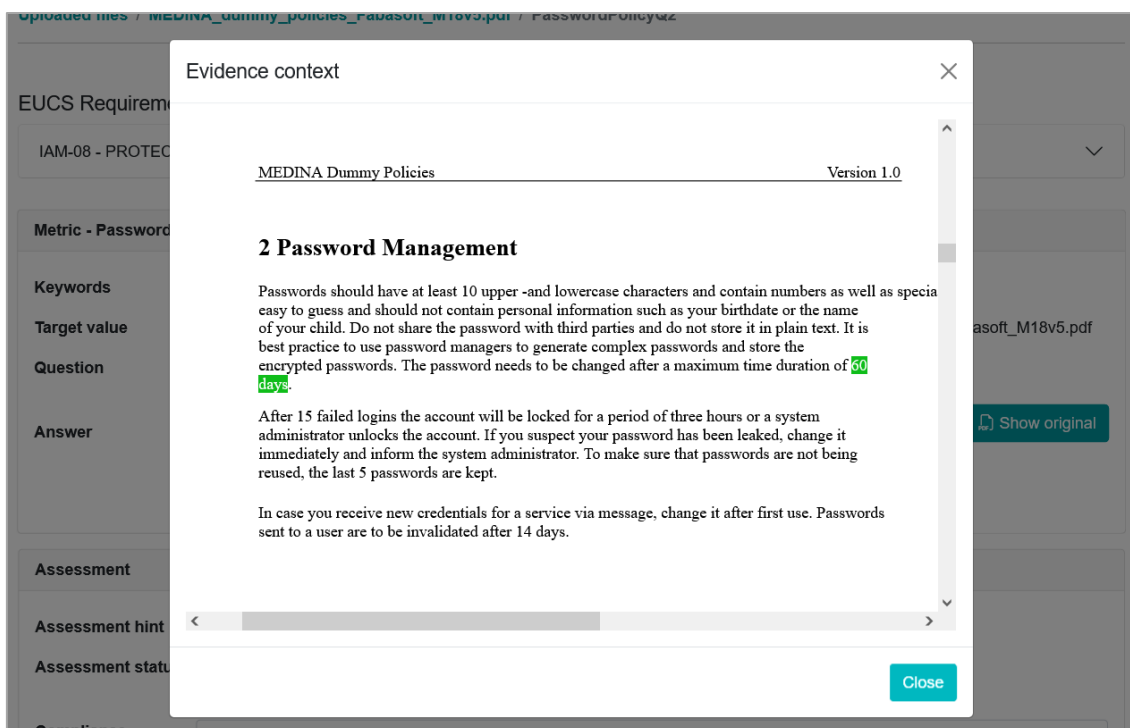


Figure 9. Show processed evidence (HTML view)

2.3. API

The OpenAPI file can be downloaded by accessing <host>/openapi.json

Figure 10 shows the list of available endpoints. The request must include a valid keycloak token issued from the MEDINA setup.

GET	/api/v1/files/{cloud_service_id}	AMOE List Files Cloud Sevice	▼
POST	/api/v1/files/	AMOE List Files Cloud Sevice	▼
GET	/api/v1/file/{file_id}	AMOE Get File	▼
GET	/api/v1/file/last/{cloud_service_id}	get_amoe_last_file	▼
GET	/api/v1/evidence/list/{file_id}	AMOE Get List Evidence For File	▼
POST	/api/v1/evidence/list_per_metric_id	AMOE Get List Evidence Per Metric	▼
GET	/api/v1/evidence/{evidence_id}	AMOE Get Evidence	▼
POST	/api/v1/evidence/assessment	AMOE Set Assessment Result	▼
GET	/api/v1/evidence/send_to_orchestrator/{evidence_id}	AMOE Send Assessment Result	▼
GET	/api/v1/evidence/file/{evidence_id}	AMOE Get HTML File	▼
GET	/api/v1/file/pdf/{file_id}	AMOE Get PDF File	▼
POST	/api/v1/file/{cloud_service}	AMOE Upload PDF File	▼
GET	/api/v1/file/delete/{file_id}	AMOE Delete File And Evidence	▼

Figure 10. AMOE API

3. Delivery and Usage

3.1. Licensing information

This component is offered under Apache 2.0 license. The license files and more detailed information can be found in the MEDINA Public GitLab repository⁴.

3.2. Download

The code of the component is available at the public GitLab repository of the MEDINA project:

<https://git.code.tecnalia.com/medina/public/AMOE>

3.3. More information

Interested readers can find more information about *AMOE* in the following links:

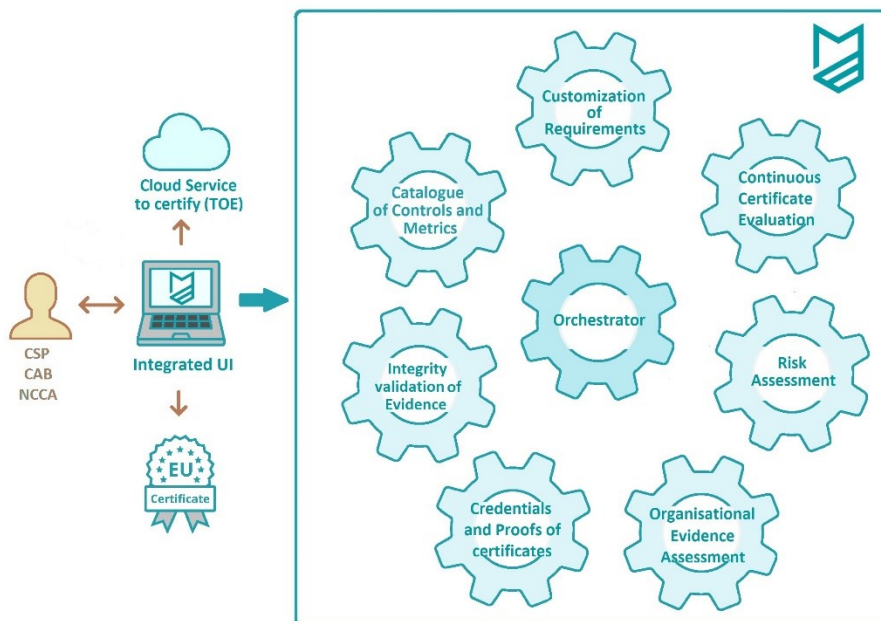
<https://doi.org/10.5281/zenodo.7927225> “D3.6 Tools and techniques for collecting evidence of technical and organisational measures – v3”

The MEDINA web site (<https://medina-project.eu/>) also includes several deliverables related to *AMOE*.

⁴ <https://git.code.tecnalia.com/medina/public/AMOE>

Continuous Certificate Evaluation

- User Manual -



Project Title:	MEDINA - Security Framework to achieve a continuous audit-based certification in compliance with the EU-wide cloud security certification scheme
Project Number:	952633
Editor:	Hrvoje Ratkajec (XLAB)
Version:	v1.0
Date:	31.07.2023
Distribution level:	PU



Table of contents

1. Introduction 4

 1.1. User Roles and Permissions 4

2. User Manual 5

 2.1. Toolbar 5

 2.2. Evaluation overview 5

 2.3. Evaluation tree 6

 2.3.1. Navigation toolbar 6

 2.3.2. Tree view 9

3. Delivery and usage 11

 3.1. Licensing information 11

 3.2. Download 11

 3.3. More information 11

List of Figures

FIGURE 1. CCE TOOLBAR	5
FIGURE 2. CCE EVALUATION OVERVIEW	5
FIGURE 3. CCE EVALUATION TREE	6
FIGURE 4. NAVIGATION TOOLBAR.....	6
FIGURE 5. SELECT TARGET OF EVALUATION	6
FIGURE 6. HISTORY	7
FIGURE 7. ZOOM IN	7
FIGURE 8. ZOOM OUT	7
FIGURE 9. COLLAPSE ALL	8
FIGURE 10. EXPAND ALL	8
FIGURE 11. SHOW YELLOW NODES	9
FIGURE 12. EXAMPLE OF THE DETAILS OF A REQUIREMENT NODE.....	9
FIGURE 13. EXAMPLE OF THE DETAILS OF A RESOURCE NODE	10

1. Introduction

The *Continuous Certification Evaluation (CCE)* component of MEDINA collects assessment results and builds an evaluation tree representing the aggregated assessment results on higher levels of the certification scheme to determine compliance with the different certification elements.

The evaluation of security compliance in MEDINA starts with the gathering of evidence by different tools and techniques. Security assessment components assess this evidence based on the target values as configured for the specific requirement and provide their output (assessment results with the state of fulfilment of a specific metric for a specific monitored resource) to the *Continuous Certification Evaluation (CCE)* component. If the assessment result value represents the lowest-level information about the certification state, the role of the CCE component is to combine the received assessment results into information about the fulfilment of higher-level certification objects: requirements, controls, control groups, and the selected certificate scheme in its entirety. This information does not directly determine the cloud service's eligibility for a certificate, but serves as input for other components, the *Risk Assessment and Optimisation Framework (RAOF)*¹ and the *Certificate Lifecycle Manager (LCM)*², as well as for easy visualisation of the certificate state for the users (Cloud Service Providers - CSPs and auditors).

1.1. User Roles and Permissions

Access to CCE is managed by Keycloak³. The visibility of the different elements of the *Continuous Certification Evaluation (CCE)*, and the operations that are allowed to be carried out, are conditioned by the role to which each authenticated user is assigned.

The table below details which actions are allowed for each of the defined roles:

Roles	Allowed Actions
IT Security Governance	Read the evaluation tree
Security Analyst	Read the evaluation tree
Domain Governance	Read the evaluation tree
Product and Service Owner	Read the evaluation tree
Product (Security) Engineer	Read the evaluation tree
Chief Information Security Office (CISO)	Read the evaluation tree
Customer	None
Auditor	Read the evaluation tree

¹ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D4.5 <https://doi.org/10.5281/zenodo.7927237>

² For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D4.3 <https://doi.org/10.5281/zenodo.7927231>

³ <https://www.keycloak.org>

2. User Manual

2.1. Toolbar

The *Continuous Certification Evaluation (CCE)* includes a toolbar (see Figure 1), always accessible in the upper area, with all the options that are available in the tool:



Figure 1. CCE toolbar

The different menu options, which are described in the following sections, are as follows:

- **Evaluation overview:** Provides an aggregated view of compliant and non-compliant resources and requirements for all cloud services.
- **Evaluation tree:** Provides a graphical representation of the evaluation tree for each cloud service, showing detailed information about the assessments results.
- **Help:** Provides access to the user manual.

2.2. Evaluation overview

The *Evaluation overview* menu option (see Figure 2) provides an overview of compliant and non-compliant resources and requirements for all cloud services.

Cloud Service Name	Target of Evaluation	Resources Total	Resources Compliant	Resources Non-compliant	Requirements Total	Requirements Compliant	Requirements Non-compliant
MichelaCS2	MichelaCS2 : EUCS	0			998		
Bosch_SaaS	Bosch_SaaS : EUCS	0			998		
AlwaysGreenExceptOne	AlwaysGreenExceptOne : EUCS	71	71 ↑	0 ↓	1996	71 ↑	0 ↓
AlwaysGreen	AlwaysGreen : EUCS	71	71 ↑	0 →	1996	71 ↑	0 →
Bosch Cloud Service	Bosch Cloud Service : EUCS	0			998		
Bosch_PaaS	Bosch_PaaS : EUCS	16	9 →	7 →	1996	3 →	1 →
Bosch_IaaS	Bosch_IaaS : EUCS	39	9 →	30 ↑	1996	3 →	4 →
AlwaysRed	AlwaysRed : EUCS	71	0 →	71 ↑	1996	0 →	71 ↑

Figure 2. CCE Evaluation overview

On the left side is the list of cloud service and their Targets of evaluation. The Target of Evaluation (ToE) binds a cloud service to a certification framework (or catalogue). The right side shows total, compliant, and non-compliant resources and requirements.

