MEDINA

Deliverable D4.5

Methodology and tools for risk-based assessment and security control reconfiguration - v2

Editor(s):	Artsiom Yautsiukhin
Responsible Partner:	National Research Council CNR
Status-Version:	Final – v1.0
Date:	30.04.2023
Distribution level (CO, PU):	PU

Project Number:	952633
Project Title:	MEDINA

Title of Deliverable:	Methodology and tools for risk-based assessment and security control reconfiguration - v2
Due Date of Delivery to the EC	30.04.2023

Workpackage responsible for the Deliverable:	WP4 - Continuous Life-Cycle Management of Cloud Security Certifications
Editor(s):	Artsiom Yautsiukhin (CNR)
Contributor(s):	Artsiom Yautsiukhin (CNR) Immanuel Kunz (FhG)
Reviewer(s):	Mika Leskinen (NIXU) Cristina Martínez (TECNALIA)
Approved by:	All Partners
Recommended/mandatory readers:	WP2, WP4, WP5

Abstract:	This deliverable comprises the methodology as well as the prototype implementation of the risk-based auditor component. To follow the approach taken in other tasks, there will be three iterations of the tool integration, an initial prototype, showcasing the methodology, a second release, which will be based on a refinement of the technical architecture and finally the third iteration, which will reflect the implementation of the use cases. This deliverable is the second and final version reporting results of Task 4.4.
Keyword List:	Risk Assessment, Compliance, Non-conformity degree, Dynamic risk-based non-conformity assessment
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
v0.3	01.04.2023	First draft version	Artsiom Yautsiukhin (CNR)
v0.7	15.04.2023	Final draft version	Artsiom Yautsiukhin (CNR) Immanuel Kunz (FhG)
v0.8	24.04.2023	Comments and corrections	Mika Leskinen (NIXU)
v0.9	27.04.2023	Comments addressed	Artsiom Yautsiukhin (CNR) Immanuel Kunz (FhG)
v1.0	30.04.2023	Ready for submission	Cristina Martínez (TECNALIA)

Table of Contents

List of Tables

List of Figures

Terms and Abbreviations

AMOE	Assessment and Management of Organizational Evidence
API	Application Programming Interface
DBMS	Data Base Management System
CCD	Company Compliance Dashboard
CCE	Continuous Certification Evaluation
CIA	Confidentiality, Integrity, and Availability
CSS	Cascading Style Sheets
CSP	Cloud Service Provider
EC	European Commission
EUCS	European Cybersecurity Certification Scheme for Cloud Services
GA	Grant Agreement to the project
GUI	Graphical User Interface
IaaS	Infrastructure as a Service
JSON	JavaScript Object Notation
JSP	Jakarta Server Pages
LCM	Life-Cycle Manager
HTML	Hypertext Mark-up Language
OS	Operating System
PaaS	Platform as a Service
RAOF	Risk Assessment and Optimisation Framework
REST	Representational State Transfer
SaaS	Software as a Service
SATRA	Self-Assessment Tool for Risk Analysis
ToE	Target of Evaluation
ToC	Target of Certification
TOM	Technical and Organizational Measure
UUID	Universally Unique Identifier
VM	Virtual Machine
WF	Workflow
WP	Work package

Executive Summary

This deliverable reports the final findings of task 4.4 that is dedicated to the *dynamic* risk-based assessment and security control configuration. In the scope of this task, we develop and implement a risk-based approach for assessment of non-conformity with a selected certification schema. The risk-based approach allows evaluating the deviation from the complete compliance focussing on protecting the most sensitive assets from the likely threats. Such an approach will help the CSP to focus on its concrete needs, justify the distribution of security effort and decrease cyber risks for CSP's customers (e.g., since the customer's trust is one of the assets in our risk computational model[1]).

The deliverable first focusses on the description of the overall approach, which is based on our risk computational model reported in D2.8 [1]. Section 2 describes in detail the methodology for our risk-based assessment, which allows automating our approach. Now, the direct human intervention (i.e., the reliance of the tool on manually provided input) is required only during the initial asset assessment. After that, the continuous monitoring phase does not require human to be involved in the loop and the whole process becomes automatic.

In section 3, we show how the dynamic risk-based non-conformity assessment functionality can be integrated into the MEDINA framework, describing which elements of the framework are involved in the non-conformity assessment process and which data should be exchanged for its execution. We note that the dynamic risk-based non-conformity assessment is implemented as a part of our *Risk Assessment and Optimisation Framework*. In this deliverable, although we repeat some concepts reported in D2.8 [1], we dedicate our attention to the dynamic usage of our risk assessment approach.

We provide the final version of the delivery and usage descriptions of our tool, called *Self-Assessment Tool for Risk Analysis* (SATRA), which implements the RAOF functionalities (section 4). This section is very close to D2.8 [1], since it is dedicated to the description of the same tool. Nevertheless, we do our best to focus on the new functionalities and report the most up-to-date information.

Finally, section 5 outlines all improvements made in SATRA with respect to the dynamic risk assessment, limitations of the tool and approach, and possible future evolutions.

This is the final version of the deliverable reporting the results of task T4.4. This document extends D4.4 [4] and focusses on the additional modifications in the approach and integration with other MEDINA components. In particular, a possibility to consider specific resources (i.e., with different sensitivity level than other resources of the same type) has been added. The procedure for computing risk per resource has been slightly updated, considering a specific resource type ("PolicyDocument"), which allows monitoring organisational procedures and practices relevant for the service as a whole. In order to support reconfiguration of the implemented security controls, a new functionality helping to prioritise detected non-conformities have been added. This procedure should support the CSP in identifying the most influential failures and in focusing its effort on fixing them.

[1] By the risk computational model, we mean a detailed approach for computation of cyber risks. This model includes threat and risk models, which are used to drive the computation.

BY-SA

1 Introduction

The main focus of MEDINA is on the continuous monitoring of compliance with a selected certification schema (and chosen assurance level). During this phase, a cloud service is to be monitored by means of applied metrics and the results of these measurements are used in order to decide whether the service could maintain its certificate, or the certificate should be revoked.

Naturally, once the evaluation of every metric succeeds the certificate should be maintained. Such situation is ideal, but the real life shows that deviations are frequent. Some of such deviations could be temporary, due to the dynamic nature of the cloud environment (e.g., a new virtual machine could be added and destroyed within a period of several minutes). Other deviations could be long living, but they may relate to a very insignificant asset, which is not secured properly simply because it cannot rise a severe security problem (e.g., non-private data may be sent through an open channel). Although, the later indicates a failure to fulfil strictly all requirements of the selected certification schema, such non-conformities are insignificant and should not lead to the revocation of a certificate.

One way to evaluate non-conformities and to decide which of them could be considered as minor (insignificant for revocation) and which ones should be taken into account seriously (e.g., lead to certificate suspension or revocation) could be to empower the decision-making procedure with a risk-based assessment of non-conformities.

1.1 About this deliverable

This deliverable is dedicated to the description of our approach to the *dynamic* risk-based non-conformity assessment and implementation of this approach with a tool in scope of the MEDINA framework.

Our approach is based on the risk computation model reported in D2.8 [1]. On the other hand, the main communication channel used by the static risk-based support described in D2.8 is a GUI, since it is assumed to interact with a compliance manager. In this deliverable we focus on dynamic, and, thus, automatic operation of our risk-based assessment, which, in its turn, means that human involvement in the process should be minimised. Our dynamic risk-based assessment of non-conformity requires interaction with the compliance manager only in the set-up phase, during which the compliance manager should define expected sensitivity levels for all assets, letting the later assessment to be fully automatic during the continuous monitoring.

We should underline once again that our dynamic risk-based assessment is not completely independent since it uses the same core computational model as the static one. On the other hand, this approach ensures that the computation made in both phases is very similar, and that the results of the non-conformity assessment received during the static phase are to be confirmed during the continuous monitoring.

Thus, even though this report has much in common with D2.8, it focusses on the dynamic functionality, and (in some cases) repeats what has already been stated in D2.8 only for providing a complete picture. This deliverable is also linked with other deliverables from WP4 (e.g., D4.3 [5]) as the described assessment is an integral part of the certificate evaluation process.

This document is the final report summarising the effort done in scope of T4.4 dedicated to the development of the dynamic risk-based non-conformity assessment approach and supporting tools.

1.2 Document Structure

The document is structured as follows. Section 2 describes the methodology for the risk-based non-conformity assessment. This section explains how, when, and why this assessment is required for a continuous monitoring of certification and provides the details on how this assessment is realised in the scope of MEDINA. Section 3 describes how the dynamic functionality of the *Risk Assessment and Optimisation Framework* (RAOF) is designed and is integrated into the MEDINA framework. Section 4 provides the recent updates of the delivery and usage of RAOF. Section 5 lists the improvements of SATRA in scope of MEDINA, discusses the limitations of the approach and possible directions for the tool to evolve. Finally, section 6 concludes the report.

1.3 Updates from D4.4

This deliverable is an updated version of D4.4 [4] and most of its content remains as it was in D4.4 (with some changes), allowing D4.5 to be self-contained. For simpler tracking of progress and updates with regards to the previous deliverable version, Table 1 gives a brief overview of changes and additions to each of the document sections.

Table 1. Overview of deliverable updates with respect to D4.4

Section	Changes
2	Modifications to the dynamic risk computational procedure related to: • Considering specific resources • Considering the “PolicyDocument” resource type New functionality evaluating contribution of failed evaluations.
3	Various small updates to report the latest modifications in the supporting tool (SATRA). The component card of the RAOF component has been included.
4	Technical details have been updated.
5	New section including the progress made in the scope of T4.4, limitations and future work.
6	Conclusions are aligned.
Appendix A	Appendix already present in D4.4, showing an example of the input JSON file sent by CCE to RAOF
Appendix B	New Appendix showing the RAOF Sequence Diagram.

(cc) BY-SA

2 A Methodology for Dynamic Risk-based Assessment and Security Control Recommendations

This section describes our approach to apply risk-based decision making into the process of a certificate status evaluation. This approach is based on the core risk assessment procedure reported in D2.8 [1] and applies it for the continuous monitoring and evaluation process. The dynamic risk assessment differs from its static counterpart in a number of aspects, including different modelling assumptions, changed approach to provisioning the required input and output values, modified workflow in risk calculation, etc. All these aspects will be covered and described in this section.

2.1 Dynamic Risk-based Support during the Continuous Monitoring of MEDINA

As stated above, the main goal of applying risk assessment during the continuous monitoring phase is to use its results for a more CSP-oriented certificate evaluation. Such an approach is more CSP oriented comparing to many others because it weights the requirements imposed by a selected certification schema with the needs of the CSP (i.e., focusing on protecting the most important assets). Therefore, following a risk-based approach should make the MEDINA platform more flexible, more security focused, and more attractive for CSPs.

Similar to the static risk assessment, a dynamic risk assessment model should be based on identification and evaluation of the same three components: *assets*, *threats,* and *vulnerabilities*. Moreover, it is required to align static and dynamic risk assessments in a way that they return the same (or very close) results if the same input parameters are provided. In other words, the dynamic and static risk assessments must be based on the same computational model, with the changes that mostly affect the way the parameters are provided and avoid elements which may significantly change the result of the computation. What is important to achieve is the assurance that in case of correct[2] manual provisioning of the input parameters during the static risk assessment, the result (i.e., non-conformity assessment decision) will be the same as during the dynamic risk assessment with input parameters automatically collected by the verification tools.

On the other hand, in contrast to the static risk assessment, the dynamic risk assessment must be automatic, i.e., it should be executed without human intervention. Automation of the risk assessment process requires automatic provisioning of the changing information, its automatic processing and propagation to the decision-making engine. The latter is mostly an implementation problem (instead of displaying the results though GUI, the risk assessment tool should provide it though API). But the change in the type of the provided information requires re-evaluation of initial assumptions and ensuring their correct processing by the risk computation engine. This step highly depends on the sources of the information, which are able to detect and evaluate all risk components (e.g., all assets and their sensitivity levels).

