Deliverable D2.7

Risk-based techniques and tools for Cloud Security Certification – v2

Editor(s):	Artsiom Yautsiukhin
Responsible Partner:	CONSIGLIO NAZIONALE DELLE RICERCHE
Status-Version:	Final – v1.0
Date:	04.11.2022
Distribution level (CO, PU):	PU

Project Number:	952633
Project Title:	MEDINA

Title of Deliverable:	Risk-based techniques and tools for Cloud Security Certification-v2
Due Date of Delivery to the EC	31.10.2022

Workpackage responsible for the Deliverable:	WP2 - Certification Metrics and Specification Languages
Editor(s):	Artsiom Yautsiukhin (CNR)
Contributor(s):	
Reviewer(s):	Anže Žitnik (XLAB), Cristina Martínez (TECNALIA)
Approved by:	All Partners
Recommended/mandatory readers:	WP2, WP4, WP5

Abstract:	This set of deliverables will contain the risk-based cost-benefit analysis for the selection of security controls. This deliverable will describe the core model of the risk-based framework (M15) and its implementation as an integral part of the MEDINA solution (M24, M30). These deliverables are the result of Task 2.6.
Keyword List:	Risk assessment, assets, threats, controls, requirements
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	30.09.2022	ToC version	CNR
v0.3	03.10.2022	First draft version	CNR
v0.7	10.10.2022	First complete version	CNR
v0.9	27.10.2022	The reviewer's comments are addressed	XLAB, CNR
v1.0	04.11.2022	Ready for submission	TECNALIA

(cc) BY-SA

Table of contents

List of tables

List of figures

(cc) BY-SA

Terms and abbreviations

API	Application Programming Interface
AT	Asset Type
CI/CD	Continuous Integration / Continuous Delivery
CIS	Center for Internet Security
CSA or EU CSA	Coordination and Support Action
CSC	Cloud Service Customer
CSP	Cloud Service Provider
CSS	Cascading Style Sheets
DBMS	Database Management System
DoA	Description of Action
DoS	Denial of Service
EC	European Commission
EUCS	European Cybersecurity Certification Scheme for Cloud Services
GA	Grant Agreement to the project
GUI	Graphical User Interface
HTML	HyperText Markup Language
IaaS	Infrastructure as a Service
IAM	Identification and Authentication Management
IoT	Internet of Things
JSP	Java Server Page
KPI	Key Performance Indicator
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
SQL	Structured Query Language
SW	Software
ToC	Target of Certification
TOM	Technical or Organisational Measure (aka requirement)
VM	Virtual Machine

Executive Summary

This deliverable presents an intermediate version of the risk assessment model and supporting tool, which will be used as a decision-making instrument for the analysis of non-conformities of a cloud service with a selected certification scheme. The model defines the main risk components (i.e., assets, threats, and vulnerabilities) and relations between them. Also, the model is applied to the cloud domain with the set-up of the model for this context.

The model is implemented with a Self-Assessment Tool for Risk Analysis (SATRA), as a Risk Assessment and Optimisation Framework (RAOF) component of MEDINA. The current version implements all core functionalities and options of the model, leaving for the future integration issues and values calibration.

This deliverable reports the results of the two years of Task 2.6 and demonstrates implementation of Risk Assessment and Optimisation Framework, which will support a CSP in the analysis of non-conformities with a selected certification scheme during the preparation phase and during the continuous compliance monitoring. In addition, this deliverable also provides the theoretical background and first implementation of the functionality for optimal selection of additional security requirements to implement.

The document consists of the following main sections:

- Section 2 describes and specifies the role and place of the risk assessment in the scope of the MEDINA framework. Both usages of risk assessment, including the preparation phase and continuous compliance monitoring, are outlined.
- Section 3 provides a short state of the art on the topic of risk assessment.
- Section 4 is dedicated to the risk assessment model, which consists of the following three layers: conceptual, domain and individual. The core focus of this document is on the first two, which define the mathematical model and set it up for the usage in the cloud domain. The individual layer is defined for collection and analysis of the inputs provided by a CSP. This section also provides the background for our risk optimisation.
- Sections 5 and 6 describe the current status of the supporting tool, its place in the MEDINA workflow and provide some technical details about its implementation.

The first version of this deliverable, D2.6 [1], was released 9 months ago (and was recently updated according to the request of the EC). In the first version we outlined the place of risk assessment in the scope of MEDINA, provided the core description of the risk computational model, and released the first version of the RAOF component.