Each compliant and non-compliant value also has a coloured arrow. There are three types of arrows:

- 1) Arrow up means that the number of compliant or non-compliant resources or requirements is increasing in relation to last couple of assessment result.
- 2) Arrow down means the number of compliant or non-compliant resources or requirements is decreasing in relation to last couple of assessment result.
- 3) Arrow to the right (in black colour) means the number of compliant or non-compliant resources or requirements has not changed from the last couple of assessment result.

The colour of the arrow indicates if the change is positive (green) or negative (red).

2.3. Evaluation tree

The *Evaluation tree* menu option (see Figure 3) provides a graphical representation of the evaluation tree for each cloud service, with detailed information about assessments results for the selected cloud service.

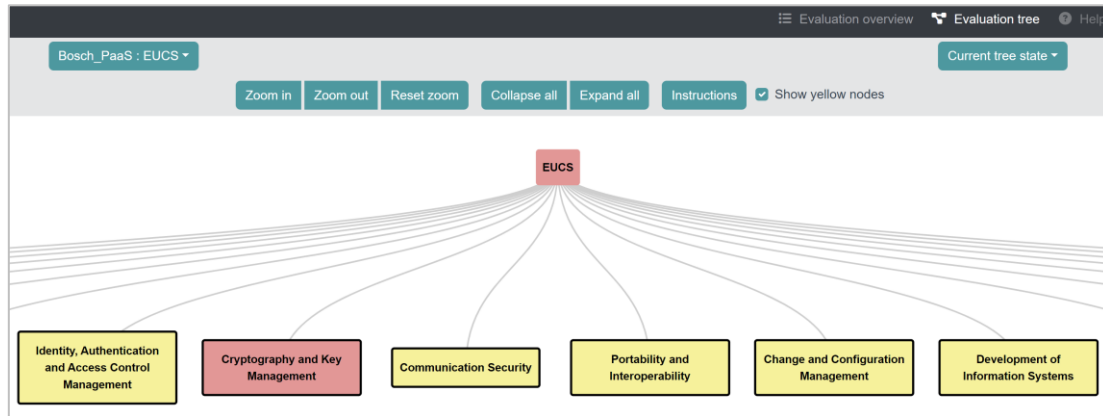


Figure 3. CCE Evaluation tree

2.3.1. Navigation toolbar

The *Navigation toolbar* option (see Figure 4) enables the user to move around and to interact with the evaluation tree using the following buttons.



Figure 4. Navigation toolbar

- 1) **Select Target of Evaluation:** if a user has access to multiple Targets of Evaluation, they can switch between them with the drop-down menu on the top left (see Figure 5).

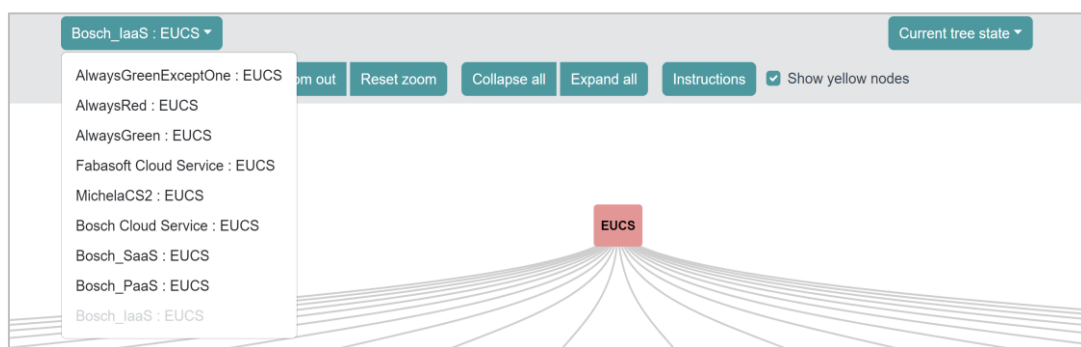


Figure 5. Select Target of Evaluation

- 2) **History:** the top-right button enables to user to review the history (past results) for each selected Target of Evaluation (see Figure 6).

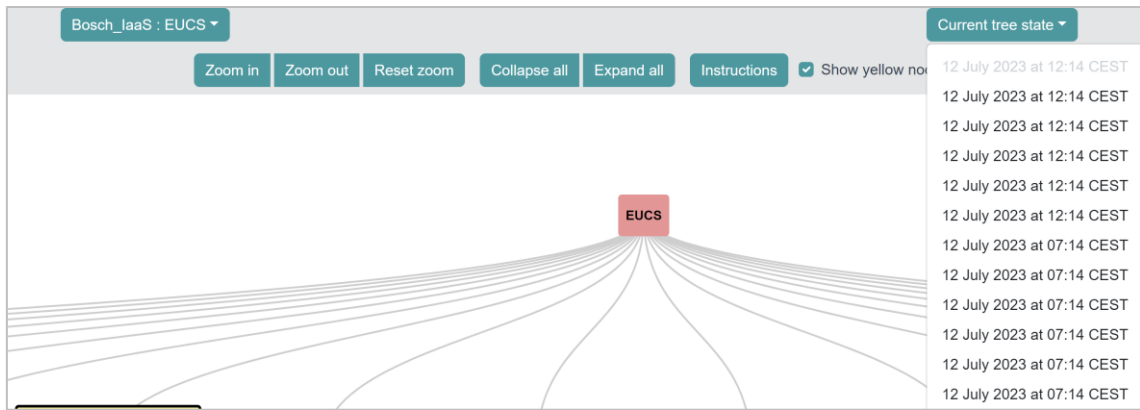


Figure 6. History

3) **Zoom in:** enables to zoom in on the current part of the tree (see Figure 7).

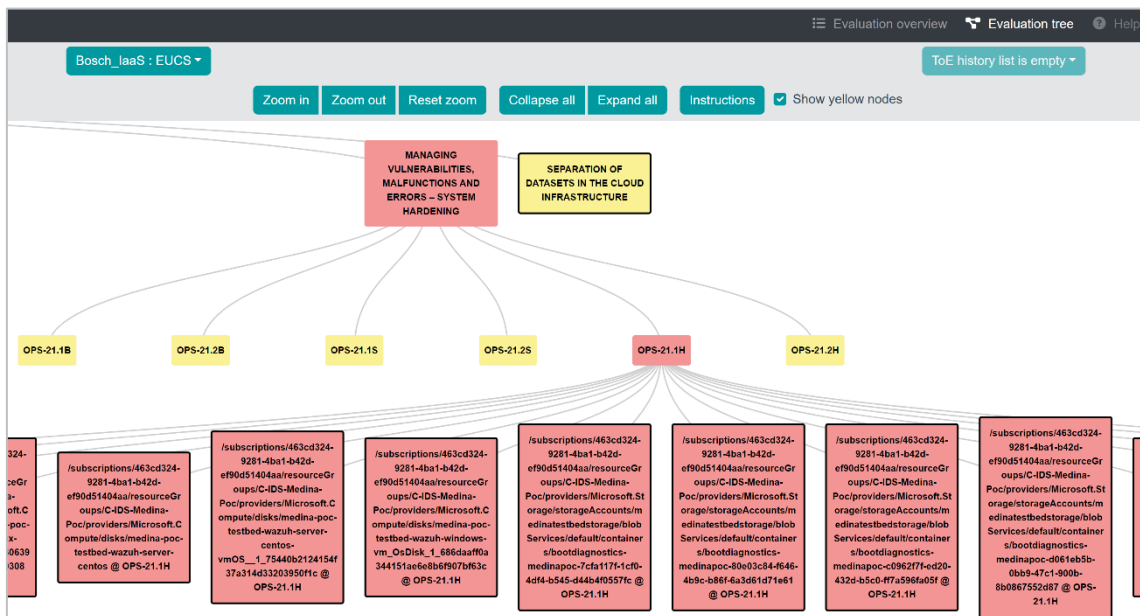


Figure 7. Zoom in

4) **Zoom out:** enables to zoom out to see an entire tree (see Figure 8).

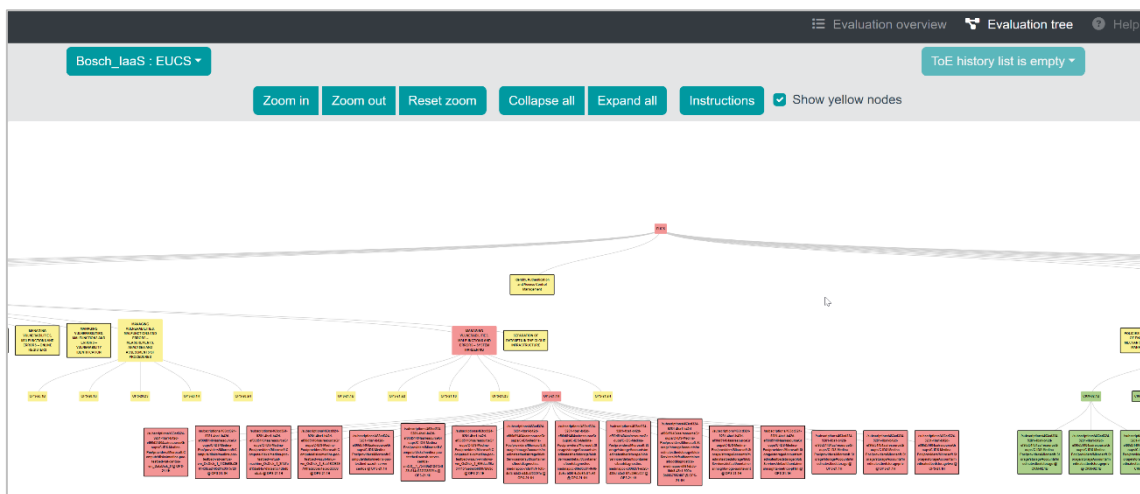


Figure 8. Zoom out

- 5) **Reset zoom:** sets zoom to the default value, i.e., the value used when the tree is first displayed.
- 6) **Collapse all:** removes all nodes and shows only the highest node (security framework) (see Figure 9).

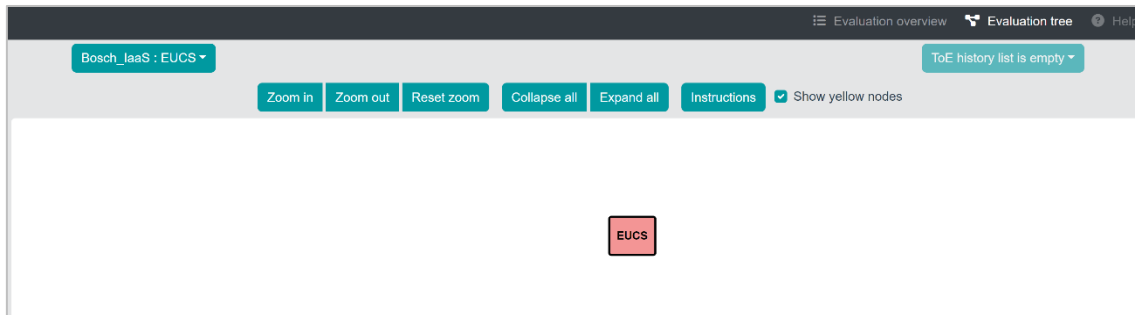


Figure 9. Collapse all

- 7) **Expand all:** shows all nodes in a tree like structure, expanding from the security framework to Control groups, Controls, Requirements, Resources and Metrics (see Figure 10).