The focus of the conducted risk assessment is a Target of Evaluation (ToE), i.e., the cloud service to be assessed and the certification scheme (together with the target assurance level) against which the service is assessed. The required input for the risk assessment is the information about the main assets of the cloud and the fulfilment of the security requirements defined by the selected certification schema. It is assumed that these two types of input may change in time (e.g., new VMs could be added to the service, or an insecure protocol could be temporary used for transferring data). This information is to be collected by the *MEDINA Evidence Management*

[2] Here by “correct” inputs we mean genuine answers provided by the analyst (e.g., compliance manager) which accurately represent the real state of the service (i.e., existing assets and implemented security features).

cc BY-SA

Tools and aggregated and pre-processed by an evaluation unit, the *Continuous Certificate Evaluation* component (CCE) in MEDINA[3], pre-processed and provided to the risk assessment tool via a dedicated API. To overcome the difficulty with dynamic evaluation of the sensitivity of the assets, these values are estimated before starting the continuous monitoring for all possible asset types.

Unfortunately, some information can only be provided by a human. First, this is the information about the selected certification scheme and the target assurance level. This information is required to set up the risk computational model for further operation and must be provided by the human operator (e.g., compliance manager) before the continuous monitoring starts operating.

Another piece of information which is not possible to collect with automatic means is the sensitivity of the available assets. In the scope of the task under consideration, assets are those of cloud resources which, once compromised, cause the major damage to the CSP The resource types selected for the computational model are aligned with the Cloud Resource Ontology developed by Fraunhofer (a partner of the project). These resource types have been filtered to focus on 14 types which are potentially most sensitive (in order to simplify the work for a compliance manager). The component reporting the evaluation results (i.e., CCE) to RAOF uses the mentioned ontology to define for which resource an evaluation has been performed. We refer the interested reader to D2.8 [1], which lists all supported resource types, how they are mapped to the limited list of resource types used for risk computation (i.e., asset types), and briefly describes the mentioned ontology. In short, in the scope of this document, the terms "asset" and "resource" are used interchangeably, but we always consider only the potentially sensitive resources.

Since resources could be changed over time, before the continuous monitoring starts, the operator is asked to provide sensitivity values for all possible types of resources. The sensitivity values represent the expected damage due to compromise of Confidentiality, Integrity, and Availability ("CIA impact" for short) of the considered resource. The values for specific resources will be assigned depending on their type on the fly. The result of execution of our computation model (and supporting tool) is the assessment of non-conformity, which could be either *major* or *minor*. In other words, risk assessment should be used only if a non-conformity is detected. The results of the risk-based non-conformity assessment are provided to the engine responsible for evaluation of the certificate, the *Automated Certificate Life-Cycle Manager*, which will make its decision using its internal logic about the state of the certificate (continue, suspend, revoke, etc.). This process is considered in detail in deliverable D4.3 [5].

In addition, the risk-based approach for analysis of non-conformities can be used to evaluate the importance of the detected failures of the considered requirement. Every failed requirement is to be analysed separately to determine how much it contributes to the overall deviation of risk level from the complete conformity. This value could serve as an indicator of importance for the CSP in order to prioritise the effort on fixing the detected non-conformities.

2.2 A Methodology for Risk-based Assessment during the Continuous Monitoring of MEDINA

Our methodology for risk-based non-conformity assessment during the continuous monitoring is grounded on the basic risk assessment computational model described in D2.8 [1], and has the main focus on pre-processing and preparing the required inputs and specific usage of the

[3] *MEDINA Evidence Management Tools* and CCE are developed in the scope of WP3 and WP4 and reported in D3.6 [9] and D4.3 [5] correspondingly.

mentioned computational model. Thus, in this deliverable we are not going to repeat the basic functionality of the model but focus on the methodology for its usage during the continuous monitoring. The main elements and steps of this methodology are indicated in Figure 1.

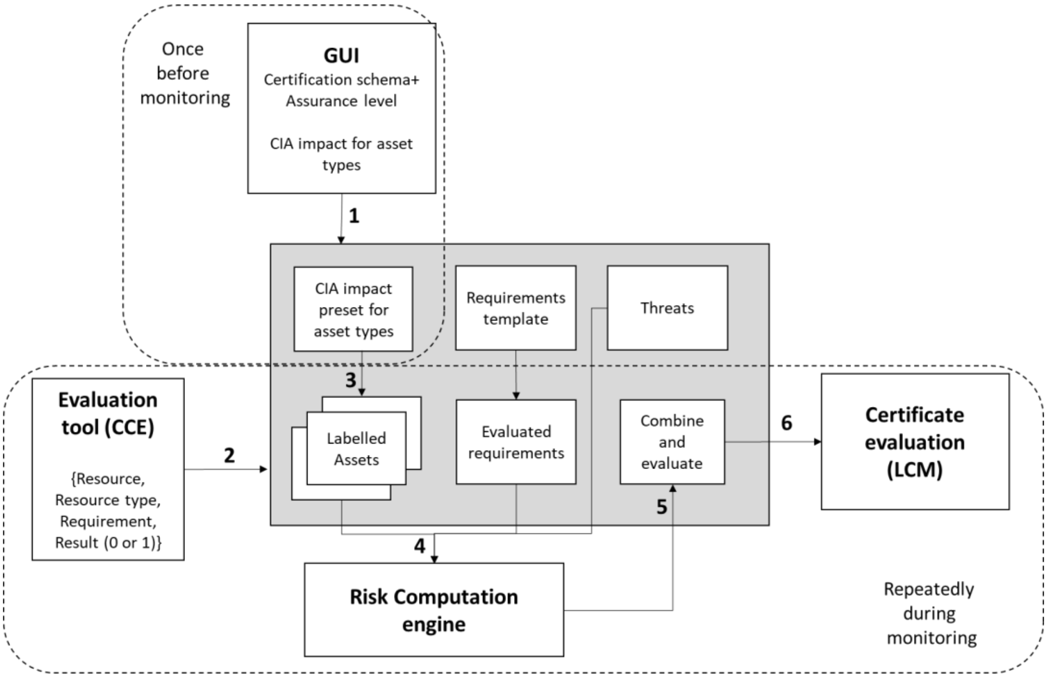

Once before monitoring
GUI
Certification schema+ Assurance level
CIA impact for asset types
1
CIA impact preset for asset types
Requirements template
Threats
3
Evaluation tool (CCE)
{Resource, Resource type, Requirement, Result (0 or 1)}
2
Labelled Assets
Evaluated requirements
Combine and evaluate
6
Certificate evaluation (LCM)
5
4
Risk Computation engine
Repeatedly during monitoring

Figure 1. Dynamic risk-based non-conformity assessment workflow

The input for the dynamic risk-based assessment is provided during two different phases: before starting the continuous monitoring (done once) and during this phase (periodically). The process for the analysis of reported non-conformities is as follows.

Step 1. Set up the model/tool for continuous usage

The first step of the methodology is to set up the computational model for its usage during the continuous monitoring. By "setting up the model" we mean setting up the elements of the model related to the selected certification scheme and selected assurance level, such as requirements and controls. This is required because, in theory, the user is allowed to select a scheme (and an assurance level) to comply with. On the other hand, in the scope of the MEDINA project, only one certification scheme (EUCS [4]) and one assurance level ("High") are supported. Yet, to describe the potential operation of our computational model (and the supporting tool), it is considered that a schema and an assurance level can be chosen by the CSP.

Another parameter, which is required to set up the computational model is the Cloud Service type (SaaS, PaaS, or IaaS). This type is required to identify threats more typical for the considered service.

The selection of a certification scheme, an assurance level, and a cloud service type drives the automatic setting up of the following parameters:

- a list of requirements (defined as *R* in D2.8 [1])
- the associated list of Boolean values *RV* for all requirements, which denotes fulfilment (1) or not fulfilment (0) of the corresponding requirement
- a list of controls *C*
- the mapping matrix ***RC***, in which every element denotes the degree up to which a requirement *r* contributes to control *c*

- the mapping matrix ***RT,*** with cells denoting the probability for a security control *c* to prevent a threat *t*, obtained from various statistical sources
- weight vectors $W_{C'}$ and $W_{C''}$ denoting the degree to which management controls affect the power of the controls directly preventing threats from occurrence.

All these values are already defined in our supporting tool and do not require any modifications from the user.

The second part of this step is to pre-set the CIA impact values for assets. Since it is assumed that the main resources of the service may vary in time, the expected impact values could be set up only for resource types (e.g., Virtual Machine, Database, etc.). Once the values are set up, the risk assessment tool is able to assign these values to concrete resources once they are detected and reported to the risk assessment engine. This operation can be done automatically during the continuous monitoring phase.

In short, for all asset/resource types AT, we ask the user to define the estimated impact in case Confidentiality, Integrity or Availability is compromised, obtaining vectors AT_C, AT_I, AT_A, which are to be used in the next steps to define vectors A_C, A_I, A_A (i.e., CIA impact) for concrete resources, vectors which are required as input by our computation model. Moreover, in order to treat more precisely the most important resources, it is allowed to set up impact values A_C, A_I, A_A for these specific resources directly, using their IDs.

Step 2. Provisioning of the monitored data

Risk assessment is periodically used to assess the detected non-conformity. Obviously, if no deviation from the certificate requirements is detected, conducting the risk-based assessment is redundant.

Once a non-conformity is detected, the required input parameters should be provided for risk assessment. We are not going to discuss how this information is collected, as it is the topic considered in other MEDINA deliverables such as D4.3 [5]. The main information required, and which is going to be provided as an input for risk assessment is a set of the tuples[4] containing the following information[5]:

- Resource ID
- Type of the resource
- Requirement ID
- Requirement evaluation status: fulfilled (1) or not fulfilled (0).

Using this information, the engine is able to single out a set of requirements $R^r \subseteq R$ evaluated in relation to a reported resource r.

In addition to this information, the evaluation tool provides the information about the considered service (UUID).

Among all resource types, there is one special resource type, called "PolicyDocument". This resource type is monitored by AMOE tool (see D3.6 section 5 [7] for its description) and is used for reporting evaluation results linked to analysis of various documents (e.g., in form of pdf files). These documents relate to overall procedures followed by the CSP (e.g., presence of cyber

[4] A tuple is a finite sequence of elements.

[5] In real data exchange (between CCE and RAOF) a more complex structure (Evaluation Result) is used, which contains more information about the evaluation, but this data is not relevant for the discussion in this document.

security policies or signing of non-disclosure agreements), i.e., they support requirements relevant for the *service as a whole*, rather than for specific resources only. In short, evaluations with resource type "PolicyDocument" will be used for the assessment of *all* assessed resources.

Step 3. Preparation of input parameters for risk assessment

Once a request for a risk-based assessment of a non-conformity is received, the provided input should be pre-processed for the computational engine to perform the assessment. The following actions are to be performed to prepare input parameters for risk assessment.

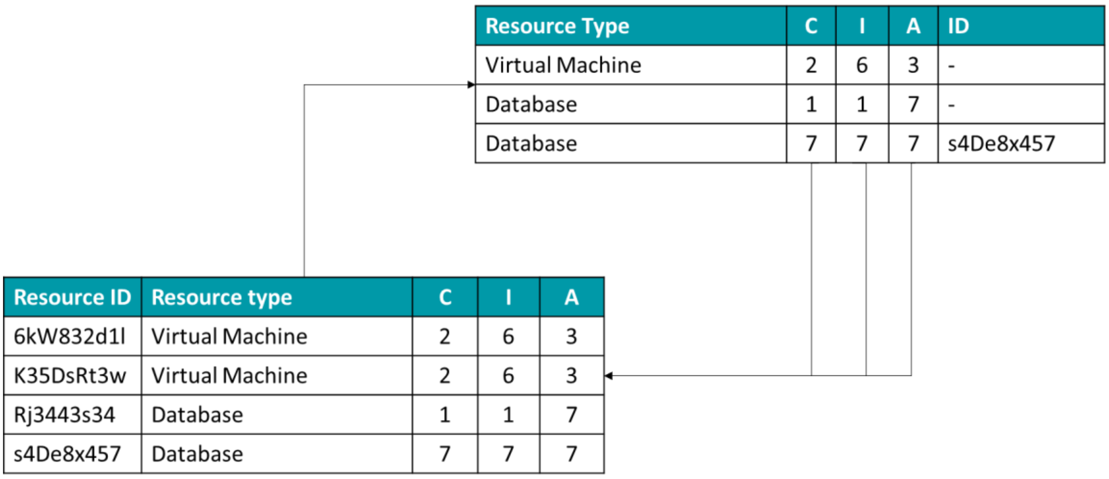
First, the resource types are mapped to the limited list of types used by the computational model, according to the following mapping (copied from D2.8 [1] for consistency):

Table 2. Mapping of resources from the Fraunhofer's ontology and resources used in our risk assessment.