This deliverable is an updated version of D2.6 and most of its content remains as it was in D2.6 (with some changes), allowing D2.7 to be self-contained. This update is focused on implementing additional features for the RAOF component and implementing the new functionality for optimising implemented security requirements. In the final deliverable (D2.8) we will focus on elaborating the pre-set parameters for risk assessment and smoothing integration with other MEDINA components.

Finally, it is important to note that the computation model described in this deliverable is also used for continuous monitoring phase, and thus contributes to successful completion of D4.4 [2] and D4.5 [3].

(cc) BY-SA

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 we use the risk assessment 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).

The risk assessment model presented in this deliverable is based on the certification scheme to be used, and, thus, helps to analyse the risk from the certification scheme perspective. The approach itself is simple, fast and much less dependent on the knowledge of the CSP than many other risk assessment methods. Thus, once it is set up properly, it can be used for dynamic risk assessment and non-conformity analysis. At the same time, our risk assessment does not have a goal to substitute the risk assessment performed by the CSP to set up its system according to its own risk management strategy (as it is demanded by many certification schemes, e.g., EUCS [4]). In short, our risk assessment model and a supporting tool are made for the purpose of supporting MEDINA's certification management process.

1.1 About this deliverable

The main goal of this deliverable is to describe the computational model and tool for risk assessment which supports compliance verification and certification process. It reports the main findings and results of task T2.6 "Risk-based techniques for Certification Assurance Levels" for the first 24 months of the MEDINA project.

This document is devoted to the second version of the proposed model and supporting tool. The first version, reported in D2.6 [1], defined the basic concepts, relations, and the computational model. It also listed the settings of the cloud-specific parameters, i.e., threats, asset types and vulnerabilities (which, in our model, are considered as lack of implementing requirements from the selected certification scheme).

The second version of the model reported in this deliverable does not introduce changes in the computational model, but mostly extends the functionalities to the supporting tool. In particular, we added the possibility to adjust relevance of different threats depending on the selected cloud service layer. Also, we modified the tool to assess non-conformity depending on the selected assurance level. Our tool now also has the possibility to compute non-conformity for different certification schemes, but this functionality will not be used in MEDINA, since other components focus only on EUCS [4]. In addition, completely new functionality is added to help the CSP to optimise its expenditure in selecting additional requirements to be implemented. The statement of the optimisation problem and a brief description of its solution (based on Genetic Algorithm approach) is provided in Section 4.6 and its implementation is displayed in Section 5.1.2.

The supporting tool (Risk Assessment and Optimisation Framework (RAOF)[1]) implements the outlined functionalities. Similar to the core model, the tool has been updated and set up for the purpose of MEDINA (i.e., for the use in the cloud security certification process). The tool provides the required interfaces for its integration into the MEDINA platform.

It is also worth mentioning that the provided computational model will be used for static risk assessment during the preparation for certification by the CSP, as well as during the continuous monitoring phase, during which the compliance of the cloud service with the selected certification scheme will be continuously evaluated. The use of the risk assessment model for evaluation of non-compliance of the system during the continuous monitoring phase has been described in the deliverable D4.4 [2]and will be updated in D4.5 [3]. This deliverable only briefly outlines how risk assessment can be used during this phase.

1.2 Document structure

The document is structured as follows. *Section 2* describes how risk assessment could support (continuous) compliance management process, in general, and the one of MEDINA, in particular. A short state of the art analysis is provided in *Section 3*. The main part of this document is *Section 4*. This section provides in detail the description of the risk assessment model, including the identification of its main components: assets, threats and vulnerabilities, and aggregation of the estimated values to receive a risk level and analyse it from the compliance point of view. This section also describes our approach for risk optimisation. *Section 5* includes the description of how the supporting tool (called SATRA), implementing the functionality of Risk Assessment and Optimisation Framework, is developed and integrated into the overall MEDINA Framework. Finally, the information about the delivery and usage of the tool is provided in *Section 6*. The conclusions, limitations, and future steps are outlined in *Section* 7.

1.3 Updates from D2.6

This deliverable is an updated version of D2.6 [1] and most of its content remains as it was in D2.6 (with some changes), allowing D2.7 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 D2.6

Section	Changes
2	Minor updates
3	Minor updates
4	The list of assets has been purged, to focus on the resources which could cause losses once compromised. Also, now this section includes the background behind resolving the optimisation problems used by our optimisation module
5	This section was aligned with the latest decisions made in scope of the project, changes in the GUI and new functionalities of the component.
6	Similarly, the modifications in the structure and in the installation procedure were added to Section 6.
7	Section 7 now also includes limitations of the model and its implementation.