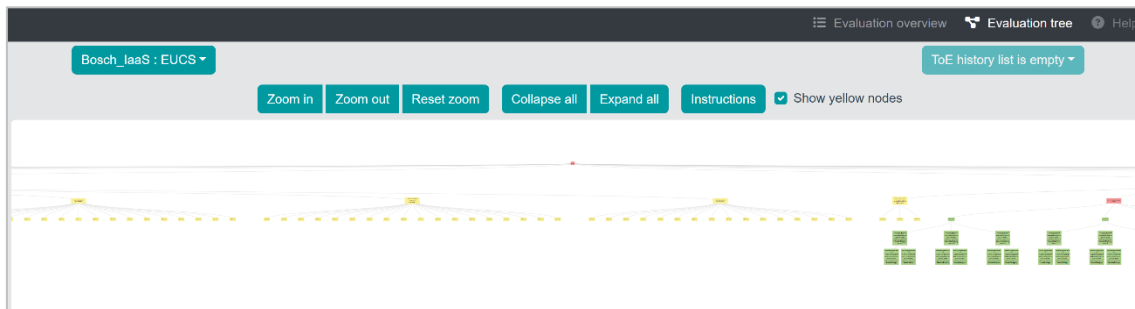


Figure 10. Expand all

- 8) **Show yellow nodes:** yellow nodes (no assessment results) are shown by default. Unticking the box hides yellow nodes and leaves only compliant (green) and non-compliant (red) nodes (see Figure 11).

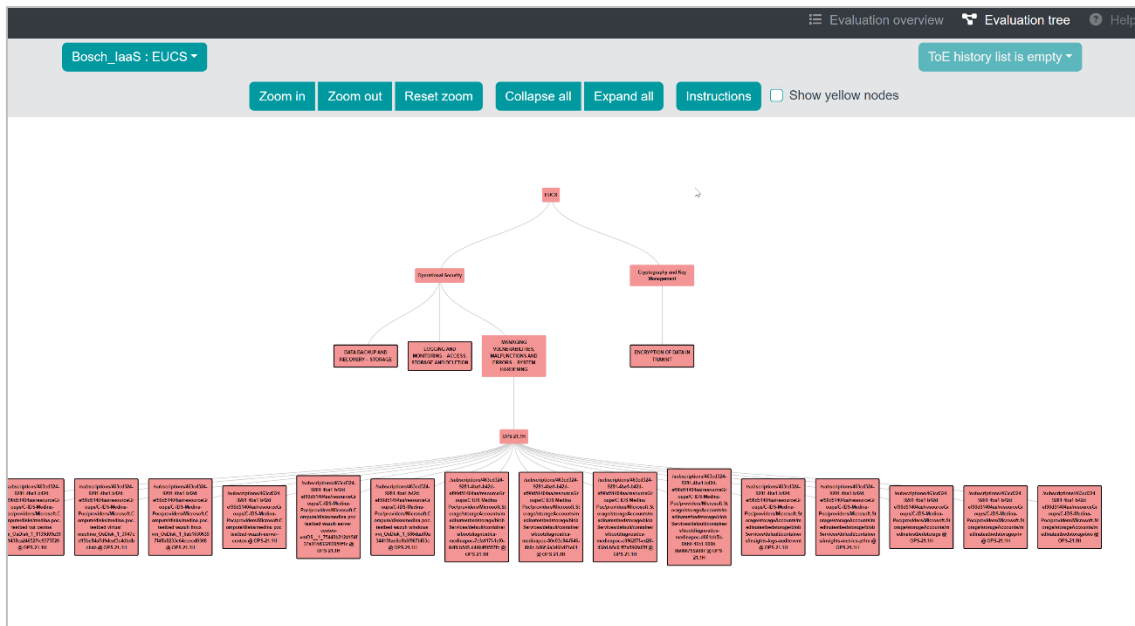


Figure 11. Show yellow nodes

- 9) **Instructions:** provides information on how to manipulate the nodes (left click: hide/show child nodes; right click: permanently open node details).

2.3.2. Tree view

The user can move the tree by holding down the left mouse button and moving the mouse. By left-clicking on the node, the user can show or hide child nodes. Right clicking on a node permanently opens an additional window with details of the node (see Figure 13 and Figure 13). This window also opens temporary when hovering the mouse over the node.

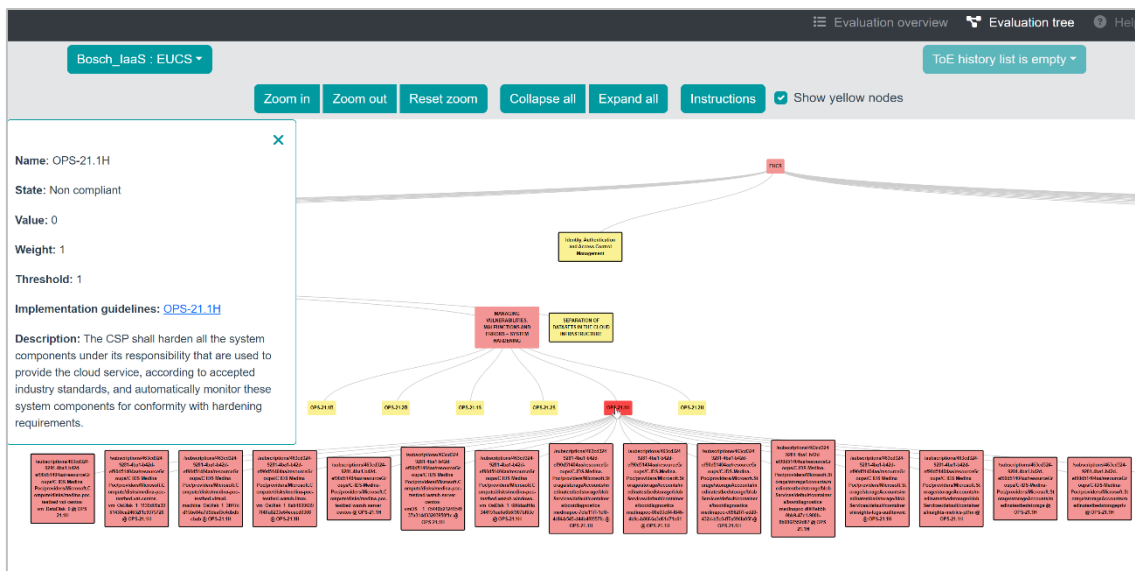


Figure 12. Example of the details of a Requirement node

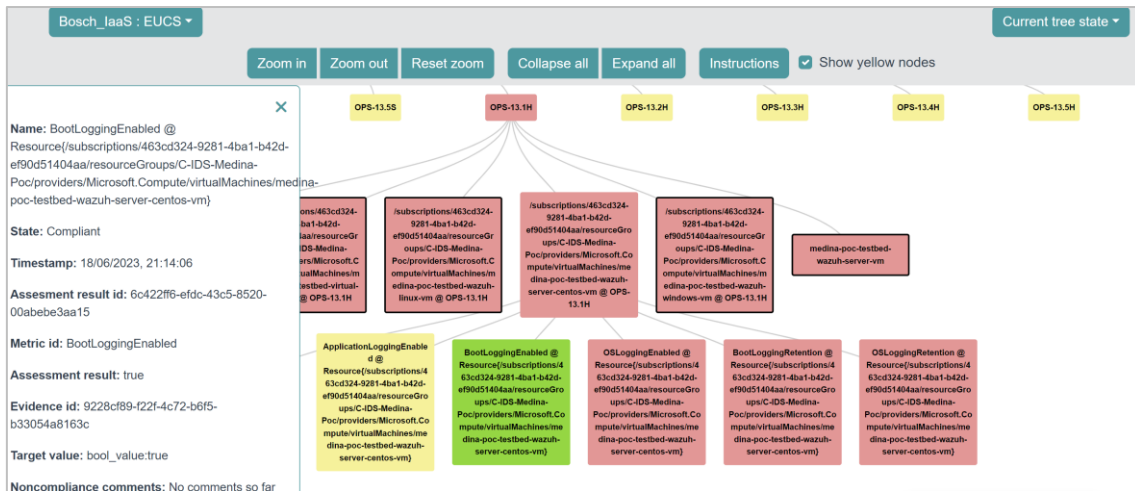


Figure 13. Example of the details of a Resource node

3. Delivery and usage

3.1. Licensing information

This component is offered under Apache 2.0 license. The license files and more detailed information can be found in the MEDINA Public GitLab repository⁴.

3.2. Download

The code of the component is available at the public GitLab repository of the MEDINA project:

- CCE back-end (core): <https://git.code.tecnalia.com/medina/public/continuous-certification-evaluation>
- CCE front-end (web UI): <https://git.code.tecnalia.com/medina/public/cce-frontend>
- Java library for communication with the *Catalogue of Controls and Metrics*⁵: <https://git.code.tecnalia.com/medina/public/catalogue-client-java>
- Java library for communication with the *Orchestrator*⁶: <https://git.code.tecnalia.com/medina/public/orchestrator-client-java>

3.3. More information

Interested readers can find more information about the *Continuous Certificate Evaluation* at this link: <https://doi.org/10.5281/zenodo.7927231> “D4.3 Tools and Techniques for the Management and Evaluation of Cloud Security Certifications – v3”.

The MEDINA web site (<https://medina-project.eu/>) also includes several deliverables and blog posts related to the *Continuous Certificate Evaluation* component.

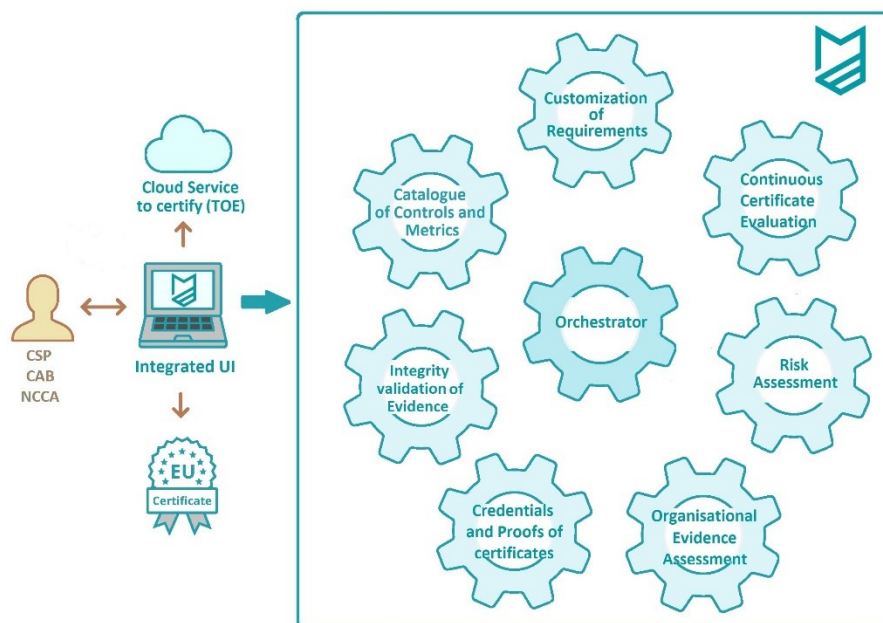
⁴ <https://git.code.tecnalia.com/medina/public/continuous-certification-evaluation>

⁵ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D2.2 <https://doi.org/10.5281/zenodo.7794478>

⁶ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D3.6 <https://doi.org/10.5281/zenodo.7927225>

Credentials and Proofs of Certificates

- User Manual -



Project Title:	MEDINA - Security Framework to achieve a continuous audit-based certification in compliance with the EU-wide cloud security certification scheme
Project Number:	952633
Editor:	Cristina Regueiro (Fundación TECNALIA Research and Innovation)
Version:	v1.0
Date:	31.07.2023
Distribution level:	PU



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952633

Table of contents

1. Introduction	4
1.1. User Roles and Permissions	4
2. User Manual	5
2.1. Toolbar	5
2.2. Certificate Credentials Lifecycle	5
2.3. Certificate Credentials	6
2.4. Certificate Proofs	6
2.5. Administration	6
2.5.1. Connections	6
3. Certificate Credentials Issuance (CAB)	9
4. Certificate Proofs Request (CSP Customer)	11
5. Delivery and usage	13
5.1. Licensing information	13
5.2. More information	13

List of Figures

FIGURE 1. SELF-SOVEREIGN IDENTITY (SSI) FRAMEWORK TOOLBAR

FIGURE 2. SELF-SOVEREIGN IDENTITY (SSI) FRAMEWORK CONTEXT.....