FhG Resource types (reported)	Resource/asset types (used for risk assessment)
Account	---
Job	CI CD Service
Workflow	CI CD Service
Container	Container
Function	Function
Virtual Machine	Virtual Machine
ContainerOrchestration	ContainerOrchestration
ContainerRegistry	ContainerRegistry
Identity	---
RoleAssignment	---
Container Image	Image. Container Image
VMImage	Image. VM Image
DeviceProvisioningService	IoT. Device Provisioning Service
MessagingHub	IoT. Messaging Hub
NetworkInterface	Network
NetworkSecurityGroup	Network
VirtualNetwork	---
VirtualSubNetwork	---
DocumentDatabaseService	Database
KeyValueDatabaseService	Database
RelationalDatabaseService	Database
LoadBalancer	---
LoggingService	---
ObjectStorageService	---
PasswordPolicy	---
BlockStorage	local storage
FileStorage	local storage
ObjectStorage	local storage
DatabaseStorage	local storage
---	CSC trust

Second, a list of assets is formed. For every reported tuple, distinct Resource IDs are extracted which form the list of assets *A*. For every Resource ID there are two ways to set up the impact values A_C, A_I, A_A:

1. If a considered Resource ID has the impact values A_C, A_I, A_A defined in step 1, then, these values are used.
2. If a considered Resource ID was not explicitly set up in step 1, the associated resource type is used in order to retrieve the pre-defined impact values (AT_C, AT_I, AT_A) and assign the corresponding impact types in case confidentiality A_C, integrity A_I, and/or availability A_A is violated.



Resource Type	C	I	A	ID
Virtual Machine	2	6	3	-
Database	1	1	7	-
Database	7	7	7	s4De8x457

Resource ID	Resource type	C	I	A
6kW832d1l	Virtual Machine	2	6	3
K35DsRt3w	Virtual Machine	2	6	3
Rj3443s34	Database	1	1	7
s4De8x457	Database	7	7	7

Figure 2. Assigning impact to discovered resources using pre-set values for resource types

The core difference with the static risk assessment approach is that using monitoring functionality, it is possible to identify how certification requirements are addressed for every resource. We remind that for the static risk assessment, fulfilment of requirements is assessed for all resources together. Considering different security controls applied to different resources can be also done for static assessment, but then, it would require answering a long questionnaire for every resource separately, which makes the service hardly usable. On the other hand, such a tedious operation could be performed by a tool with automatic input provisioning.

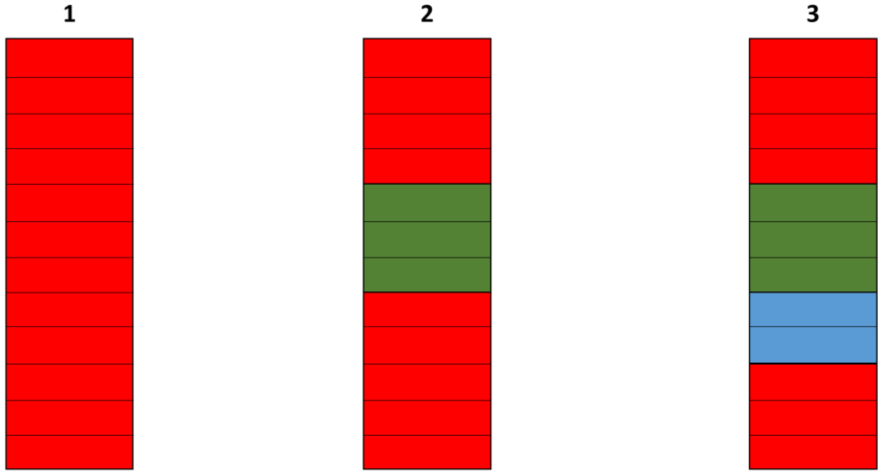
With the inputs provided for analysis, it is possible to perform a risk assessment for every resource separately, considering satisfaction for every resource. Thus, we are able to sense different risks in case one virtual machine has a malware protection and another one does not.

Unfortunately, it is often impossible to obtain information about the assessment of *all* requirements for every asset. This may happen because of many reasons: some requirements may have no assessment methods and metrics for assessment (i.e., the monitoring system cannot neither confirm nor disprove satisfaction of some requirements); some metrics could not be used by a CSP; or some requirements should be measured for different resource types (i.e., the strong encryption of the communication channel cannot be checked for a database).

There could be different approaches on how to determine the values for the requirements which are not directly measured for a resource. In the scope of MEDINA, we form a list of requirements as follows:

1. For every Resource ID reported to RAOF, a full list of requirements is considered. All these requirements are pre-set as satisfied.
2. For every evaluation result related to the considered resource, set the monitored requirement to the corresponding state.
3. For every evaluation result related to "PolicyDocument" resource type, set the reported requirement to the corresponding state. If several evaluation results with this resource type for the same requirement (i.e., with different resources) are available, then the

requirement is considered as failed if at least one reported evaluation result has the failed status.



Set all requirements for the considered resource as satisfied by default

Set the requirements reported in the provided evaluation results linked with the considered resource as reported

Set the requirements reported in the provided evaluation results linked with PolicyDocument resource type as reported

Figure 3. The process of setting the vector of requirement values (RV) per resource

The reasoning behind the setting of not reported requirements to 1 (satisfied) is as follows. First, the requirements which are not monitored are impossible to track. In theory, we may ask the CSP to declare the status of all requirements during the static assessment, but, since the focus of MEDINA is on continuous monitoring with as much automation (i.e., as less human involvement) as possible, this step is only optional in the overall MEDINA process. Moreover, if a user does implement any monitor facility for a requirement, the MEDINA framework has no means to verify it and a CSP may declare any status of these requirements as it likes. Thus, here we assume that all relevant requirements are monitored and the information about them is provided to RAOF for the analysis. Non monitored requirements are considered as not-relevant (and, thus, can be assigned to 1). This is the most realistic approach we can follow, considering the available means and the overall setting of the MEDINA project. We acknowledge that for real application such an assumption could be too strong. In that case, additional procedural (e.g., an obligation to monitor all requirements) or technical (additional monitoring tools) means should be added, significantly extending the capabilities available for the project.

Now, in order to define the required list of requirement values *RV*, we identify those requirements which have the reported value and assign it, assign requirement values reported for “PolicyDocument” type, and leaving others as 1 (“satisfied”).

Step 4. Risk computation per resource

Now, we have the list of requirement values *RV*, and the impact values in case confidentiality A_C, integrity A_I, and/or availability A_A are violated for every resource. This is enough to execute risk assessment for every resource separately using the computation model described in the deliverable D2.8 [1].

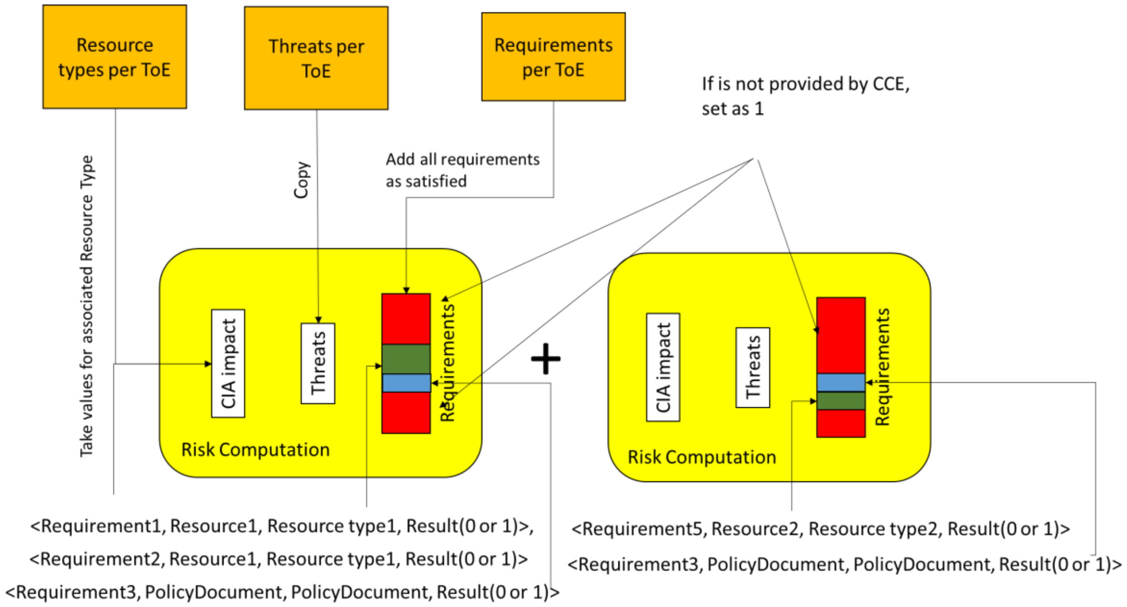

Resource types per ToE
Threats per ToE
Requirements per ToE
If is not provided by CCE, set as 1
Take values for associated Resource Type
Copy
Add all requirements as satisfied
CIA impact
Threats
Requirements
Risk Computation
+
CIA impact
Threats
Requirements
Risk Computation
<Requirement1, Resource1, Resource type1, Result(0 or 1)>,
<Requirement2, Resource1, Resource type1, Result(0 or 1)>
<Requirement3, PolicyDocument, PolicyDocument, Result(0 or 1)>
<Requirement5, Resource2, Resource type2, Result(0 or 1)>
<Requirement3, PolicyDocument, PolicyDocument, Result(0 or 1)>

Figure 4. Computation of risk during the dynamic risk assessment

The result of the computation is a list of risk values *R* for every threat and total risk (*Risk*).

Step 5. Combine risks and evaluate non-conformity

Once risk for every resource is computed, it is possible to aggregate these values for the whole service. Since the risk levels are the expected losses, i.e., monetary values (approximated from the provided impact levels), it is possible just to sum up the corresponding values. Finally, we obtain the real risk value ($risk_{real}$) for a service.

The next step is to assess a non-conformity degree. In order to compute it we need the *ideal* risk value ($risk_{ideal}$), which can be computed in the same way as it has been described above, but assuming that all requirements are satisfied for all assets. Since now all requirements are considered as satisfied independently of the considered resource, this simplifies the overall computation: there is no longer needed to consider every resource separately, but just consider all resources together, as it is defined by the core computation model. Thus, for computing ideal risk value it is required to compute the risk level only once for the whole service.

Finally, the degree of non-conformity is computed as follows:

$$10*log_{10}(Risk_{real}) - 10*log_{10}(Risk_{ideal}) < threshold$$

The usage of logarithm is required to transform the values into an interval [0;100] (rare values higher than 10^{10} are mapped to 100), which is often seen as more suitable for evaluation by a risk or compliance manager than an absolute value.

The difference (which also can be seen as the ratio of absolute values) now can be compared with a threshold. If the difference exceeds the threshold, the non-conformity is considered as major. If it is less than the threshold, then the non-conformity is minor.

Step 6. Using non-conformity assessment for the decision about the certificate status

The identification of minor and major non-conformities is essential to make decisions about the status of the certificate.

The EUCS [4] defines several certificate statuses, most importantly including the continuance, the suspension, and the withdrawal of a certificate. Additionally, a certificate can be renewed or updated, for example when certain information, like its expiration date, change. Note that the suspension of a certificate means a temporary state where the CSP can still do remedial actions, while the withdrawal of a certificate is a final decision.

The most important decisions with respect to potential for automation are therefore the continuance and suspension of a certificate. This importance stems from the fact that in an automated, continuous process, these are the most common decisions to be made and their automation would therefore replace a considerable amount of manual auditing.