[1] This framework is realized by a tool called Self-Assessment Tool for Risk Analysis (SATRA).

2 Risk-based Support for the Certification Process in MEDINA

This section explains when and how risk assessment contributes to the delivery of the main goal of MEDINA. It is dedicated to the brief, high level description in order to set up a clear vision of the position of the risk assessment in scope of the project, leaving the related technical details to Sections 4 and 5.

First and foremost, we would like to underline that our risk assessment process (although could, but) is not aimed to substitute the one performed by the CSP. The cyber risk assessment process of a CSP is (or should be) an integral part of the cyber risk management process, which in its turn should be a part of the overall CSP's risk management framework. Thus, CSP may be constrained by the management to use specific methods, tools, and approaches for cyber risk assessment, which will be further used as an integral part of the risk management framework.

Second, CSP's risk assessment most probably will be more customised for the needs of this CSP, supported by customary sub processes for collecting risk-related information (e.g., analysis of business goals of the CSP, applied business processes, collected statistics, and other types of similar private information), interviewing different members of the security team, consulting with external experts, etc. This process requires a lot of time, effort, and knowledge, but provides more CSP-focussed results of the assessment.

On the other hand, a risk assessment process supporting continuous monitoring of certification must be fast, cheap and as less dependent on the evaluator as possible. Moreover, it must provide the results relevant for the decision making about the state of a certification/compliance. That is why our risk assessment method is more suitable for this purpose than the more fine-grained and in-depth process often[2] followed by CSPs.

Risk assessment contributes to the MEDINA process in two ways. First, it provides a risk-based evaluation support for the CSP which is preparing for certification. A CSP may evaluate its readiness to be certified by running our risk assessment engine and analysing the degree of non-conformity. Naturally, for some certification schemes like EUCS, the CSP should aim to implement all requirements for the selected assurance level. On the other hand, in case of presence of non-conformities, the CSP, with the help of our risk assessment framework, may show that the existing non-conformities are only minor ones (insignificant) and in this specific case are not essential.

Moreover, in the case of a limited budget a CSP may prefer to consider different alternatives for the implementation of requirements, aiming to satisfy the targeted level as much as possible. Our risk assessment tool will provide an instrument for the CSP to compare the alternatives. Furthermore, the tool will also automatically select the most risk-optimal configuration by selecting the not satisfied requirements of the chosen certification scheme, which will help the CSP to reduce the risk in a cost-efficient way and stay within the budget limits. Naturally, the latest functionality is useful only if the available budget does not allow satisfaction of all requirements and if satisfaction of additional requirements is cost-efficient.

Second, the risk assessment will be used during the continuous monitoring for analysing the detected non-conformities. This assessment is to be performed on the fly taking the current state of satisfaction of requirements per asset as an input and aggregating the risk level for all resources of the CSP, providing MEDINA with the assessed level of non-conformity: major or minor.

[2] We need to note that although risk assessment is a widely acknowledged best practice for cyber security management, unfortunately, some CSPs still do not use it.

2.1 Preparation phase

First, risk assessment is to be applied to support the CSP in *preparation* of the system for certification against a selected certification scheme. Our supporting tool could be used by a CSP to help it in the decision-making process about covering the security requirements of the scheme, which are essential for this specific CSP (i.e., according to its risk level). Naturally, satisfaction of all requirements for some schemes (e.g., EUCS) is important, but some requirements of the scheme could be, on the one hand, not very effective for a specific provider (e.g., no sensitive cloud service customer (CSC) data are stored), and could be costly to implement, on the other one. Thus, risk assessment could help to evaluate the level of non-conformity and support the decision of the CSP in justification of why some requirements are not implemented. On the other hand, major non conformities could be spotted before engaging in the certification process and the CSP will know what should be corrected.

In order to perform risk assessment for its service, the CSP is asked to provide the following information:

1. The certification scheme and (if available) the assurance level against which the system is to be certified.
2. The cloud service level, i.e., IaaS, PaaS, or SaaS.
3. A list of resources (assets) it manages and the following information about them:
    a. the pre-defined resource types (see Section 4.1) to which the defined resource belongs,
    b. the potential impact in case Confidentiality, Integrity or Availability of the resource is compromised,
    c. the approximate number of such resources.
4. The information about which requirements from the selected scheme are covered.

This input data is collected in a form of a questionnaire and a dedicated table for resources.