FIGURE 3. SSI FRAMEWORK SECURITY CREDENTIALS LIST

FIGURE 4. SSI FRAMEWORK SECURITY PROOFS LIST

FIGURE 5. SELF-SOVEREIGN IDENTITY (SSI) FRAMEWORK EXISTING CONNECTIONS

FIGURE 6. SELF-SOVEREIGN IDENTITY (SSI) FRAMEWORK SHARING CONNECTION

FIGURE 7. SELF-SOVEREIGN IDENTITY (SSI) FRAMEWORK CONNECTION DEFINITION.....

FIGURE 8. SELF-SOVEREIGN IDENTITY (SSI) FRAMEWORK EXTERNAL CONNECTION PAGE FOR ISSUER

FIGURE 9. SELF-SOVEREIGN IDENTITY (SSI) FRAMEWORK CERTIFICATE CREDENTIAL ISSUANCE

FIGURE 10. SSI FRAMEWORK EXTERNAL CONNECTION PAGE FOR VERIFIER

FIGURE 11. SSI FRAMEWORK CERTIFICATE PROOFS REQUEST

FIGURE 12. SSI FRAMEWORK CERTIFICATE PROOFS RESPONSE.....

5

5

6

6

7

7

8

9

9

11

11

12

1. Introduction

The functionality of checking Credentials and Proofs of Certificates in MEDINA is provided by the *Self-Sovereign Identity (SSI) Framework* tool. The *SSI Framework* provides Cloud Service Providers (CSPs) with the capability to manage their own security certificates as part of their identity through verifiable credentials. “To manage their own identity” ultimately means that they store their identity on their own “user space” without intervention of a third-party.

The *SSI Framework* is not only composed of the CSP component to store and control its own credentials. It is also composed of the *Issuer* component, which provides Conformance Assessment Bodies (CABs) a way to issue verifiable credentials about the security certificates related to the CSPs; and the *Client* component, which provides a way to ask and verify proofs of different security certificates features. In this sense, privacy is an important requirement within MEDINA, as several security certificates features are considered sensitive and must be treated carefully. The *SSI Framework* is capable of sharing sensitive information in a confidential way by keeping the user’s identity out of the reach of third parties, that act as identity silos, reducing the risk of identity theft; but also by using Zero-Knowledge Proofs (ZKPs).

1.1. User Roles and Permissions

Access to the *SSI Framework* is managed by Keycloak¹. The visibility of the different elements of the *Self-Sovereign Identity (SSI) Framework*, and the operations that are allowed to be carried out are conditioned by the role to which each authenticated user is assigned.

The table below details which actions are allowed for each of the defined roles:

Roles	Allowed Actions
IT Security Governance	Read credentials and proofs
Security Analyst	None
Domain Governance	Read credentials and proofs
Product and Service Owner	None
Product (Security) Engineer	None
Chief Information Security Office (CISO)	Read credentials and proofs
Customer	Read proofs
Auditor	Read credentials and proofs, Issue credentials

¹ <https://www.keycloak.org>

2. User Manual

2.1. Toolbar

The *Self-Sovereign Identity (SSI) Framework* includes a toolbar (see Figure 1), always accessible in the upper area, with all the options that are available in the tool:

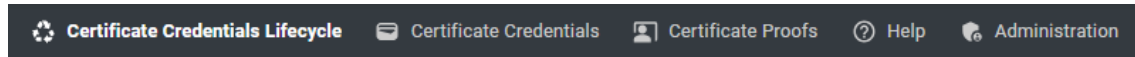


Figure 1. Self-Sovereign Identity (SSI) Framework toolbar

The different menu options, which are described in the following sections, are as follows:

- **Certificate Credentials Lifecycle:** Provides contextualization information for a better understanding of the tool operation.
- **Certificate Credentials:** Provides access to the list of stored verifiable credentials about the security certificate status of the CSP's Cloud Services
- **Certificate Proofs:** Provides access to the list of the proof requests received about the security certificate status of the CSP's Cloud Services.
- **Help:** Provides access to the user manual of the tool.
- **Administration:** Provides access to information on connections. This option is only available to users with administration rights.

2.2. Certificate Credentials Lifecycle

The *Certificate Credentials Lifecycle* view (see Figure 2) provides an overview of the tool's components and operation for a better understanding of the tool.

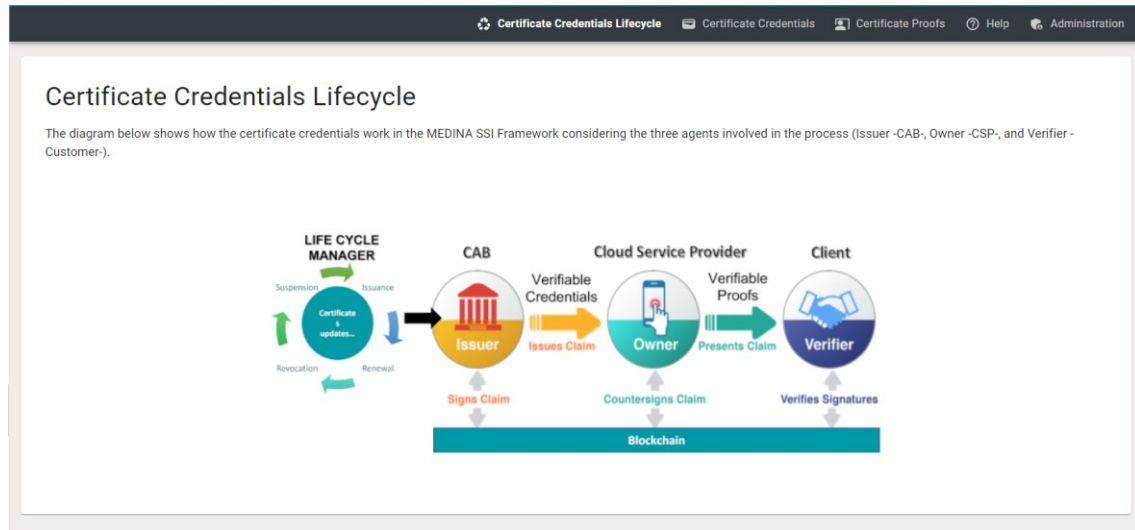


Figure 2. Self-Sovereign Identity (SSI) Framework context

As a summary, the *SSI Framework* is composed of three main components:

- Certificate signing graphical application for the CAB: to issue, update, or revoke security certificates of a CSP's Cloud Service based on the updated certificate state received from the *Automated Certificate Lifecycle Manager*² component.

² For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D4.3 <https://doi.org/10.5281/zenodo.7927231>

- Graphical application for the CSPs: to save the signed security certificates as well as to generate verifiable proofs based on them. This application has been integrated in the *MEDINA Integrated UI*.
- Graphical application for CSP clients: to request and verify proofs of security certificates.

2.3. Certificate Credentials

The *Certificate Credentials* view provides a list of the existing certificate credentials of the Cloud Services in the CSP. The complete list of security credentials received appears as shown in Figure 3. The certificate credentials can be copied or removed using the buttons on the right.

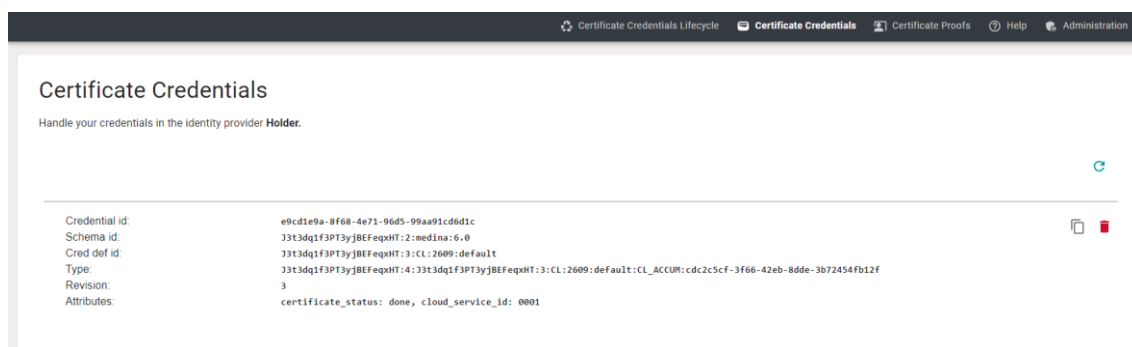


Figure 3. SSI Framework security credentials list

2.4. Certificate Proofs

The *Certificate Proofs* view provides a list of the received certificate proofs of the Cloud Services in the CSP. The complete list of certificate proofs received appears as shown in Figure 4. The certificate proofs can be copied or removed using the buttons on the right.

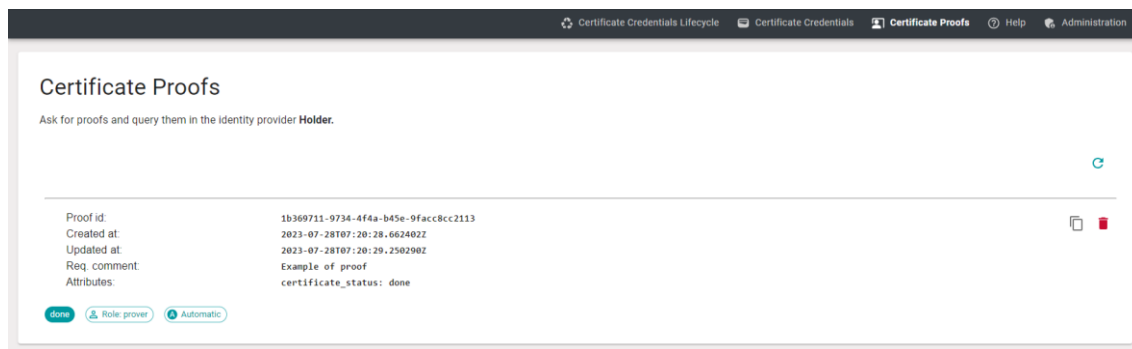


Figure 4. SSI Framework security proofs list

2.5. Administration

The *Administration* view provides a way for administrators to correctly configure the *SSI Framework* to work with other SSI agents (issuer and verifier).

2.5.1. Connections

The *Connections* view provides a way to list existing connections, as shown in Figure 5, as well as to create new connections to other SSI agents (issuer and verifier), if needed. This option is only available to users with administration rights.