Deliverable D4.3 [5] presents the parameters that can be considered in automatic certificate state changes in detail. Of these parameters, the level of security risk that exists in the cloud service is a critical one: If only a minor non-compliance or no non-compliance is identified, the certificate can be continued. If, however, a major non-compliance is identified, it can be said that there is a significant deviation between the requirements and the cloud service's security and the certificate should be suspended. Further parameters include data about compliance over time, for instance the compliance ratio of a certain requirement in the previous three months. Note that the EUCS defines further certificate statuses that are discussed in D4.3 [5].

2.3 Prioritising Non-conformities

In order to support a CSP in addressing the detected non-conformities and optimise the resources for this, RAOF ranks the failed evaluation results according to their risk reduction capabilities. This is done by performing the what-if analysis.

In short, for each *failed* evaluation result, the prioritising functionality re-computes the risk, assuming that the considered evaluation result is satisfied. The difference between the new risk value and the old one is used as a metric for prioritizing detected non-conformities[6].

An exception is done for the evaluation results linked with the resource type "PolicyDocument". Since several different evaluation results for this resource type for the same requirement are possible, only one, united failure is considered in this case. This is done in this way, since failure of this requirement affects the whole system, in contrast to specific resources, as in all other cases.

Example. Assume that the following five evaluation results failed:

1 <Req1, Res1, RType1, failed>
2 <Req2, Res2, RType1, failed>
3 <Req3, Res3, PolicyDocument, failed>
4 <Req3, Res4, PolicyDocument, satisfied>
5 <Req3, Res5, PolicyDocument, failed>

Since, the third, fourth, and fifth evaluation results are measured for the "PolicyDocument" resource type and relate to the same requirement (Req3), then only one, united case should be considered.

1 <Req1, Res1, RType1, failed>
2 <Req2, Res2, RType1, failed>

[6] Note that since we are interested only in the ordered relation between the considered cases, it is not important whether absolute or logarithmic values are considered.

3 <Req3, Res3, PolicyDocument, failed>, <Req3, Res4, PolicyDocument, satisfied>, <Req3, Res5, PolicyDocument, failed>

Let the initial value of risk be $Risk_{real}$ = 53.34.

Now, we set the first evaluation result as satisfied

1 <Req1, Res1, RType1, satisfied>
2 <Req2, Res2, RType1, failed>
3 <Req3, Res3, PolicyDocument, failed>, <Req3, Res4, PolicyDocument, satisfied>, <Req3, Res5, PolicyDocument, failed>,

and re-compute the risk (following the same procedure described above, but with this result as 1), getting lower risk value $Risk_{real}^{1}$ = 53.11. The reduction capability of the first evaluation result is $Risk_{real} - Risk_{real}^{1} = 0.2$.

Now, we set as satisfied the second case (leaving the others as failed)

1 <Req1, Res1, RType1, failed>
2 <Req2, Res2, RType1, satisfied>
3 <Req3, Res3, PolicyDocument, failed>, <Req3, Res4, PolicyDocument, satisfied>, <Req3, Res5, PolicyDocument, failed>,

and re-compute the risk value: $Risk_{real}^{2}$=52.56. The reduction capability of the second evaluation result is $Risk_{real} - Risk_{real}^{2} = 0.78$.

Finally, we do the same for the third case. Since this case consideres multiple evaluation results for our special "PolicyDocument" resource type, then all related evaluation results are considered satisfied for the analysis, i.e.,

1 <Req1, Res1, RType1, failed>
2 <Req2, Res2, RType1, failed>
3 <Req3, Res3, PolicyDocument, satisfied>, <Req3, Res4, PolicyDocument, satisfied>, <Req3, Res5, PolicyDocument, satisfied>

Let $Risk_{real}^{3}$=53.01 and $Risk_{real} - Risk_{real}^{3} = 0.33$.

As a result, we see that the second evaluation result has the highest potential reduction capability (0.78), then, follows the third one (0.33), and the weakest one is the first evaluation result. Thus, it could be suggested to the CSP to first fix the second failure, then the third and only after that the first one. Yet, this is only recommendation from the risk reduction point of view, the CSP may have other reasons (e.g., cost or required effort) to consider before defining the course of action.

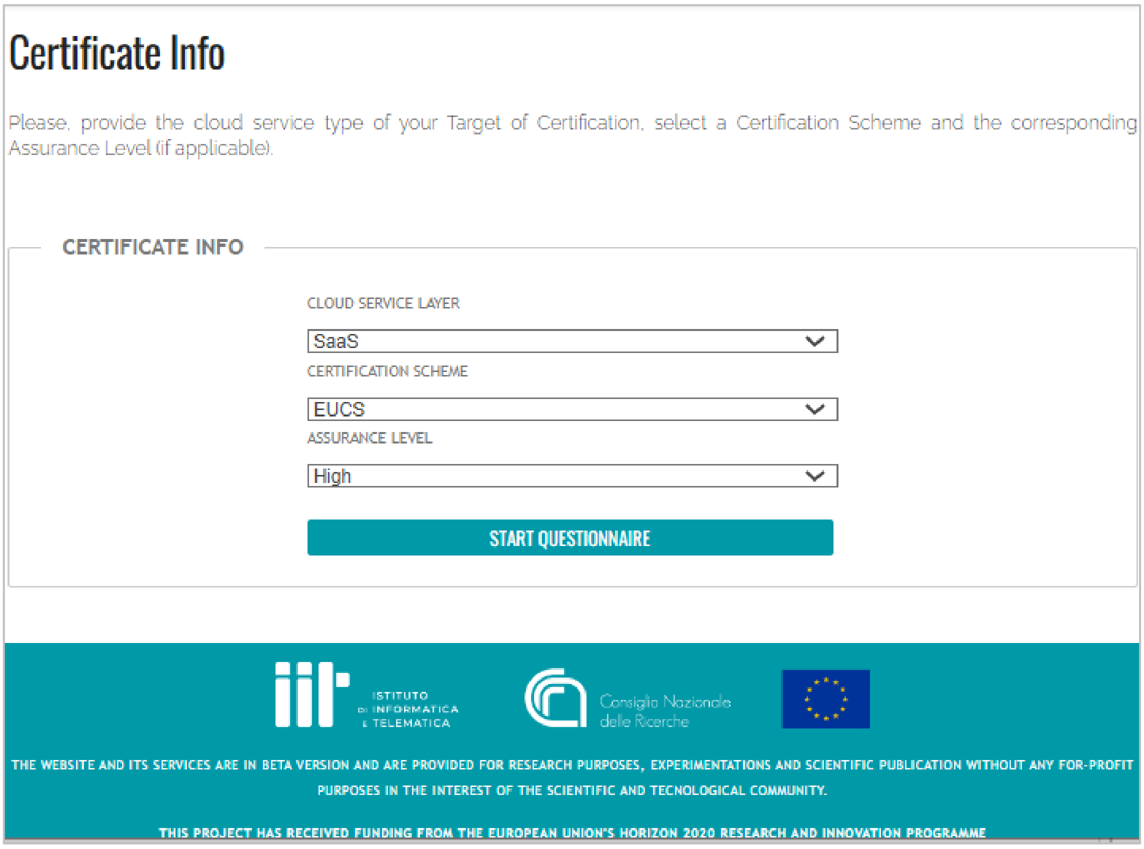
3 Implementation

The dynamic risk-based non-conformity assessment methodology is implemented as a part of the *Risk Assessment and Optimisation Framework* (RAOF). It utilises the same risk computation engine used for static risk assessment but focusses on automatic processing of input data and provisioning them further for a more comprehensive decision on the certification status.

3.1 Functional Description

The RAOF implements a service for a quick and simple risk assessment, which is used as a background for the assessment of non-conformity. Although, the static risk assessment can be used as a standalone preparation tool, the dynamic risk assessment is an integral part of the MEDINA framework.

For the dynamic risk-based non-conformity assessment, a GUI is used just before the start of the continuous evaluation in order to select a certification scheme (and the associated assurance level) (see Figure 5) and the pre-set impact values for resource types (see Figure 6). After providing these settings, the tool is ready for dynamic risk-based non-conformity assessment.



Certificate Info

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

CERTIFICATE INFO

CLOUD SERVICE LAYER: SaaS

CERTIFICATION SCHEME: EUCS

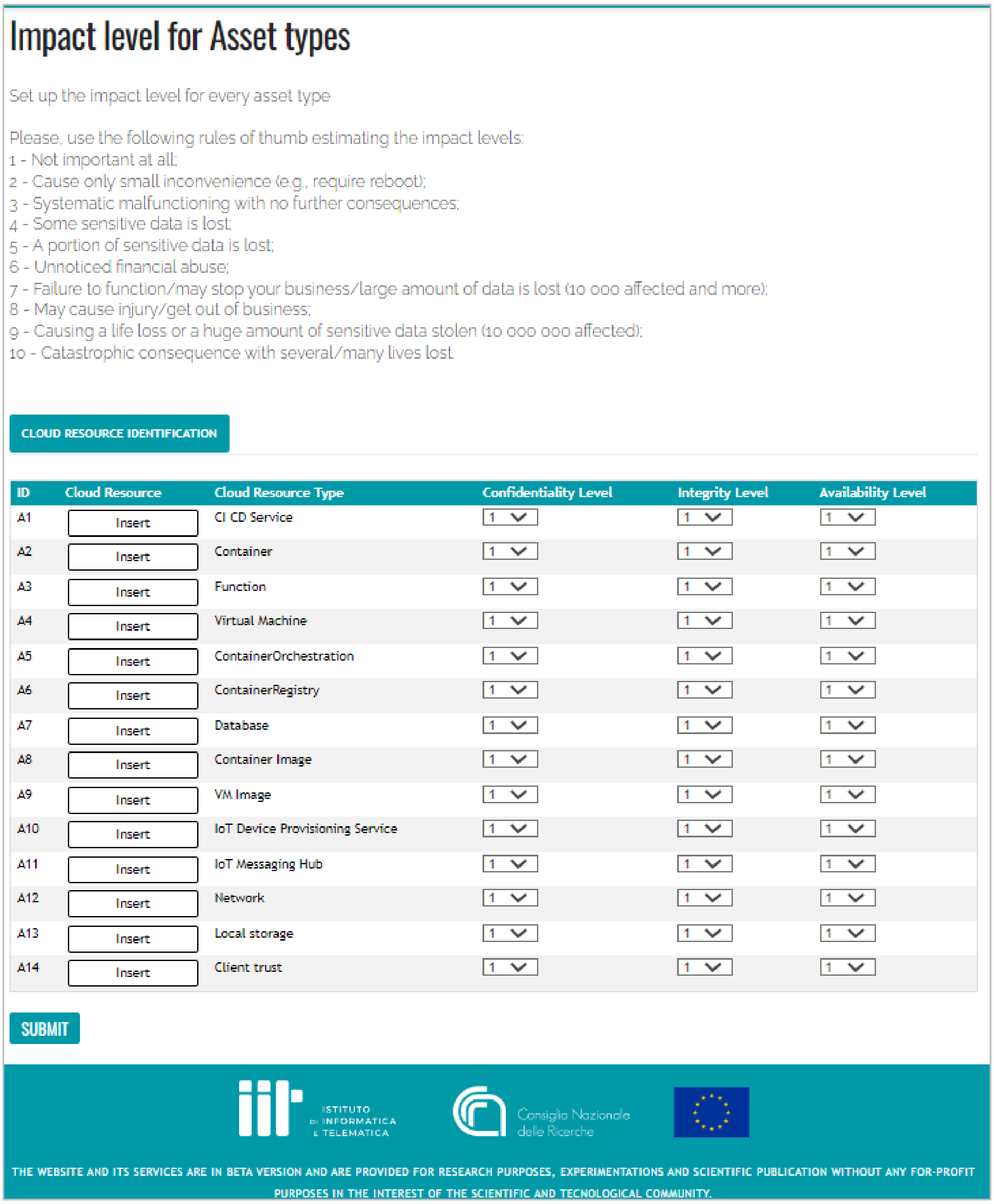
ASSURANCE LEVEL: High

START QUESTIONNAIRE
ISTITUTO DI INFORMATICA E TELEMATICA
Consiglio Nazionale delle Ricerche