If a CSP is able to cover all requirements from the selected scheme, there is no need for further analysis, since such CSP is doing well and should proceed with asking for certification (and start monitoring its claims during the continuous monitoring phase). In case some requirements cannot be covered, risk assessment may help to perform the following types of assessment:

1. *Non-conformity evaluation.* The risks assessment may help to estimate how far the service is from the “ideal” state (i.e., a state in which all requirements are covered). The CSP may evaluate whether the existing non-conformity is major (and it is unlikely for an auditor to certify the system) or minor (and the existing non-conformities could be justified in front of an auditor). For doing this, we compute the ideal risk level for the CSP (assuming that all requirements are satisfied) and compare the value with the risk level computed with the values provided by CSP, using the same information about the assets.
2. *Compare different systems (different states)*. The risk assessment may help to compare risks of different system states and select the one which will be more probably certified (i.e., with lower non-conformity). This can be especially important if additional investments (which are required to cover additional requirements) are limited or there are other reasons preventing satisfaction of all requirements.
3. *Select the requirements/TOMs which should be covered (in addition to already covered ones) to ensure only minor non-conformity with available budget.* The risk assessment can be used to optimise investments and ensure good (minor non-conformity) coverage of requirements. This optimisation problem will require automatic selection of

requirements which can be covered with the identified budget and verification of the level of non-conformities with risk assessment.

Last, we should also underline that for some CSPs it can be useful to see alternative results of the risk analysis to compare them with the results obtained by their in-house risk assessment. Moreover, the risk level provided by our framework may serve as an indicator of a security state for those CSPs which target lower assurance levels (e.g., Basic or Substantial for EUCS [4]), but would like to improve their security by implementing additional requirements which belong to a higher level of assurance (even though they are not aiming to be certified against them).

2.2 Continuous monitoring phase

Risk assessment also provides an important service during the continuous monitoring phase, the core phase targeted by MEDINA. The main goal of risk assessment in this phase is to analyse the detected cases of non-conformity and evaluate them with respect to the deviation from the ideal level.

In contrast to the preparation phase, in the continuous monitoring phase risk assessment has another source of input about the fulfilled requirements, i.e., results of the metric assessment. First, this allows making the analysis more objective, eliminating human errors (deliberate and incidental) from the equation. Second, it is possible to compute the current risk level based on up-to-date information (taking into account all recent changes). Third, it is possible to estimate up to which degree a requirement is satisfied based on the assessment of different metrics associated with this requirement.

The risk assessment for continuous monitoring must be automatic, fast and independent from human input. Thus, our risk assessment in this phase is based on:

- The information about the certification scheme, assurance level and cloud market type selected before starting the continuous monitoring phase.
- Assets and the related information (e.g., types or severity levels) determined before starting continuous monitoring phase (although, there could be a possibility to update this information).
- The information about the failed assessments of some metrics (provided by assessment tools of MEDINA) and their contribution to the requirements (contained in the MEDINA's Catalogue of controls and metrics). This information can and should be updated as frequently as assessment tools are able to provide it.

Once a non-conformity is detected, the risk assessment tool will be able to analyse how important it is (major or minor) and provide the result of the assessment to the component making a decision about certification status (and/or auditor).

2.3 Current status

The contribution of the risk assessment to the overall process of MEDINA stated above represents the current vision about its involvement. The current version of the model and supporting tool implements all the planned functionalities, including the risk assessment, non-conformity evaluation, recommendations for optimisation, etc.

We would also like to note that our risk assessment model must be set up for providing the intended service. By "setting up the model" we mean populating it with concrete values (e.g., asset types, threats, etc.) as well as relations between them (e.g., reduction of attack probability by TOMs) that will be embedded in the supporting tool. The current versions of the model and

BY-SA

tools have been initially pre-set with values, which will be re-evaluated and corrected as finalising the activities of Task 2.6 and will reported in the concluding deliverable (D2.8).

3 State of the Art on Risk Assessment Techniques

Risk management is a well-known management practice for evaluation, treatment and keeping under control various events of uncertain nature. Since occurrence of cyber security incidents is uncertain it is natural to apply risk management procedures for managing cyber security risks. Moreover, recent reports show that organisations see cyber risk as one of the top risks for their operation [5]. It is not surprising to see the requirement for proper cyber security risk management in all major cyber security standards, like ISO 27001 [6], NIST CSF [7] and EUCS [4].