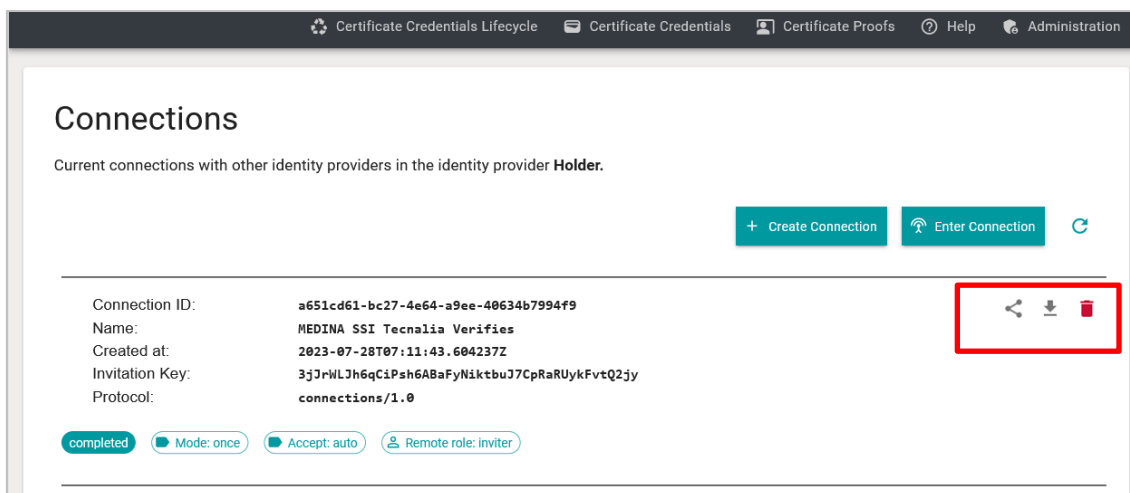


Figure 5. Self-Sovereign Identity (SSI) Framework existing connections

The required connections are set by default. However, if new connections are needed, they can be created via the *Create Connection* button. A new entry will then appear in the list. This new entry includes the *Share*, *Download* and *Remove* button.

Clicking on the *Share* button, a dialog box will be opened as shown in Figure 6. This dialog box contains a QR code with the invitation that the other party can scan to comfortably enter the invitation. Apart from that code, a *Copy to clipboard* button will allow to copy the invitation to the clipboard.



Figure 6. Self-Sovereign Identity (SSI) Framework sharing connection

The user who is going to use this invitation to open a new connection to the former SSI agent will have to enter it manually (as shown in Figure 7) or simply scan it from its browser if both users are in the same location. The invitation can be shared using any external secure communication mechanism, such as email or SMS.

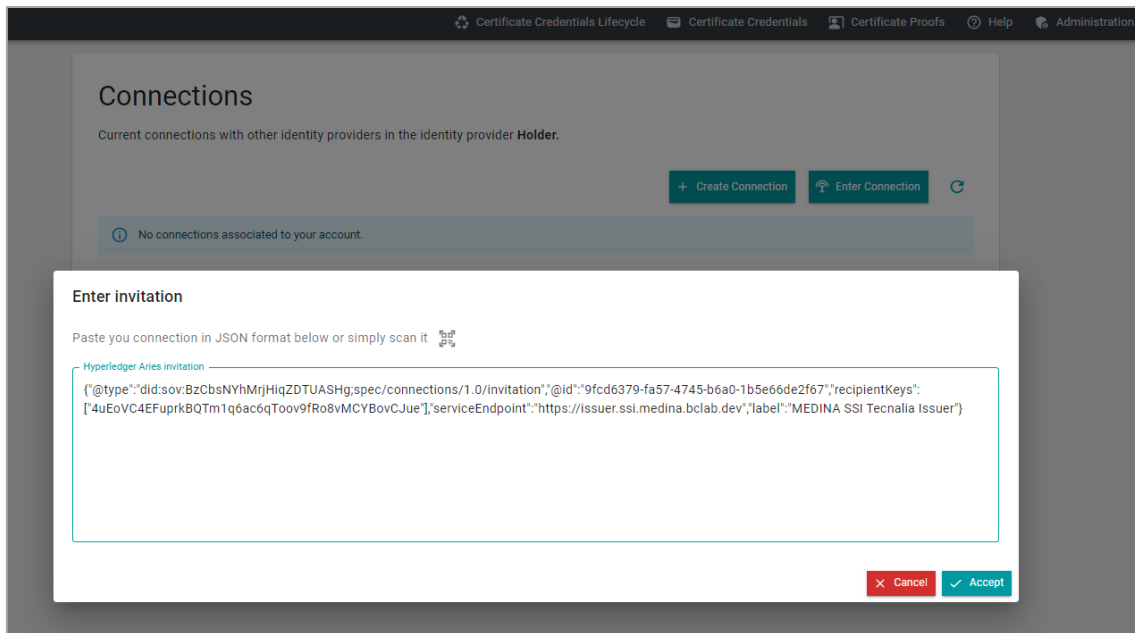


Figure 7. Self-Sovereign Identity (SSI) Framework connection definition

Any SSI-agent will automatically accept the invitation and complete the invitation procedure. Eventually both the invitation sender and receiver will see the new connection listed and marked as “completed”, as shown in Figure 5.

3. Certificate Credentials Issuance (CAB)

In addition to the graphical application for the CSPs integrated in the *MEDINA Integrated UI* described in Section 2, there is also an additional graphical UI for the CAB to issue new certificate credentials, which is available at: <https://medina-webapp.cybersec.digital.tecnalia.dev/>.

The first thing the webapp asks the user to do is to connect to one of the available SSI-agents. In this case, the “Issuer” agent must be selected, as shown in Figure 8.

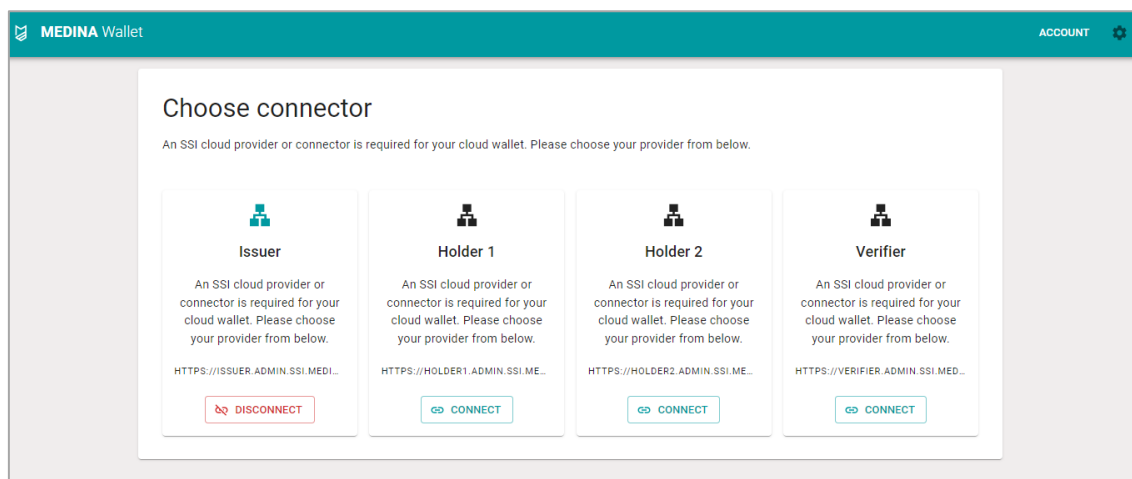


Figure 8. Self-Sovereign Identity (SSI) Framework external connection page for issuer

Once connected, the most interested webapp page for the CAB is related to the “Certificate Credentials”, where new certificate credentials can be issued through the *Create Credential* button. Automatically, a form like the one in Figure 9 is displayed.

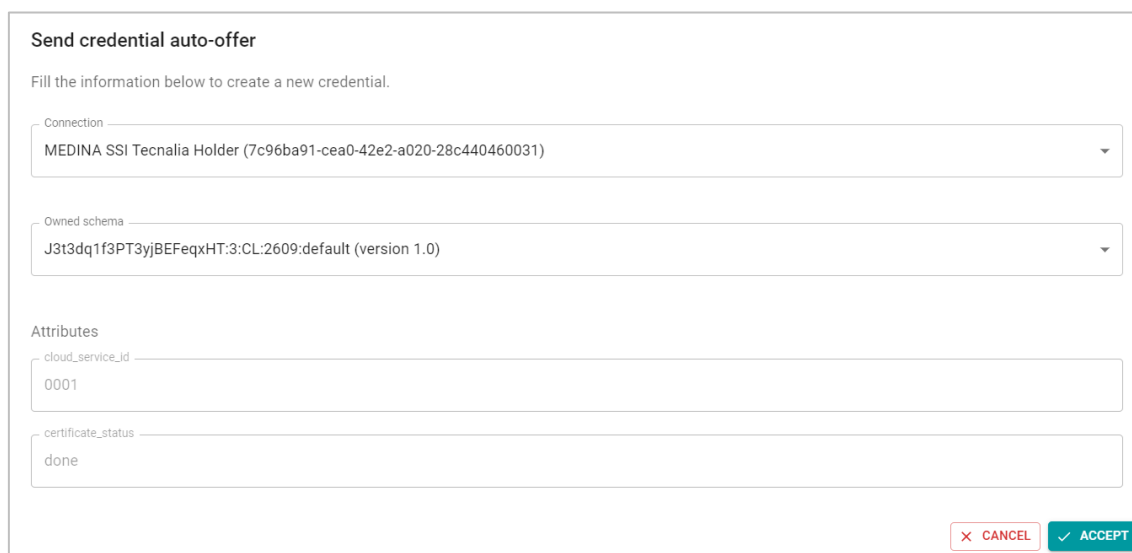


Figure 9. Self-Sovereign Identity (SSI) Framework certificate credential issuance

The required details are as follows:

- **Connection:** to whom the credential is to be issued. By default, “MEDINA SSI Tecnalia Holder TEST” should be selected.
- **Owned schema:** the specific format considered for the credential. By default, medina-ssi (version 1.0).

- **Attributes** (id, status): these will be automatically completed every time new information is received from the *Certificate Lifecycle Manager*³ (LCM), i.e., every time the LCM detects a change on the certificate status.

³ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D4.3 <https://zenodo.org/record/7927231>

4. Certificate Proofs Request (CSP Customer)

The *SSI Framework* provides a potential customer (or other entity) with a graphical interface to asks for proofs of the certificate status of the CSPs. It is available at: <https://medina-webapp.cybersec.digital.tecnalia.dev/>.

The first thing the webapp asks the user to do is to connect to one of the available SSI-agents. In this case, the “Verifier” agent must be selected as shown in Figure 10.

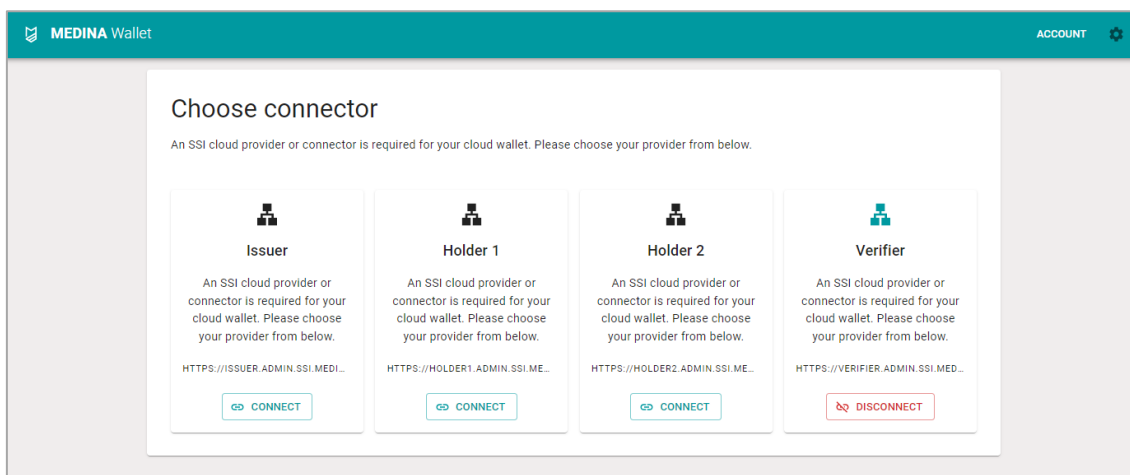


Figure 10. SSI Framework external connection page for verifier

Once connected, the most interesting webapp page for customers is related to the “Certificate Proofs”, where new certificate proofs can be requested through the *Request Proofs* button. Automatically, a form like the one in Figure 11 is displayed.