THE WEBSITE AND ITS SERVICES ARE IN BETA VERSION AND ARE PROVIDED FOR RESEARCH PURPOSES, EXPERIMENTATIONS AND SCIENTIFIC PUBLICATION WITHOUT ANY FOR-PROFIT PURPOSES IN THE INTEREST OF THE SCIENTIFIC AND TECNOLOGICAL COMMUNITY.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON 2020 RESEARCH AND INNOVATION PROGRAMME

Figure 5. Selection of a certification scheme and associated assurance level



Impact level for Asset types

Set up the impact level for every asset type

Please, use the following rules of thumb estimating the impact levels:
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

ID	Cloud Resource	Cloud Resource Type	Confidentiality Level	Integrity Level	Availability Level
A1	Insert	CI CD Service	1	1	1
A2	Insert	Container	1	1	1
A3	Insert	Function	1	1	1
A4	Insert	Virtual Machine	1	1	1
A5	Insert	ContainerOrchestration	1	1	1
A6	Insert	ContainerRegistry	1	1	1
A7	Insert	Database	1	1	1
A8	Insert	Container Image	1	1	1
A9	Insert	VM Image	1	1	1
A10	Insert	IoT Device Provisioning Service	1	1	1
A11	Insert	IoT Messaging Hub	1	1	1
A12	Insert	Network	1	1	1
A13	Insert	Local storage	1	1	1
A14	Insert	Client trust	1	1	1

SUBMIT
ISTITUTO DI INFORMATICA E TELEMATICA
Consiglio Nazionale delle Ricerche

THE WEBSITE AND ITS SERVICES ARE IN BETA VERSION AND ARE PROVIDED FOR RESEARCH PURPOSES, EXPERIMENTATIONS AND SCIENTIFIC PUBLICATION WITHOUT ANY FOR-PROFIT PURPOSES IN THE INTEREST OF THE SCIENTIFIC AND TECNOLOGICAL COMMUNITY.

Figure 6. Setting up impact values for resource types

During the continuous monitoring phase, the tool periodically receives the required input data (evaluation results) from CCE through a predefined API, performs the required assessment according to the methodology described in section 2, and sends the result of the assessment (major or minor result) further for deciding about the status of the certificate (i.e., to LCM). It should be noted here that the tool does not interact with a human operator at this phase and, thus, does not use a GUI.

It should be noted, that although the main goal for the tool is to evaluate the degree of non-conformity, the tool also provides the calculated risk level. This information is stored with the collected evidence and can be retrieved by a user through a compliance dashboard or CCE.

3.1.1 Fitting into overall MEDINA Architecture

Figure 7 shows how the *Risk Assessment and Optimisation Framework* fits into the overall MEDINA architecture. It worth noting again, that RAOF is used in static (reported in deliverable D2.8 [1]) and dynamic mode (reported in this document). As shown in Figure 7, RAOF communicates with the following MEDINA components:

- *Company Compliance Dashboard* (CCD) [7] (1c). RAOF functionalities can be used through a custom tool developed by the CSP.
- *Catalogue of Controls and Metrics*[8] (2b). The Catalogue can help to provide SATRA with the responses obtained from the user when filling in its questionnaires, and thus automatically fill in a similar SATRA questionnaire. This should help to avoid doing the same tedious work twice.
- *Continuous Certification Evaluation* (CCE)[9] (12). CCE reports evaluation results to RAOF.
- *Automated Certificate Lifecycle Manager* (LCM)[10] (12c). RAOF reports the results of the dynamic risk assessment result to LCM to make the overall decision about the certificate status.

[7] *CCD* is developed in the scope of WP6 and reported in D6.3 [55]

[8] *Catalogue of Controls and Metrics* is developed in the scope of WP2 and reported in D2.2 [5]

[9] *CCE* is developed in the scope of WP4 and reported in D4.3 [54]

[10] *LCM* is developed in the scope of WP4 and reported in D4.3 [54]

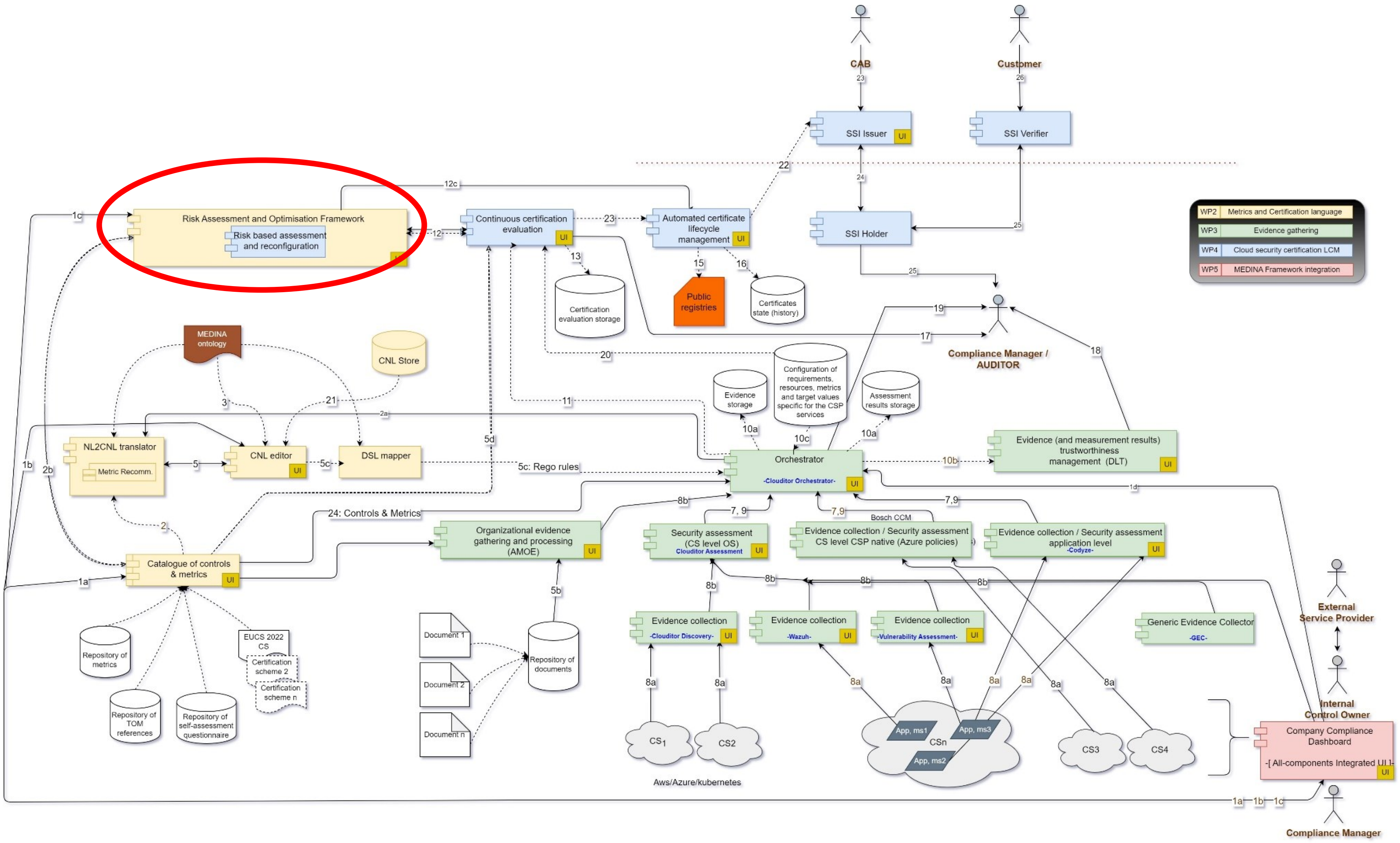

CAB
Customer
SSI Issuer
SSI Verifier
SSI Holder
UI
WP2 Metrics and Certification language
WP3 Evidence gathering
WP4 Cloud security certification LCM
WP5 MEDINA Framework integration
Risk Assessment and Optimisation Framework
Risk based assessment and reconfiguration
Continuous certification evaluation
Automated certificate lifecycle management
Certification evaluation storage
Public registries
Certificates state (history)
Compliance Manager / AUDITOR
MEDINA ontology
CNL Store
Evidence storage
Configuration of requirements, resources, metrics and target values specific for the CSP services
Assessment results storage
NL2CNL translator
Metric Recomm.
CNL editor
DSL mapper
5c: Rego rules
Orchestrator
-Clouditor Orchestrator-
Evidence (and measurement results) trustworthiness management (DLT)
24: Controls & Metrics
Organizational evidence gathering and processing (AMOE)
Security assessment (CS level OS)
Clouditor Assessment
Bosch CCM
Evidence collection / Security assessment CS level CSP native (Azure policies)
Evidence collection / Security assessment application level
-Codyze-
Catalogue of controls & metrics
Repository of metrics
Repository of TOM references
Repository of self-assessment questionnaire
EUCS 2022 CS
Certification scheme 2
Certification scheme n
Document 1
Document 2
Document n
Repository of documents
Evidence collection
-Clouditor Discovery-
Evidence collection
-Wazuh-
Evidence collection
-Vulnerability Assessment-
Generic Evidence Collector
-GEC-
External Service Provider
Internal Control Owner
Company Compliance Dashboard
-[All-components Integrated UI]-
CS1
CS2
Aws/Azure/kubernetes
App, ms1
App, ms3
App, ms2
CSn
CS3
CS4
Compliance Manager
1a
1b
1c
1d
2
2a
2b
3
5
5b
5c
5d
7, 9
7,9
8a
8b
10a
10b
10c
11
12
12c
13
15
16
17
18
19
20
21
22
23
24
25
26

Figure 7. Position of RAOF within the MEDINA Architecture (source: D5.2 [8])

BY-SA

During the MEDINA continuous monitoring phase, real measurements are collected by various assessment tools, and various metrics are used to evaluate if the requirements of the selected certification scheme are fulfilled by the considered service. Then, a decision on whether to maintain or revoke the certificate is made. The RAOF takes part in this process and provides its assessment of the detected non-conformity using risk assessment (as it is described in section 2).

Figure 8 focalises the part of the overall diagram of MEDINA framework on the considered functionality and Figure 9 shows the interfaces used in the dynamic risk assessment.