There are various books describing the basics of cyber security risk assessment (e.g., [8]) and a plethora of various approaches ranging from generic methodologies (e.g., ISO 27005 [9], NIST 800-30 [10] Octave Allegro [11], Magerit [12], RiskIT [13]) up to specific computational methods [14], [15], [16], [17] and tools (e.g., [18], [19]).

The methodologies mostly focus on defining a risk assessment process, describing the required activities, helping to identify the stakeholders for conducting every activity, etc. G. Wangen et al., [20] conducted a detailed analysis and comparison of the procedural activities. These generic methodologies often do not specify precisely how activities should be executed, leaving this for the analyst, but may suggest some various techniques which can be of use. For example, Magerit [12] suggests several techniques for several crucial steps, like identification of threats (e.g., Dephi evaluation, attack trees, etc.); ISO 27005 [9] provides lists of possible actors, threats and consequences, Octave [11] and NIST [10] propose worksheets to be filled in. Most of the methodologies follow the qualitative risk assessment method, which computes the risk level (usually, high, medium or low) using the estimated probability and impact levels as input. The "computation" is performed with help of a simple risk table/matrix (see an example of such risk table in Table 2). Naturally, qualitative risk assessment is simple to apply, but is very imprecise and confusing [21], [22]. At the same time, it is worth noting that these generic methodologies often do not mandate using this qualitative approach and can be used with semi-quantitative and quantitative computation methods, but no specific guidelines are, usually, provided.

Table 2. Risk Table

		Impact		
		Low	Medium	High
Probability	Low	Low	Low	Medium
	Medium	Low	Medium	High
	High	Medium	High	High

Regardless of the applied computational method and used techniques for identification and estimation of the main risk components (i.e., threat, vulnerability and impact), the risk assessment (and treatment, if available) process heavily depends on the analyst(s), who is(are) required to execute every step. The process is long, effort-demanding and requires very good knowledge of cyber security and current trends.

Factor Analysis of Information Risk (FAIR) [23] is a risk assessment approach which aims to estimate loss exceedance probability (i.e., the probability that the loss will be greater than a specific amount). The approach defines a simple (three levels) ontology of the basic terms which are used to estimate the factors for the basic risk components. FAIR uses a quantitative risk assessment, i.e., the factors are estimated with quantitative values which have minimal and maximal limits (aiming to limit the values with 90/95% confidence). Then, all the factors are aggregated (by a tool) using the Monte Carlo method. The result is a graph, which represents the probability of facing a loss greater than a specified amount. The method is criticised for being too complex (e.g., it requires quantification of many factors) [24]. On the other hand, D. Hybbard and R. Seiersen [21] argue that such complexity could be overcome with time, gained experience and more data collected, but the usefulness of the result is much higher comparing not only with semi-quantitative or qualitative methods but also with usual quantitative ones. Some approaches, like [14], [17], [15], are very similar, but reduce the complexity by using the confidence intervals for quantifying directly the basic risk components (instead of multitude of factors).

CORAS[3] is a model-based risk assessment approach, which provides a graphical language, an assessment method and a process. The process of the CORAS approach, in general, follows the well-known risk assessment methodologies. The graphical language supports the modelling activity, which helps the analyst to identify possible threat actors, attack scenarios, vulnerabilities, unwanted incidents and affected assets. These basic risk components are to be identified and related, and values of likelihood and impact estimated. The risk assessment method uses the defined model and is primarily qualitative, but quantitative risk analysis is also possible.

Since risk assessment is a well-known practice, various tools have been developed to conduct the assessment. Some of these tools are proprietary tools of the consulting agencies (e.g., DGS' RiS[4]) and freely available information about them is very limited. For example, MONARC[5] [19] is a semi-quantitative tool for cyber risk analysis. The tool allows adding main resources/assets and pre-filling the results with pre-defined values for likelihood and impact. Similarly, SPIDERISK[6] has the possibility for automatic prefilling the results of the assessment, but it requires defining the model of the system and relation between assets. Both these tools have the capability to suggest actions for risk reduction. Although these tools aim to significantly reduce time and effort for assessment, they still rely heavily on the analyst to model the system and estimate some parameters (e.g., risk reduction amount), which requires good knowledge of cyber security.

The scientific literature either reviews and analyses existing risk assessment methods [20], [25], [26], focuses on improved computational methods [16], [27], [28], [29], [14], [15], [16] or applies risk assessment in a specific domain (e.g., military [30], SCADA [31], automotive [32] , maritime [33], cloud [34], etc.). Here we focus only on those works which propose specific computational methods.