The screenshot shows the 'Request proofs' form. It has a title 'Request proofs' and a subtitle 'Fill the information below to send a proof request.' The form contains four main sections: 'Connection' with a dropdown menu showing 'MEDINA SSI Tecnalia Holder (a8112e99-97f6-4180-a2a4-8e0cf9722e8b)'; 'Comment' with a text input field containing 'Example of proof'; 'Attributes' with a text input field containing 'certificate_status' and a 'New attribute' link; and 'Conditions' with a table-like structure for adding conditions. The table has columns for 'Name (New condition)', 'Comparison (New condition)' (with a dropdown showing '<'), and 'Value (New condition)'. There are '+' and '-' buttons for adding and removing conditions. At the bottom right, there are 'CANCEL' and 'ACCEPT' buttons.

Figure 11. SSI Framework certificate proofs request

The required details are as follows:

- **Connection:** to whom the proofs will be requested. By default, “MEDINA SSI Tecnalia Holder TEST” should be selected.

- **Comment:** comment for the proof (the reason of the proof could be provided). This is a text parameter.
- **Attributes:** list of the attributes the verifier wants to know about the Cloud Services in the CSP. In MEDINA there are only two attributes (id and status). Any of them (or both of them) should be included. If a different attribute is provided, the process works but the proof will be finally abandoned as the CSP cannot probe the required information.
- **Conditions:** this is an optional parameter related to the ZKP (Zero Knowledge Proof) concept. For example, an attribute such as “postcode” could be considered in this case to indicate a geographical area without disclosing the exact location. This is not really applicable in MEDINA although it can be verified with the id attribute.

Once requested, the new request will be automatically shown in the certificate proofs list on the CSP *SSI Framework* (see section 2.4 for more details), and will also be automatically included in the proofs list on the webapp, obtaining the requested values and indicating the “done” state (shown in green in Figure 12). Additionally, it is also shown as “verified” if the credential has not been revoked.

Certificate Proofs

Ask for proofs and query them in the identity provider **Verifier**.

REQUEST PROOFS

Proof id:41faa0cc-406d-442a-9202-0e46a1910e0d

Created at:2023-07-28T07:20:28.599835Z

Updated at:2023-07-28T07:20:29.194056Z

Req. comment:Example of proof

Pres. comment:auto-presented for proof requests, pres_ex_record: 1b369711-9734-4f4a-b45e-9facc8cc2113

Attributes:certificate_status: done

doneRole: verifierVerified

Figure 12. SSI Framework certificate proofs response

5. Delivery and usage

5.1. Licensing information

This component is offered under Proprietary License. Copyright by TECNALIA.

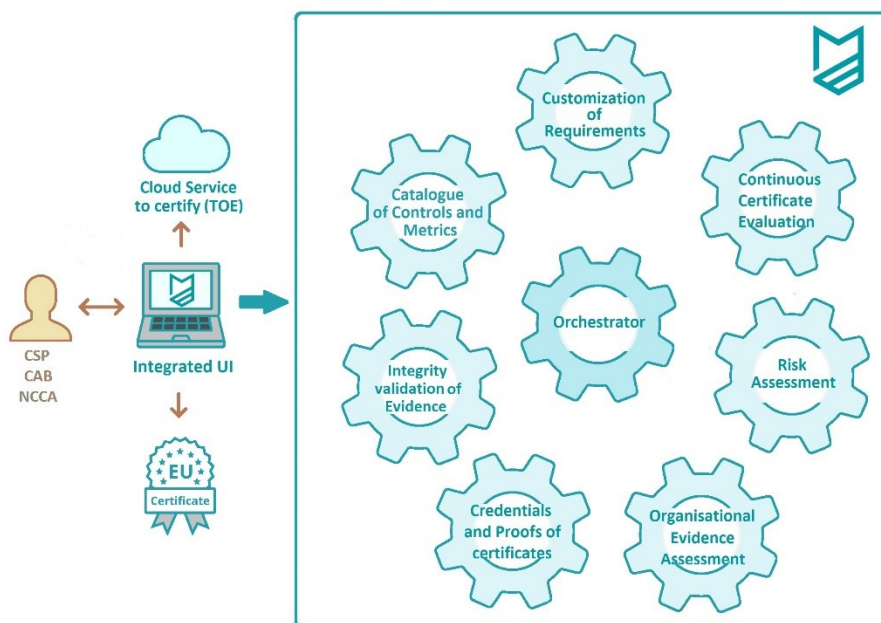
5.2. More information

Interested readers can find more information about the *Self-Sovereign Identity (SSI) Framework* at this link: <https://doi.org/10.5281/zenodo.7927231> “D4.3 Tools and Techniques for the Management and Evaluation of Cloud Security Certifications – v3”.

The MEDINA web site (<https://medina-project.eu/>) also includes several deliverables and blog posts related to the *Self-Sovereign Identity (SSI) Framework*.

Integrity Validation of Evidence

- User Manual -



Project Title:	MEDINA - Security Framework to achieve a continuous audit-based certification in compliance with the EU-wide cloud security certification scheme
Project Number:	952633
Editor:	Cristina Regueiro (Fundación TECNALIA Research and Innovation)
Version:	v1.0
Date:	31.07.2023
Distribution level:	PU



Table of contents

1. Introduction 4

 1.1. User Roles and Permissions 4

2. User Manual 5

 2.1. Toolbar 5

 2.2. List of Evidence 5

 2.3. Evidence 6

 2.4. List of Assessment Results 7

 2.5. Assessment Result 8

3. Blockchain viewer client 11

4. Delivery and usage 13

 4.1. Licensing information 13

 4.2. More information 13

List of Figures

FIGURE 1. MEDINA EVIDENCE TRUSTWORTHINESS MANAGEMENT SYSTEM TOOLBAR.....	5
FIGURE 2. LIST OF EVIDENCE FILTERING OPTIONS.....	5
FIGURE 3. LIST OF EVIDENCE VALIDATION RESULT.....	6
FIGURE 4. EVIDENCE INPUT	6
FIGURE 5. EVIDENCE CORRECT RESULT.....	7
FIGURE 6. EVIDENCE INCORRECT RESULT	7
FIGURE 7. LIST OF ASSESSMENT RESULTS FILTERING OPTIONS.....	8
FIGURE 8. LIST OF ASSESSMENT RESULTS VALIDATION RESULT	8
FIGURE 9. ASSESSMENT RESULT INPUT	9
FIGURE 10. ASSESSMENT RESULT CORRECT RESULT	9
FIGURE 11. ASSESSMENT RESULT INCORRECT RESULT	10
FIGURE 12. BLOCKCHAIN VIEWER CLIENT DASHBOARD	12

1. Introduction

The functionality of Integrity Validation of Evidence in MEDINA is provided by the Blockchain based *MEDINA Evidence Trustworthiness Management System*, which provides a secure mechanism to maintain an audit trail of evidence and assessment results. The *MEDINA Evidence Trustworthiness Management System* is implemented in **Smart Contracts** backbone by a common **Blockchain network** for all the MEDINA framework instances, providing the following **functionalities**:

- Includes the logic for all *Orchestrator*¹ instances in MEDINA to **provide the required information to be audited** (about evidence and assessment results). For this purpose, an API is exposed by the Blockchain client.
- Provides **secure long-term information recording**, thanks to the inherent advantages of Blockchain (integrity, decentralization, authenticity...):
 - It provides a record of information on a verifiable way (**verification**).
 - It provides a record of information on a permanent way (**traceability**).
 - It guarantees resistance to modification of stored data (**integrity**).
- Includes the logic for external users to **access MEDINA's audited information** (about evidence and assessment results) **in a graphical and user-friendly way** through a kibana-based dashboard.
- Includes the logic for **automatic verification of hashes** from currently recorded information on the Orchestrator with hashes recorded on the Blockchain.

1.1. User Roles and Permissions

Access to the *MEDINA Evidence Trustworthiness Management System* is managed by Keycloak². All authenticated users, regardless their roles, can perform “read” actions from the *MEDINA Evidence Trustworthiness Management System* UIs (“write” actions are not allowed through UI). The non-authenticated customer role has no access to any of the UIs.

The table below details which actions are allowed for each of the defined roles:

Roles	Allowed Actions
IT Security Governance	Read information
Security Analyst	Read information
Domain Governance	Read information
Product and Service Owner	Read information
Product (Security) Engineer	Read information
Chief Information Security Office (CISO)	Read information
Customer	None
Auditor	Read information

¹ For more detailed information about this component, the interested reader is referred to the MEDINA Deliverable D3.6 <https://doi.org/10.5281/zenodo.7927225>

² <https://www.keycloak.org>

2. User Manual

2.1. Toolbar

The *MEDINA Evidence Trustworthiness Management System* includes a toolbar (see Figure 1), always accessible in the upper area, with all the options that are available in the tool:

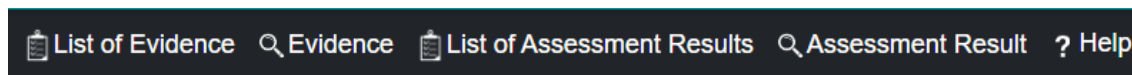


Figure 1. MEDINA Evidence Trustworthiness Management System toolbar

The different menu options, which are described in the following sections, are as follows:

- **List of Evidence:** Provides access to the automatic verification status of the complete list of recorded evidence in the *Orchestrator* comparing their current values with the ones previously recorded on the Blockchain.
- **Evidence:** Provides access to the automatic verification status of a specific piece of evidence defined by its ID.
- **List of Assessment Results:** Provides access to the automatic verification status of the complete list of recorded assessment results in the *Orchestrator* comparing their current values with the ones previously recorded on the Blockchain.
- **Assessment Result:** Provides access to the automatic verification status of a specific assessment result by its ID.
- **Help:** Provides access to the user manual.

2.2. List of Evidence

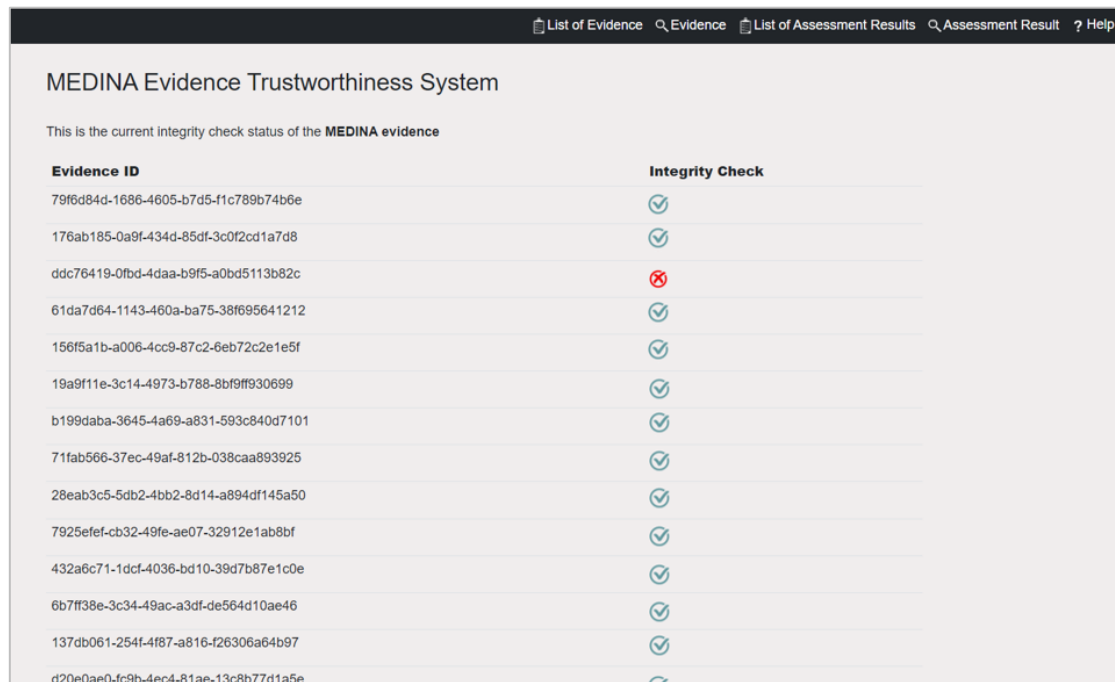
The *List of Evidence* view is useful to automatically obtain the overall evidence integrity status. Figure 2 shows the available optional filtering options to be applied to the list of evidence to be validated: cloud service ID or tool ID can be provided.