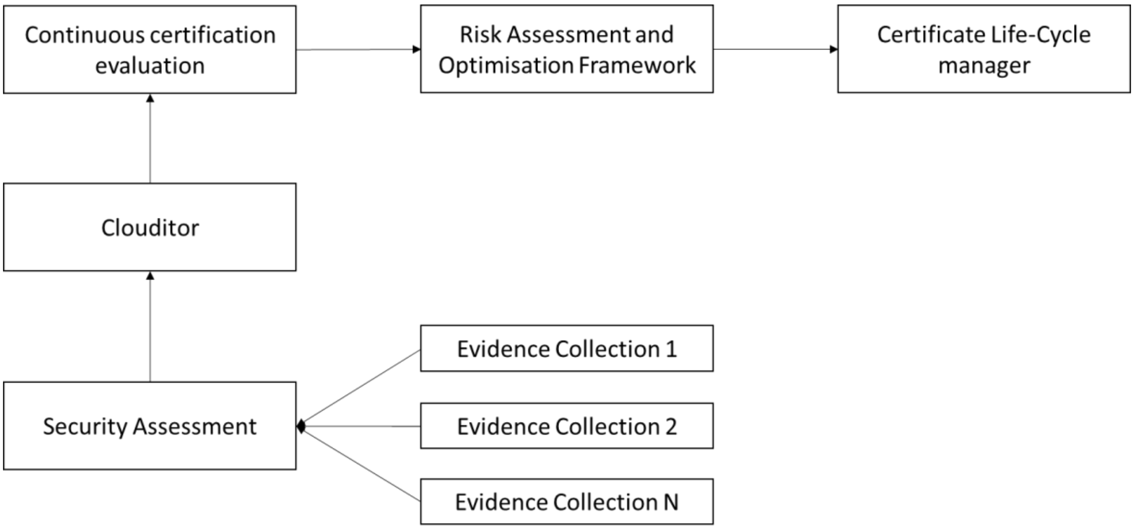

Continuous certification evaluation
Risk Assessment and Optimisation Framework
Certificate Life-Cycle manager
Clouditor
Security Assessment
Evidence Collection 1
Evidence Collection 2
Evidence Collection N

Figure 8. A part of the overall MEDINA diagram related to the dynamic risk assessment

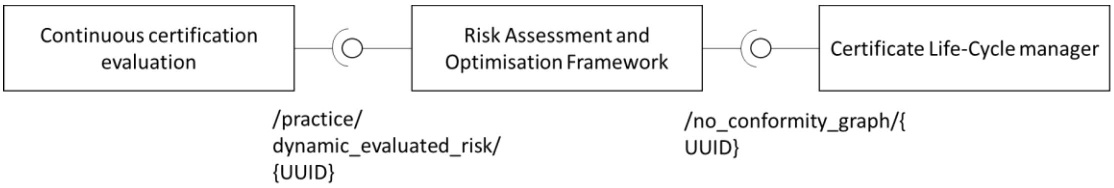

Continuous certification evaluation
Risk Assessment and Optimisation Framework
Certificate Life-Cycle manager
/practice/ dynamic_evaluated_risk/ {UUID}
/no_conformity_graph/{ UUID}

Figure 9. Components and interfaces used in the dynamic risk assessment process

The Continuous Certification Evaluation (CCE) component collects all the evidence available at some moment of time and sends them to the RAOF through a dedicated API (*/practice/dynamic_evaluated_risk/{UUID}*) providing the input for the further risk-based assessment as a JSON file (see *APPENDIX A: An example of the input JSON file sent by CCE to RAOF*). This file contains a lot of information about the assessment, but the RAOF extracts from every evaluation result only the following information:

- resource.id
- resource.resourceType
- requirement.name
- requirement.conformant (status)

This information (together with UUID) is enough to conduct the risk-based non-conformity assessment. The result of this assessment is sent to the certificate *Life-Cycle Manager* (LCM), where the decision about the state of the certificate is made using */non-conformity_gap/{UUID}* API and sending a simple JSON file, as follows:

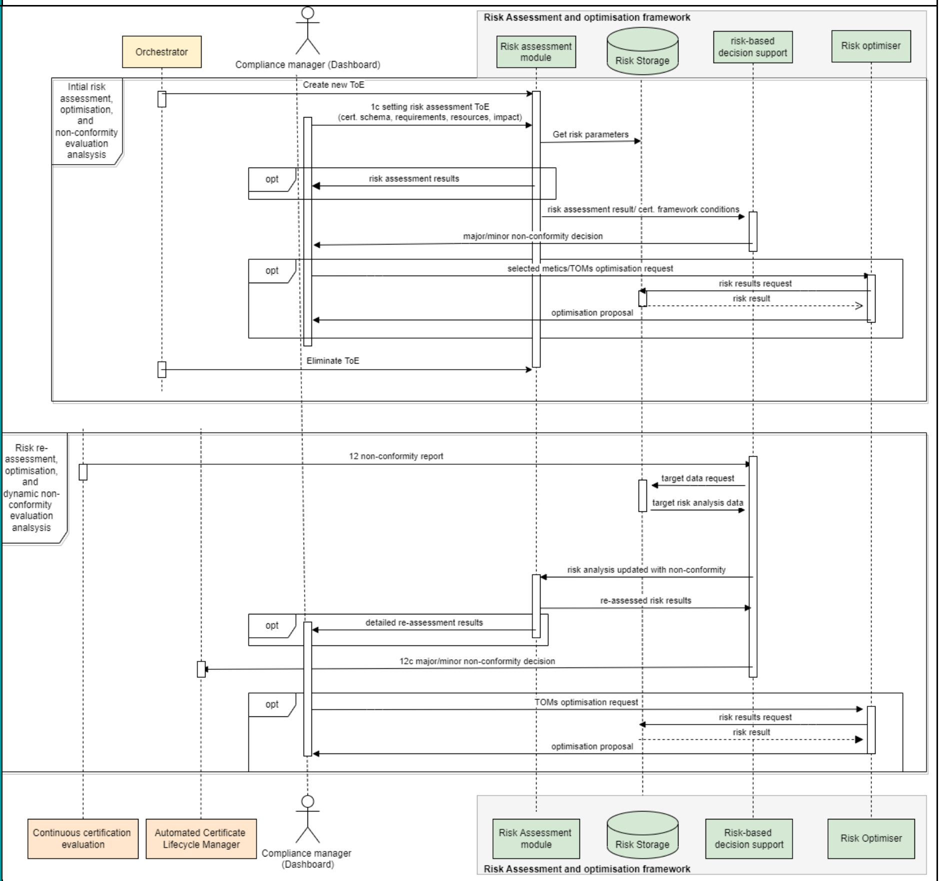
```
{
    "certificateid": str(certificate_id),
    "majordeviation": True,
    "description": 'majordeviation for '+str(uuid)
}
```

3.1.2 Component card

The component card reported here is identical to the one reported in D2.8 . It is doubled in this deliverable for the sake of completeness of the document.

<table>
<tr><th>Component Name</th><th><em>Risk Assessment and Optimisation Framework</em> (aka <em>Risk-based selection of controls Framework, SATRA</em>)</th></tr>
<tr><td><strong>Main functionalities</strong></td><td>The component provides the following functionalities:
• Risk Assessment – a questionnaire-based risk assessment facility to evaluate CSP-specific risk levels for predefined threats.
• Cost-Effective TOMs optimisation – selection the most cost-effective requirements/TOMs (to optimise investment) in case Certification Framework allows this (in contrast to rigid Frameworks).
• Risk-based analysis of deviations – risk-based evaluation of non-conformity from the framework to determine if the deviation is major or minor.</td></tr>
<tr><td><strong>Sub-components Description</strong></td><td><strong>Risk Assessment Engine</strong> – computes risk levels using the pre-established relations between asset types, threats, and requirements. Requires the list of assets and implemented requirements as input.
<strong>Risk Assessment GUI</strong> – user-friendly front-end part of the Framework which guides a user (compliance manager) through the steps for identification of main input parameters and displays results of the analysis<strong>.</strong>
<strong>Risk Assessment API –</strong> set of APIs that collect the main input parameters and provide the results of the analysis in a machine-readable format. In case all interactions with MEDINA are performed through the <em>Company Compliance Dashboard</em>, only the API is relevant.
<strong>Risk optimiser Engine</strong> – selects the most cost-relevant requirements to optimise the expected expenditure (risk + cost) given the budget or to ensure compliance with the selected Certification Framework (with, at most, minor non-conformity).
<strong>Non-conformity Assessment</strong> – internal component of the Risk Assessment Engine that compares two risk assessment results (basic and actual ones) and decides if the deviation is major or minor.
<strong>Dynamic Risk Evaluation</strong> – internal component of the Risk Assessment Engine, that manages the dynamic risk computation procedure and prioritisation of failed evaluation results.
<strong>Risk storage</strong> – storage of the current risk practices settings.</td></tr>
<tr><td><strong>Main logical Interfaces</strong></td><td><table><tr><th>Interface name</th><th>Description</th><th>Interface technology</th></tr><tr><td>Risk Assessment GUI</td><td>Graphical user interface of risk assessment</td><td>GUI</td></tr></table></td></tr>
</table>

(cc) BY-SA

<table>
<tr><td></td><td>
<table>
<tr><td>Risk Assessment APIs</td><td>Set of machine-readable APIs for risk assessment</td><td>Rest API</td></tr>
<tr><td>Non-conformity reporting API</td><td>API used for analysis and reporting a detected non-conformity.</td><td>Rest API</td></tr>
</table>
</td></tr>
<tr><td><b>Requirements Mapping</b></td><td>List of requirements covered by this component:
RBSCF.01, RBSCF.02, RBSCF.03, RBSCF.04 – covered in D2.8 [1]
RBCA.01, RBCA.02 – covered in this document</td></tr>
<tr><td><b>Interaction with other components</b></td><td>
<table>
<tr><th>Interfacing Component</th><th>Interface Description</th></tr>
<tr><td><i>Company Compliance Dashboard</i> (CCD)</td><td>Invokes <i>RAOF</i> for the selection of suggested requirements to implement, analysis of (goal) security configuration (e.g., for deviation from the target security configuration set by a certification framework), setting up resources and possible impact.</td></tr>
<tr><td><i>Continuous Certification Evaluation</i> (CCE)</td><td>Invokes <i>RAOF</i> for the evaluation of the detected non-conformity</td></tr>
<tr><td><i>Life-Cycle Manager</i> (LCM)</td><td>Consumes the result of the risk-based non-conformity evaluation.</td></tr>
<tr><td><i>Orchestrator</i></td><td>Notifies about creation/deletion of a Target of Evaluation.</td></tr>
</table>
</td></tr>
<tr><td><b>Relevant sequence diagram/s (*)</b></td><td>

Risk Assessment and optimisation framework
Orchestrator
Compliance manager (Dashboard)
Risk assessment module
Risk Storage
risk-based decision support
Risk optimiser
Intial risk assessment, optimisation, and non-conformity evaluation analsysis
Create new ToE
1c setting risk assessment ToE (cert. schema, requirements, resources, impact)
Get risk parameters
opt
risk assessment results
risk assessment result/ cert. framework conditions
major/minor non-conformity decision
opt
selected metics/TOMs optimisation request
risk results request
risk result
optimisation proposal
Eliminate ToE
Risk re-assessment, optimisation, and dynamic non-conformity evaluation analsysis
12 non-conformity report
target data request
target risk analysis data
risk analysis updated with non-conformity
re-assessed risk results
opt
detailed re-assessment results
12c major/minor non-conformity decision
opt
TOMs optimisation request
risk results request
risk result
optimisation proposal
Continuous certification evaluation
Automated Certificate Lifecycle Manager
Compliance manager (Dashboard)
Risk Assessment module
Risk Storage
Risk-based decision support
Risk Optimiser
Risk Assessment and optimisation framework
</td></tr>
</table>

Current TRL[11]	TRL4
Target TRL[12]	TRL5
Programming language	Java, Python
License	Apache License 2.0
WP and task	WP2 (Task 2.6) and WP4 (Task 4.4)
MEDINA Workflows	WF4 - EUCS Preparedness – ToC Self-Assessment, WF6 - EUCS – Maintenance of ToC certificate, WP7 - EUCS –Report on ToC Certificate (see D5.4 [9]).

(*) A more readable version of the Sequence Diagram is available at *APPENDIX B: RAOF Sequence Diagram*

3.1.3 Requirements

Deliverable D5.2 [8] defines the following set of requirements for the dynamic functionality of RAOF.

Requirement id	RBCA.01
Short title	Dynamic risk assessment
Description	Timely adjust the CSP's risk profile and re-evaluate efficiency of security configuration
Implementation status	Fully implemented

The framework is set up to automatically re-compute CSP's risk profile and use it in order to assess non-conformity of the current security configuration.

Requirement id	RBCA.02
Short title	Interface to the continuous evidence management tools
Description	Requires to consume the current status of the system configuration to re-adjust risk profile.
Implementation status	Fully implemented

The framework has an interface for consuming continuous evidence input. The concrete form of the input depends on the evaluation module, but the main information required for re-adjustment of the risk profile is as it is reported in section 2.2.

3.2 Technical Description

Since the dynamic risk-based non-conformity assessment is just a part of RAOF we only briefly outline the components of RAOF involved in this process.

3.2.1 Prototype architecture

There is no significant difference with respect to the basic architecture of RAOF presented in D2.8 [1] (see Figure 10). The main component of the framework could be split into the following subcomponents:

- **Risk storage** database where the domain layer knowledge and user input are stored.
- **Main engine** with

[11] TRL value before validation

[12] TRL value after validation

  - GUI
  - Risk assessment module
  - Non-conformity assessment
  - Dynamic risk evaluation
- **APIs**
- **Risk Optimizer**

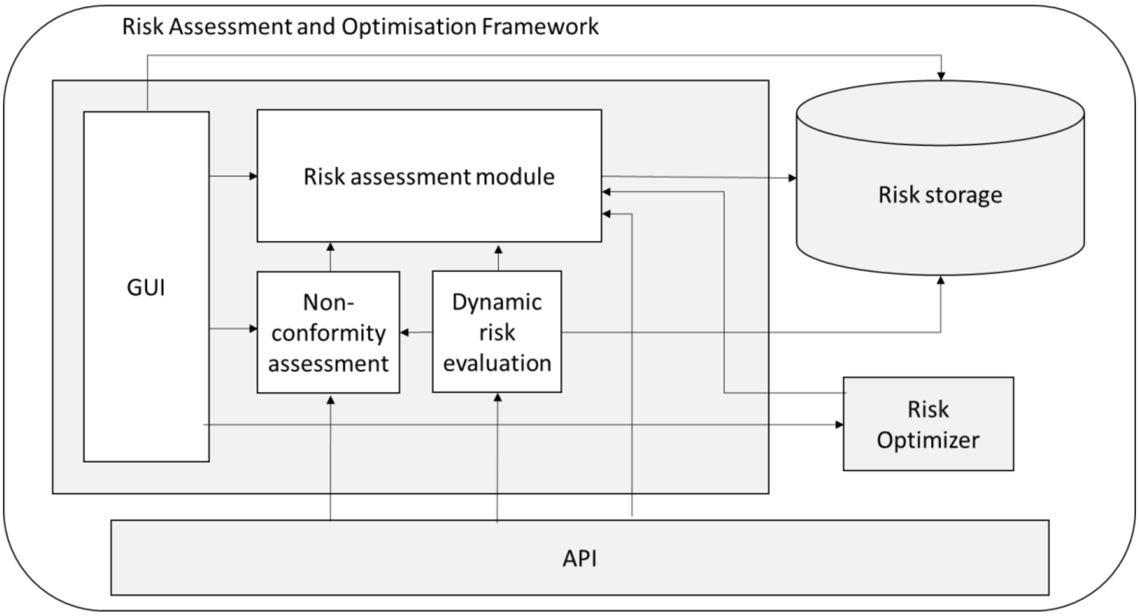


Figure 10. RAOF internal architecture

The only difference with the similar figure reported in D2.8 is the part related to the *dynamic risk evaluation*, which implements the methodology from section 2.

3.2.2 Description of Components

The *Dynamic risk evaluation* sub-component interacts with the *Risk storage database*, where the pre-set impact values for resource types are stored, and the *Non-conformity assessment* module, which performs the final comparison of real and ideal risk levels and determines if the non-conformity is major or not.

It is worth noting that prioritisation of failures is performed by the *Dynamic risk evaluation* sub-component, but not by the *Risk Optimizer*. *Risk Optimizer* implements a much more complex procedure of optimising potential investments, performed during the *static* risk assessment.

Other parts of RAOF, relevant for the static risk assessment, are explained in detail in deliverable D2.8 [1].

3.2.3 Technical specifications

There are no significant changes in the technical specification of RAOF with respect to D2.8 [1], since the dynamic risk assessment part is just an extension of the core risk assessment tool. In this subsection we only briefly summarise the technical specifications and provide the most up to date information regarding the implementation.

The RAOF component of MEDINA is implemented with the SATRA tool, which is being modified for the needs of the project. Currently it is deployed at the Kubernetes server available provided by the project:

- GUI (engine): https://integrated-ui-test.k8s.medina.esilab.org/
- APIs (app): https://risk-assessment-app-test.k8s.medina.esilab.org/api/v1/ [internal use only - authentication required]

The RAOF component has implemented the authorisation approach (using keyclock) and aligns with other MEDINA modules in this respect.

The whole project is delivered using 3 docker containers (the main engine and GUI are delivered in the same container). The main service runs over a Tomcat 8 and Apache2 Web Service. The backend is implemented in Java (and Springboot 5 framework). The frontend is developed using JSP, Javascript, HTML, and CSS. The database is the MySQL DBMS. The third part of the service is Python REST APIs created with swagger documentation. The core communication for the dynamic risk-based non-conformity assessment is executed using this facility.

4 Delivery and Usage

4.1 Package Information

The structure of the SATRA tool is the same as it is reported in D2.8 [1], since it is the same tool used in difference scenarios. We report it here for the sake of completeness.

In short, the "Risk-Assessment-tool" project is split into 3 folders: frontend-engine (which contains the main part implementing the core logic and the GUI), "app" (the APIs), and "db" (database).

Table 3. Overview and description of the project directory

Folder	Description
-app/	Contains the API interface's source code.
-db/	Contains the database backup.
-engine/deploy_war/	Contains the war file to allow docker-compose of loading this file into correct compose.
-engine/webinterfaces/src	Source code used to connect and communicate with the databases and execute the computation of risks using specific inputs and return specific output.
-engine/webinterfaces/WebContent	Contains all code and media used to implement the GUI (JSP pages/ JavaScript files, CSS, images, WEB-INF configurations).
-optimizer	Source code used to implement the risk organisation

Table 4. Overview and description of the package

Package	Description
API	
api/	Contains the source code for the API interfaces.
api.endpoints/	Contains all endpoint versions for the API interfaces.
api.endpoints.v1/	Contains the first version of the API interface.
Engine	
iit.cnr.it.hibernate.survey/	Source code to manage the connection and communication with the database that contains the survey information.
iit.cnr.it.hibernate.rat/	Source code to manage the connection and communication with the database that contains the user information.
iit.cnr.it.utility/	A sub-class and interfaces that contains functions used to perform a particular operation in computation risk class.
iit.cnr.it.security/	A sub-class to perform security features.

Package	Description
iit.cnr.it.wentool/	Contains the source code to perform risk analysis and manage input and output of this operation.
iit.cnr.it.wentool.computation/	Contains the code to compute risk analysis.
iit.cnr.it.wentool.computation.riskanalysis/	Contains the code to execute risk analysis.
iit.cnr.it.wentool.computation.input/	Contains the code to manage the input.
iit.cnr.it.wentool.computation.ouput/	Contains the code to manage the output.
utils	Contains the code to compute some operation for the API interfaces.
Optimizer	
iit.cnr.it.computation/	Contains the code to optimise risk.

4.2 Installation Instructions

This project uses docker-compose to execute and deploy the GUI and the API interfaces. There are four containers:

1. **engine**: this container contains the risk assessment module, the risk-based decision support, and the GUI
2. **app**: this container contains the API interface
3. **db**: this container is a DBMS
4. **dmm**: this container instances the risk optimizer service.

These instructions are also present in the README file in the Risk Assessment repository on TECNALIA GitLab[13]. Docker is compatible with more operating systems, such as Windows, Mac OS and Linux.

To execute the project, it is important to create a docker volume for the webserver that allows the distribution of GUI and API interfaces.

For each service there is a folder, the first service that must start is the DBMS:

- For Mac OS or Linux

```
cd -db/
sudo docker build . -t risk-assessement-db
sudo docker run -dp 32000:3306  risk-assessement-db
```

- For Windows:

```
docker build . -t risk-assessement-db
docker run -dp 32000:3306  risk-assessement-db
```

After the DBMS is started, it is possible to run the app:

- For Mac OS or Linux:

```
cd -app/
sudo docker build . -t risk-assessement-app
```

[13] https://git.code.tecnalia.com/medina/public/static-risk-assessment-and-optimization-framework/-/blob/main/README.md

```
sudo docker run -dp 5000:5000 risk-assessement-app
```

- For Windows:

```
cd -app /
docker build . -t risk-assessement-app
docker run -dp 5000:5000 risk-assessement-app
```

After the app service is started, it's possible to run the engine:

- For Mac OS or Linux:

```
cd -engine/
sudo docker build . -t risk-assessement-engine
sudo docker run -dp 8080:8080 risk-assessement- engine
```

- For Windows:

```
cd -engine/
docker build . -t risk-assessement-engine
docker run -dp 8080:8080 risk-assessement-engine
```

The last service to start is dmm:

- For Mac OS or Linux:

```
cd -dmm/
sudo docker build . -t risk-assessement-dmm
sudo docker run -dp 8082:8082 risk-assessement-dmm
```

- For Windows:

```
cd -dmm/
docker build . -t risk-assessement-dmm
docker run -dp 8082:8082 risk-assessement-dmm
```

4.3 User Manual

The user manual for SATRA is available as README in the public MEDINA repository:

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

4.4 Licensing Information

RAOF is licensed under the open-source Apache License v2.0.

4.5 Download

The source code of RAOF can be found in the public MEDINA repository:

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

5 Advancements and Future Work

5.1 Advancements within MEDINA

The SATRA tool was developed in the scope of several European projects, like CyberSure[14] and SPARTA[15]. In the scope of MEDINA, its dynamic risk assessment capability was significantly extended. Apart of the facts already mentioned in D2.8 [1] related to the change of the main focus of the tool to the analysis of non-conformities, and the application of the tool to the cloud environment and certification with the EUCS [4], the dynamic functionality was also changed in the following way:

- Dynamic risk computation was changed to perform assessment for different resources separately.
- The possibility to use of special, generic evaluations (e.g., like Policy Document) was added.
- The tool can now evaluate specific resources, as well as those that belong to a special resource type (with initial pre-set of CIA impact).
- Prioritisation of failed evaluation results has been added.
- API has been modified.

5.2 Limitations and Future Work

Naturally, our approach is not without limitations. Ideally, it should be applied when all requirements for all resources are monitored. Unfortunately, this assumption is not realistic. We have done our best to develop the computation model that still allows computing risks, yet we are constrained to assume that the not monitored requirements are fulfilled. Moreover, our assessment is limited with the resource types identified and supported by Clouditor (see D4.3 [5] for details), which are further filtered to focus only on the resources that may contain sensitive data or take part in the core business processes, i.e., whose compromise directly affects the data and the processes (the main assets). In this selection, we rely on the experience of our partners who have developed an ontology for cloud cyber security resources, which is used in this project. Last, but not least, we acknowledge that setting values for every resource rather than for all resources for the same type, could be more precise, but the potential multitude of different resources and the dynamic nature of cloud (resources could be dynamically added or removed), makes this approach impractical. Thus, after several discussions with our use case owners, we decided to stick to this generalised approach with a possibility to consider specifically the most important resources.

Possible directions for improving of the tool include, but are not limited to:

- Take into account relation between resources and inheritance of security properties
- Pre-define the state of those requirements which are hard (or impossible) to monitor
- Devise more efficient estimation of sensitivity (CIA impact) for dynamic resources.

[14] http://www.cybersure.eu/

[15] https://www.sparta.eu/

(cc) BY-SA

6 Conclusions

This deliverable describes how a cyber risk assessment for a cloud service can be applied in order to support the compliance verification process during continuous monitoring. We have provided the details on how risk-based non-conformity assessment can be performed to support the decision-making process for evaluation of the status of a certificate.

The dynamic functionality reuses the computation model defined in D2.8 [1] and extends it for the automatic operation, during which the input parameters are provided by another tool (instead of a human). Similarly, the supporting tool is not a separate service, but an extension of the exiting one, which is using API for input and output communication instead of a GUI.

This document updates the achievements reported in D4.4 [4] in a number of ways. The biggest change relates to the way the dynamic risk assessment is conducted. Now, it is possible to consider separately the most important resources that have higher level of sensitivity. The process of computation has been changed slightly and allows taking into account the assessment of the AMOE tool, focussing on analysis of documents, as a specific “PolicyDocument” resource type. Last, but not least, our tool is now able to recommend which of the detected non-conformities is the most influential and should have higher priority in the correction process.

7 References

[1] MEDINA Consortium, "D2.8 Risk-based techniques and tools for Cloud Security Certification-v3," 2023.

[2] MEDINA Consortium, "D4.4 Methodology and tools for risk-based assessment and security control reconfiguration - v1," 2022.

[3] MEDINA Consortium, "D4.3 Tools and Techniques for the Management and Evaluation of Cloud Security Certifications – v3," 2023.

[4] European Union Agency for Cybersecurity, "EUCS – Cloud Services Scheme," 2020.