Figure 2. List of Evidence filtering options

Once the filtering options have been completed as needed, the user must click on the *Submit* button. As a result, the complete list of recorded evidence on the *Orchestrator* (after filtering data) is displayed, as shown in Figure 3. For each recorded evidence, the “ID” and the “Integrity check” status is shown. The integrity status is automatically obtained; for each piece of evidence, it can be:

- Correct (“green tick”) if the automatically calculated hash of the piece of evidence recorded on the *Orchestrator* matches the hash value previously recorded on the Blockchain.

- Incorrect (“red cross”) if the automatically calculated hash of the piece of evidence recorded on the *Orchestrator* does not match the hash value previously recorded on the Blockchain.

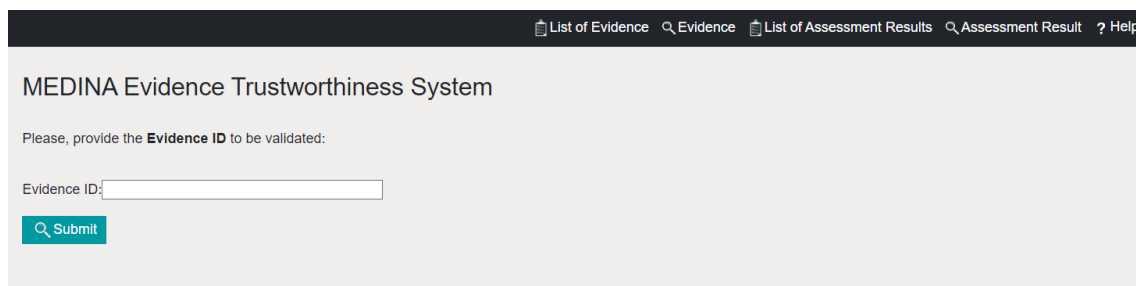


Evidence ID	Integrity Check
79f6d84d-1686-4605-b7d5-f1c789b74b6e	✓
176ab185-0a9f-434d-85df-3c0f2cd1a7d8	✓
ddc76419-0fbd-4daa-b9f5-a0bd5113b82c	✗
61da7d64-1143-460a-ba75-38f695641212	✓
156f5a1b-a006-4cc9-87c2-6eb72c2e1e5f	✓
19a9f11e-3c14-4973-b788-8bf9ff930699	✓
b199daba-3645-4a69-a831-593c840d7101	✓
71fab566-37ec-49af-812b-038caa893925	✓
28eab3c5-5db2-4bb2-8d14-a894df145a50	✓
7925efef-cb32-49fe-ae07-32912e1ab8bf	✓
432a6c71-1dcf-4036-bd10-39d7b87e1c0e	✓
6b7ff38e-3c34-49ac-a3df-de564d10ae46	✓
137db061-254f-4f87-a816-f26306a64b97	✓
d20e0ae0-fc9b-4ec4-81ae-13c8b77d1a5e	✓

Figure 3. List of Evidence validation result

2.3. Evidence

The *Evidence* view allows the automatic comparison of a piece of evidence value with the value recorded on the Blockchain. This option is especially useful when an incorrect integrity check has been obtained in the “List of Evidence” overall evidence integrity check (see Figure 3). Figure 4 shows the form to enter the required piece of evidence ID to be validated.



MEDINA Evidence Trustworthiness System

Please, provide the **Evidence ID** to be validated:

Evidence ID:

Figure 4. Evidence input

Once the evidence ID has been provided, the user must click on the *Submit* button. As a result, the following information is displayed for the specific evidence:

- **Evidence ID:** identifier of the piece of evidence.
- **Orchestrator Hash:** automatically obtained for the evidence currently recorded on the *Orchestrator*. If information related to the specific evidence ID is not available in the *Orchestrator* for any reason, “Not found” will be displayed.
- **Blockchain Hash:** automatically obtained from the Blockchain. If information related to the specific evidence ID is not available in the Blockchain for any reason, “Not found” will be displayed.

Figure 5 and Figure 6 show examples of correct and incorrect evidence integrity checks, respectively.

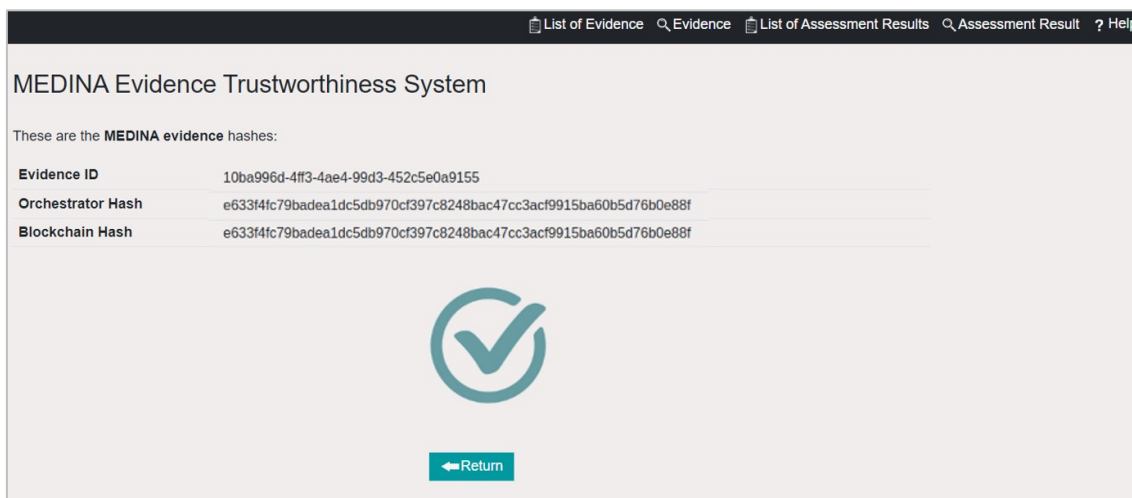


Figure 5. Evidence correct result

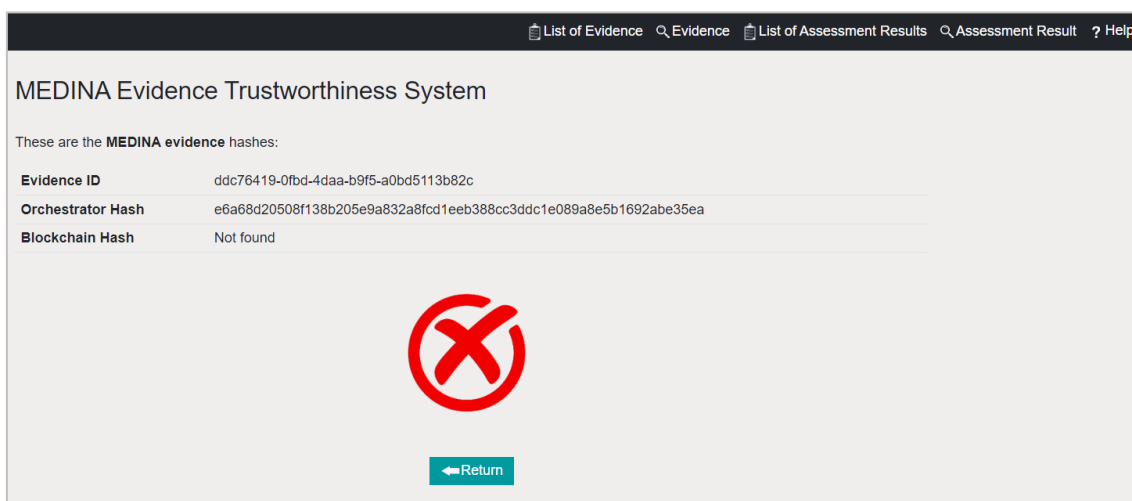


Figure 6. Evidence incorrect result

2.4. List of Assessment Results

The *List of Assessment Results* view is useful to automatically obtain the overall integrity status of the assessment results. Figure 7 shows the available optional filtering options to be applied to the list of assessment results to be validated: cloud service ID or metric ID can be provided. Additionally, it is also possible to filter out only compliant results by clicking on the *Only Compliant results* checkbox.

Once the filtering options have been completed as needed, the user must click on the *Submit* button. As a result, the complete list of recorded assessment results on the *Orchestrator* (after filtering data) is displayed, as shown in Figure 8.

MEDINA Evidence Trustworthiness System

Please, provide the **optional filters** for the integrity check of assessment results recorded on the **MEDINA Orchestrator** against those recorded on the **MEDINA Trustworthiness System**

Only compliant results: ☐

Cloud Service ID:

Metric ID:

Figure 7. List of Assessment Results filtering options

The provided results are the same as for evidence: for each recorded assessment result, the “ID” and the “Integrity check” status is shown. The integrity status is automatically obtained; for each assessment result, it can be:

- Correct (“green tick”) if the automatically calculated hash of the assessment result recorded on the *Orchestrator* matches the hash value previously recorded on the Blockchain.
- Incorrect (“red cross”) if the automatically calculated hash of the assessment result recorded on the *Orchestrator* does not match the hash value previously recorded on the Blockchain.

MEDINA Evidence Trustworthiness System

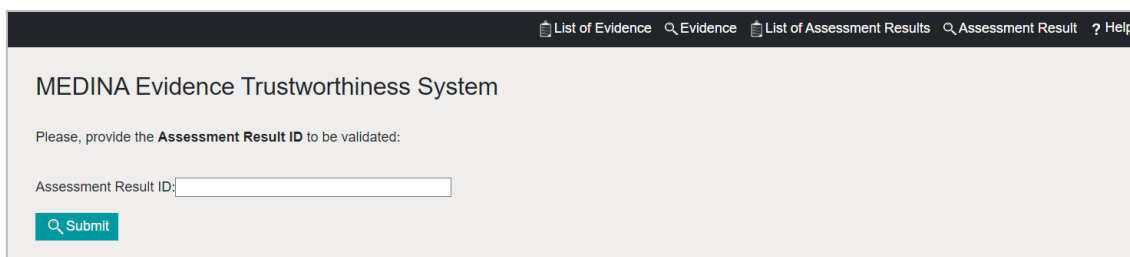
This is the current integrity check status of the **MEDINA** assessment results

Assessment Result ID	Integrity Check
d98fd62-a627-4437-afb7-e1101ad32ade	✓
70a472b0-0ec9-42d7-9be9-93696af85099	✓
99e2c349-8874-4fc8-baac-0753b5750916	✓
f2926efc-2f8f-4d2c-93ca-7c6cfeb1b5fa	✓
94f2c9f3-9c21-4a74-a98d-542d555c7f53	✓
8bd56099-fd91-4c36-aac4-b2b6e9a2cbb6	✓
7922be00-3bed-44d8-8d83-63ae661d401a	✓
90eeaf80-2477-41ba-87e4-0983f7e191a0	✓
81a83e59-5d25-46e6-9cf8-e7b7726962fb	✓
3b57875d-1ad0-4e5e-94bd-b0665360d4a8	✓
e9c2de09-745c-481e-ad62-38fbc12beb88	✓
b4cde218-dfc0-482e-be31-9b2d1cb15f8e	✓

Figure 8. List of Assessment Results validation result

2.5. Assessment Result

The *Assessment Result* view allows the automatic comparison of an assessment result value with the value recorded on the Blockchain. This option is especially useful when an incorrect integrity check has been obtained in the “List of Assessment Result” overall assessment results integrity check (see Figure 8). Figure 9 shows the form to enter the assessment result ID to be validated.



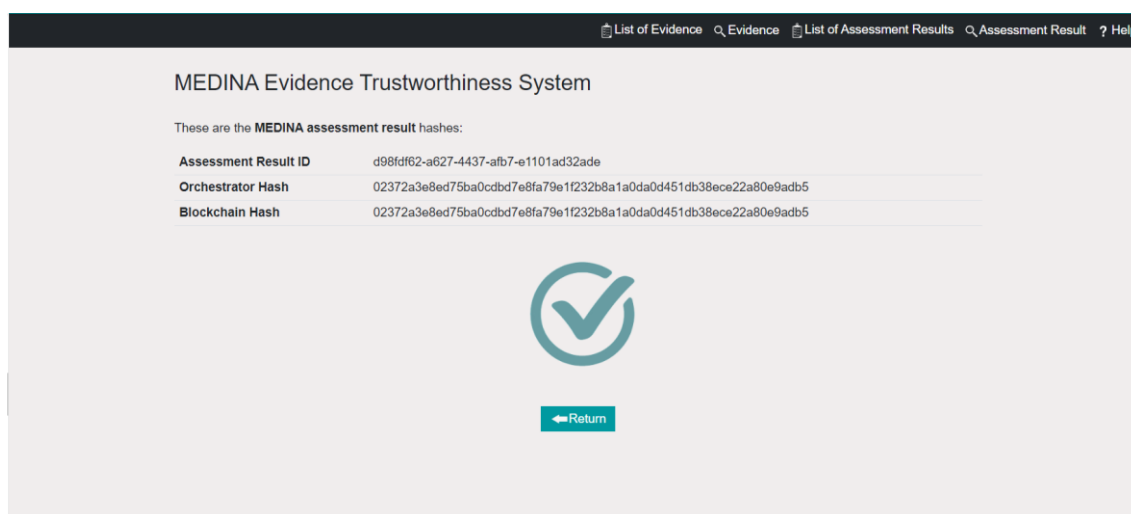
The screenshot shows the top navigation bar with links: List of Evidence, Evidence, List of Assessment Results, Assessment Result, and Help. The main heading is "MEDINA Evidence Trustworthiness System". Below it, a prompt says "Please, provide the **Assessment Result ID** to be validated:". There is a text input field labeled "Assessment Result ID:" and a green "Submit" button with a magnifying glass icon.

Figure 9. Assessment Result input

Once the assessment result ID has been provided, the user must click on the *Submit* button. As a result, the same information as for the evidence is shown:

- **Assessment Result ID:** identifier of the assessment result.
- **Orchestrator Hash:** automatically obtained for the assessment result currently recorded on the *Orchestrator*. If information related to the specific assessment result ID is not available in the *Orchestrator* for any reason, "Not found" will be shown.
- **Blockchain Hash:** automatically obtained from the Blockchain. If information related to the specific assessment result ID is not available in the Blockchain for any reason, "Not found" will be shown.

Figure 10 and Figure 11 show examples of correct and incorrect assessment result integrity checks, respectively.



The screenshot shows the top navigation bar with links: List of Evidence, Evidence, List of Assessment Results, Assessment Result, and Help. The main heading is "MEDINA Evidence Trustworthiness System". Below it, a prompt says "These are the **MEDINA assessment result hashes**:". There is a table with three rows: "Assessment Result ID", "Orchestrator Hash", and "Blockchain Hash", each with a corresponding hash value. Below the table, there is a large green checkmark icon and a green "Return" button with a left arrow icon.

These are the MEDINA assessment result hashes :	
Assessment Result ID	d98fd62-a627-4437-afb7-e1101ad32ade
Orchestrator Hash	02372a3e8ed75ba0cdbd7e8fa79e1f232b8a1a0da0d451db38ece22a80e9adb5
Blockchain Hash	02372a3e8ed75ba0cdbd7e8fa79e1f232b8a1a0da0d451db38ece22a80e9adb5

Figure 10. Assessment Result correct result

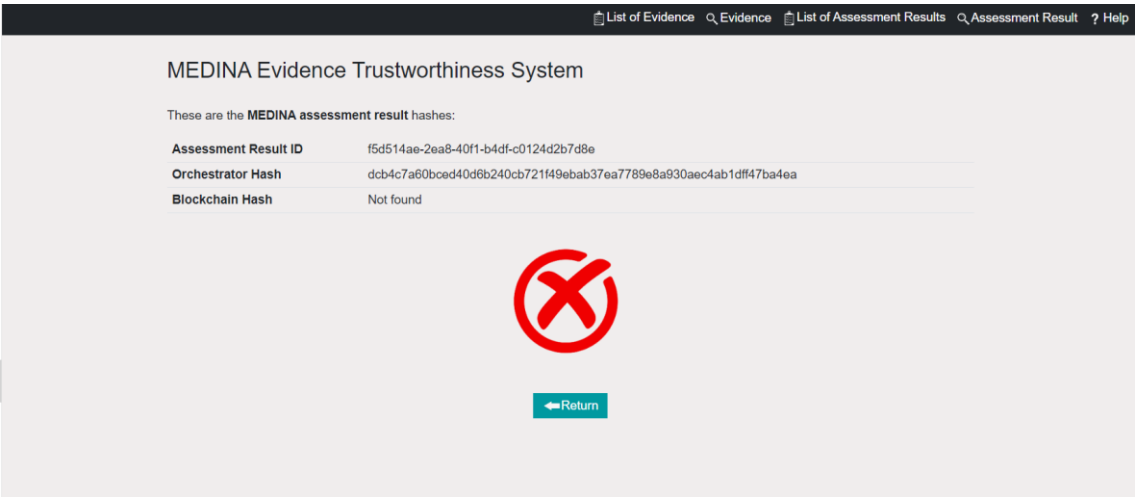


Figure 11. Assessment Result incorrect result

3. Blockchain viewer client

The *MEDINA Evidence Trustworthiness Management System* also exposes a Kibana-based graphical interface available at: <https://kibana.medina.bclab.dev/> [internal use only - authentication is required] for manual graphical access to the information recorded on the Blockchain.

Figure 12 shows the main dashboard for each *Orchestrator* instance. Here, the complete list of registered evidence (evidence hashes) and assessment results (assessment result hashes) is shown. This information is useful for manual verifications. Additionally, a summary of the total number of registered evidence and assessment results for the specific *Orchestrator* instance is shown.

For evidence, the following information is shown:

- Evidence ID
- Evidence Hash
- Resource
- Evidence Collector
- CSP
- *Orchestrator* timestamp (when evidence was received in the *Orchestrator*).
- Blockchain timestamp (when evidence was recorded on the Blockchain).

For assessment results, the following information is shown:

- Assessment Result ID
- Assessment Result Hash
- Metric
- Associated evidence
- *Orchestrator* timestamp (when assessment result was received in the *Orchestrator*).
- Blockchain timestamp (when assessment result was recorded on the Blockchain).

Finally different filters have been included for improving the usability of the system: filter by id, hash or associated metadata on the evidence and assessment results.

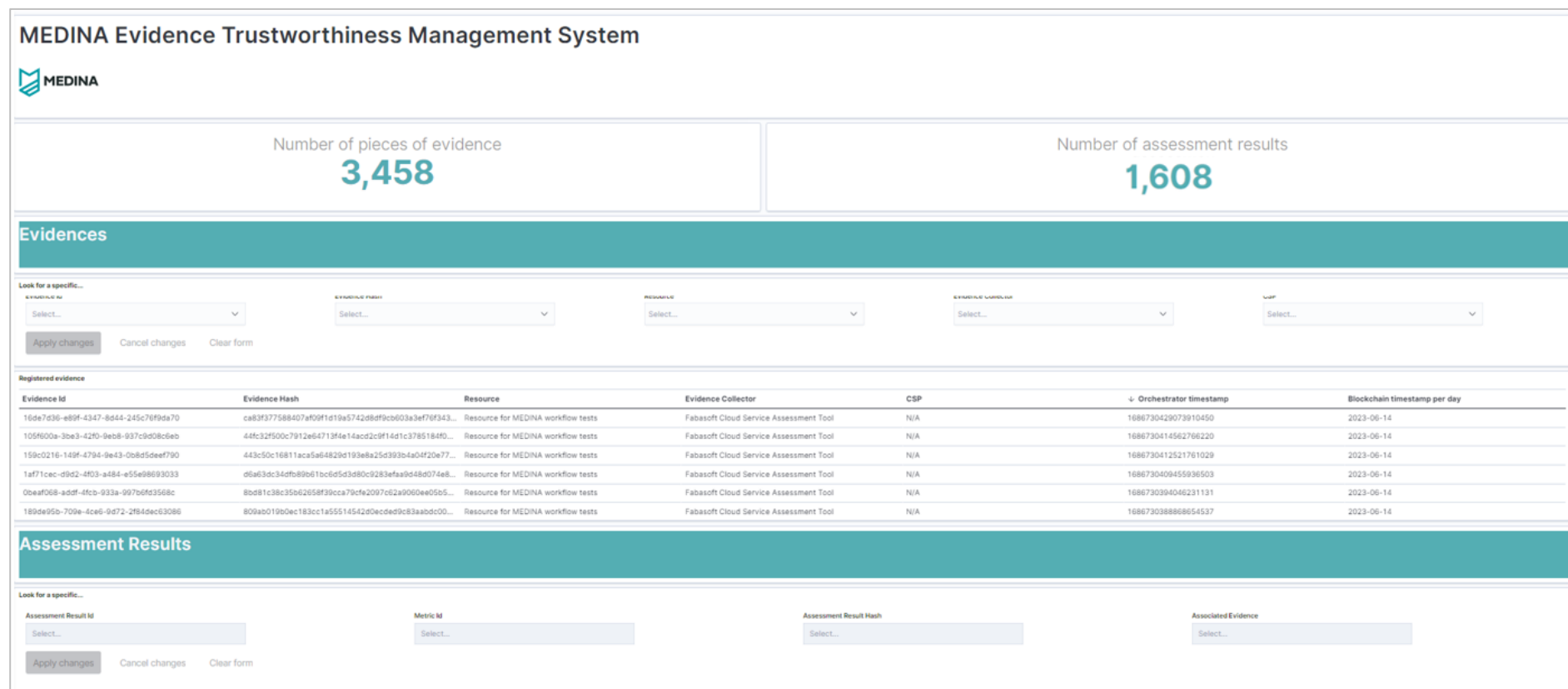


Figure 12. Blockchain viewer client dashboard

4. Delivery and usage

4.1. Licensing information

This component is offered under Proprietary License. Copyright by TECNALIA.

4.2. More information

Interested readers can find more information about the *MEDINA Evidence Trustworthiness Management System* at this link: <https://doi.org/10.5281/zenodo.7927220> “D3.3 Tools and techniques for the management of trustworthy evidence - v3”.

The MEDINA web site (<https://medina-project.eu/>) also includes several deliverables and blog posts related to the *MEDINA Evidence Trustworthiness Management System*.