[5] MEDINA Consortium, "D3.6 Tools and techniques for collecting evidence of technical and organisational measures – v3," 2023.

[6] MEDINA Consortium, "D5.2 MEDINA requirements, Detailed architecture, DevOps infrastructure and CI/CD and verification strategy - v2," 2022.

[7] MEDINA Consortium, "D5.4 MEDINA integrated solution-v2," 2023.

(cc) BY-SA

APPENDIX A: An example of the input JSON file sent by CCE to RAOF

This appendix contains an example of a JSON file which is sent to trigger the dynamic risk re-assessment process.

```
{
 "evaluationID": "6440d6e7bfa6ef1ed22e7690",
 "targetOfEvaluationId": "e3a76767-0376-499a-a8b3-3ba1394fd5f5:EUCS",
 "cloudServiceId": "e3a76767-0376-499a-a8b3-3ba1394fd5f5",
 "timeUpdated": "2023-04-20T06:08:39.237Z",
 "evaluationAnswers": [
  {
   "value": 1.0,
   "weight": 1.0,
   "threshold": 1.0,
   "code": "FABASOFT CS app.telemetry assessment tool @ IM-07.4H",
   "name": "FABASOFT CS app.telemetry assessment tool @ IM-07.4H",
   "state": "SET",
   "timeUpdated": "2023-04-20T05:09:33.782Z",
   "conformant": true,
   "resource": {
    "id": "FABASOFT CS app.telemetry assessment tool",
    "resourceType": [
     "PolicyDocument"
    ],
    "weight": 1.0
   },
   "requirement": {
    "value": 1.0,
    "weight": 1.0,
    "threshold": 1.0,
    "code": "IM-07.4H",
    "name": "IM-07.4H",
    "state": "SET",
    "timeUpdated": "2023-04-14T11:09:59.737Z",
    "conformant": true
   },
   "assessmentResults": [
    {
     "code": "IncidentManagementPolicy13 @ Resource{FABASOFT CS app.telemetry assessment tool}",
     "name": "IncidentManagementPolicy13 @ Resource{FABASOFT CS app.telemetry assessment tool}",
     "state": "SET",
     "conformant": true,
     "id": "a21ed586-b92e-42c3-a2a7-524af3c5f89a",
     "metricId": "IncidentManagementPolicy13",
     "evidenceId": "78c5c9f0-04cb-4223-b5de-e5ddcb9e478a",
     "timestamp": "2023-04-14T11:09:11Z"
    }
   ]
```

```
    },
    {
      "value": 1.0,
      "weight": 1.0,
      "threshold": 1.0,
      "code": "FABASOFT CS app.telemetry assessment tool @ OPS-08.1H",
      "name": "FABASOFT CS app.telemetry assessment tool @ OPS-08.1H",
      "state": "SET",
      "timeUpdated": "2023-04-20T05:09:45.733Z",
      "conformant": true,
      "resource": {
        "id": "FABASOFT CS app.telemetry assessment tool",
        "resourceType": [
          "PolicyDocument"
        ],
        "weight": 1.0
      },
      "requirement": {
        "value": 1.0,
        "weight": 1.0,
        "threshold": 1.0,
        "code": "OPS-08.1H",
        "name": "OPS-08.1H",
        "state": "SET",
        "timeUpdated": "2023-04-14T10:10:04.428Z",
        "conformant": true
      },
      "assessmentResults": [
        {
          "code": "DataRestoreTestFrequencyQ1 @ Resource{FABASOFT CS app.telemetry
assessment tool}",
          "name": "DataRestoreTestFrequencyQ1 @ Resource{FABASOFT CS app.telemetry
assessment tool}",
          "state": "SET",
          "conformant": true,
          "id": "df243990-eaa2-4779-9130-1a58a816ddc6",
          "metricId": "DataRestoreTestFrequencyQ1",
          "evidenceId": "e0ed1af6-df48-4ca5-804e-c2a61edc7733",
          "timestamp": "2023-04-14T10:09:15Z"
        },
        {
          "code": "SystemBackUpTesting01 @ Resource{FABASOFT CS app.telemetry assessment
tool}",
          "name": "SystemBackUpTesting01 @ Resource{FABASOFT CS app.telemetry assessment
tool}",
          "state": "SET",
          "conformant": true,
          "id": "50667f0e-a341-41f2-90a8-9ffa95f08600",
          "metricId": "SystemBackUpTesting01",
          "evidenceId": "f4cfae18-311f-4e22-82cb-f105f58197bf",
          "timestamp": "2023-04-14T10:09:15Z"
    }
   ]
  },
  {
   "value": 1.0,
   "weight": 1.0,
   "threshold": 1.0,
   "code": "FABASOFT CS app.telemetry assessment tool @ OPS-04.1H",
   "name": "FABASOFT CS app.telemetry assessment tool @ OPS-04.1H",
   "state": "SET",
   "timeUpdated": "2023-04-20T05:09:38.714Z",
   "conformant": true,
   "resource": {
    "id": "FABASOFT CS app.telemetry assessment tool",
    "resourceType": [
     "PolicyDocument"
    ],
    "weight": 1.0
   },
   "requirement": {
    "value": 1.0,
    "weight": 1.0,
    "threshold": 1.0,
    "code": "OPS-04.1H",
    "name": "OPS-04.1H",
    "state": "SET",
    "timeUpdated": "2023-04-14T13:10:06.729Z",
    "conformant": true
   },
   "assessmentResults": [
    {
     "code": "MalwareProtectionCheckQ1 @ Resource{FABASOFT CS app.telemetry assessment tool}",
     "name": "MalwareProtectionCheckQ1 @ Resource{FABASOFT CS app.telemetry assessment tool}",
     "state": "SET",
     "conformant": true,
     "id": "1bb728ac-da28-4298-bea2-77ffa6534748",
     "metricId": "MalwareProtectionCheckQ1",
     "evidenceId": "5b66fde4-7aa4-42c4-b693-58eeb4e9663f",
     "timestamp": "2023-04-14T10:09:07Z"
    },
    {
     "code": "MalwareProtectionCheckQ4 @ Resource{FABASOFT CS app.telemetry assessment tool}",
     "name": "MalwareProtectionCheckQ4 @ Resource{FABASOFT CS app.telemetry assessment tool}",
     "state": "SET",
     "conformant": true,
     "id": "58f0f326-7144-4a26-9f68-3524d9144c37",
     "metricId": "MalwareProtectionCheckQ4",
        "evidenceId": "0c8c1f33-73bf-423d-ae45-39a8c0019c72",
        "timestamp": "2023-04-14T10:09:08Z"
      },
      {
        "code": "MalwareProtectionCheckQ2 @ Resource{FABASOFT CS app.telemetry assessment tool}",
        "name": "MalwareProtectionCheckQ2 @ Resource{FABASOFT CS app.telemetry assessment tool}",
        "state": "SET",
        "conformant": true,
        "id": "317de940-d081-4563-b8c4-3aac9dd84630",
        "metricId": "MalwareProtectionCheckQ2",
        "evidenceId": "60036f25-3269-455b-a79c-e11e83984679",
        "timestamp": "2023-04-14T13:09:18Z"
      }
    ]
  },
 ]
}
```

APPENDIX B: RAOF Sequence Diagram

Figure 11 shows the sequence diagram of the *Risk Assessment and Optimisation Framework* component.

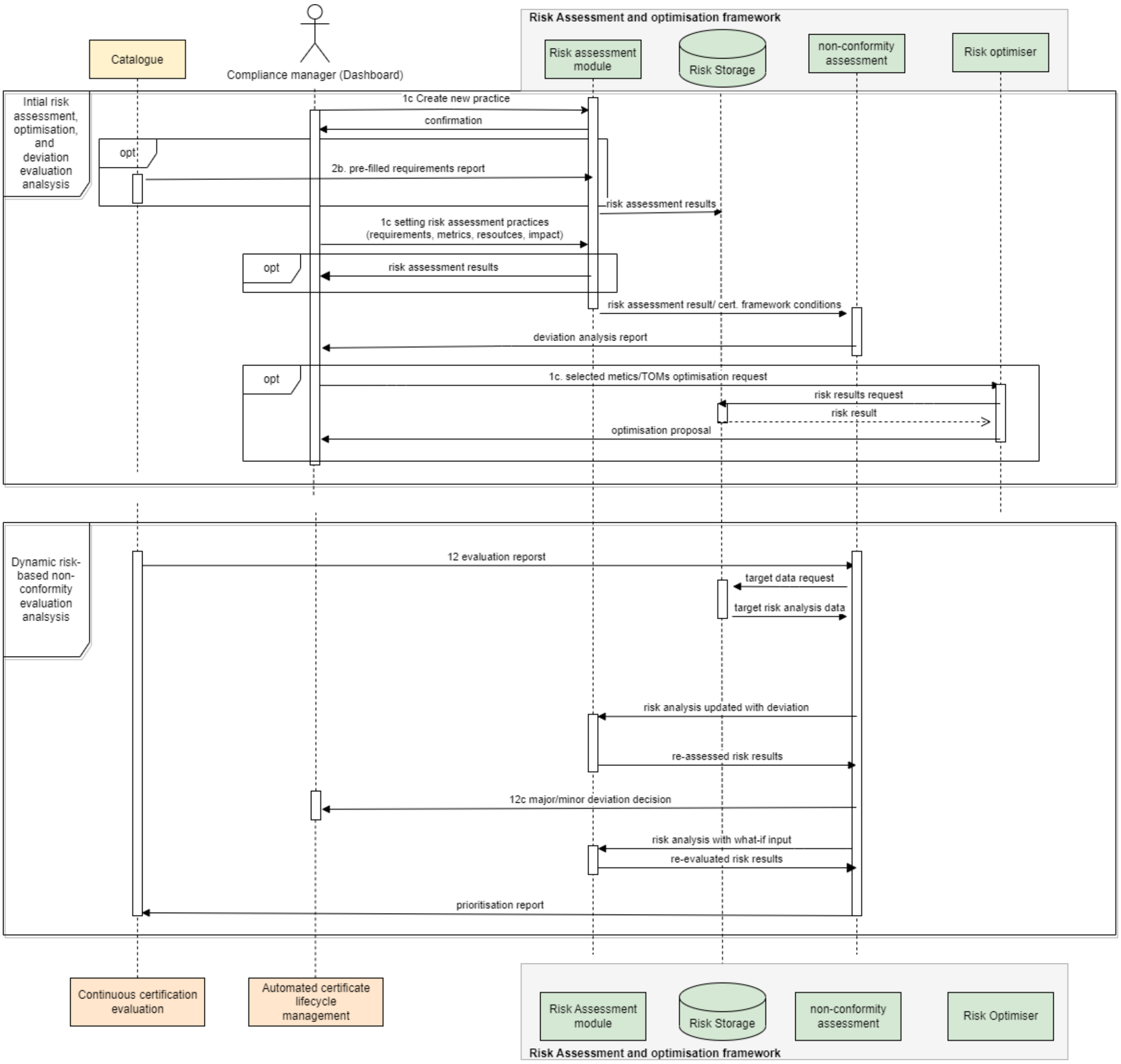

Risk Assessment and optimisation framework
Catalogue
Compliance manager (Dashboard)
Risk assessment module
Risk Storage
non-conformity assessment
Risk optimiser
Intial risk assessment, optimisation, and deviation evaluation analsysis
1c Create new practice
confirmation
opt
2b. pre-filled requirements report
risk assessment results
1c setting risk assessment practices (requirements, metrics, resoutces, impact)
opt
risk assessment results
risk assessment result/ cert. framework conditions
deviation analysis report
opt
1c. selected metics/TOMs optimisation request
risk results request
risk result
optimisation proposal
Dynamic risk-based non-conformity evaluation analsysis
12 evaluation reporst
target data request
target risk analysis data
risk analysis updated with deviation
re-assessed risk results
12c major/minor deviation decision
risk analysis with what-if input
re-evaluated risk results
prioritisation report
Continuous certification evaluation
Automated certificate lifecycle management
Risk Assessment module
Risk Storage
non-conformity assessment
Risk Optimiser
Risk Assessment and optimisation framework

Figure 11. RAOF Sequence Diagram