ISRAM [16], similar to FAIR, is the analysis which starts with identification and estimation of various factors. In contrast to FAIR, ISRAM does not have a defined ontology and the defined factors directly contribute to estimation of likelihood and impact of possible events. A weighted function is used to aggregate the results of assessment of factors for likelihood and impact (a semi-quantitative approach is followed). Several participants are assumed to take part in the assessment and average values for likelihood and impact are defined using their assessments. A very similar semi-quantitative approach applying several factors for estimation of event likelihood was used by F. Farahmand et al., [35]. Also B. Sheehan et al., [36] applied analysis of several factors (split as barriers and escalators) to aggregate opinions of experts and estimate event likelihood and impact.

One of the key problems in risk assessment is the estimation of event likelihood. Several authors [27], [28] proposed to use Common Vulnerability Scoring System (CVSS) scores[7] which help to rate identified vulnerabilities. These authors propose to scan the considered system to identify

[3] https://coras.tools

[4] https://www.dgsspa.com/pagine/15/ris

[5] https://www.monarc.lu/

[6] https://spyderisk.com/

[7] https://www.first.org/cvss/

existing vulnerabilities and then use their scores to determine the likelihood of an attack which uses them (this is often done with an attack graph model, which defines the vulnerabilities to be exploited for a successful attack [27]). The CVSS scores can be used as such [27] or only their integral parts could be considered [28] (sometimes, the part of the CVSS scores related to impact is used for estimation of the impact of the overall attack). Although this approach can be executed with existing scanning and attack graph building tools and CVSS scores are already defined, there are a number of problems with using CVSS scores for estimation of probabilities. First, CVSS scores were defined for ranking vulnerabilities, and their usage in any computations is doubtful. Second, there is no evidence that CVSS scores indeed correlate with event likelihood.

Another popular approach to estimation of event likelihood and impact is integrating an attack tree [37] into the risk computational model (e.g., [29], [17]). In short, every event/attack is broken down into simple steps and their alternatives. The model is represented as an AND-OR tree. Values are assigned to the simplest steps (leaf nodes) and aggregated to obtain the result for the considered event. Naturally, for every event/attack a tree must be built by the analyst and many values are to be assigned to the leaf nodes.

There were also several attempts to define a cyber security risk assessment approach specifically for cloud environment [38], [34], [39], [40], [41]. In most cases, these approaches simply apply existing risk analysis methods (mostly quantitative ones) for cloud [40], [42], [41]. O. Akinrolabu et al., [38], [34] proposed Cyber Supply chain cloud Risk Assessment (CSCCRA), which includes two separate analysis: 1) analysis of security of the supply chain (using 9 security categories as factors and z-score for aggregation of these factors) and (2) a FAIR-like analysis, with only probability and impact values estimated (instead of several factors and their further aggregation as FAIR does). The quantitative risk and impact assessment framework (QUIRC) [40] uses six risk criteria (confidentiality, integrity, availability, multiparty trust, mutual auditability and usability) and the Delphi evaluation [43] to aggregate opinions of expects on estimation of risk values per criteria. The weighted function is proposed for obtaining net security risk. Albakri et al., [42] proposed a usual qualitative approach, which also includes CSCs in the risk assessment process. K. Djemame [41] proposed to use a risk inventory to store risk profiles for several risks associated with specific assets. These risk profiles already contain semi-qualitative assessment values for probability and impact. Using these profiles, the authors show how risk could be changed dynamically depending on modifications in the cloud service. C.-A. Chin and Y.-L. Huang [39] proposed ACRAM (Adjustable Cloud Risk Assessment system), a risk assessment approach with an ad-hoc method for computing event probabilities based on various information (like the number of detected vulnerabilities, vulnerability score, several coefficients in the version that excludes the possibility for monitoring, and other parameters such as number of ports, number of packets, number of modified data, etc., in the version with monitoring facilities). Risks are computed per cloud resource (VMs, applications, physical machines).

We see that almost all of these methodologies and approaches require heavy involvement of an analyst to define a system model, identify basic risk components and relations between them, and estimate the main parameters (usually, likelihood and impact). This is not suitable for the main goal of the risk assessment in scope of MEDINA. First, such approaches require a lot of time and effort, while the main MEDINA's advantage is in its continuous monitoring, which requires rapid (nearly instant) risk assessment. Second, MEDINA cannot rely on the experience of the analyst because the CSP may have no good knowledge of security and risk assessment. Moreover, such assessment will be very subjective and provide a possibility for the owner to manipulate the results of the assessment. Last, but not least, risk assessment in MEDINA is used as a support for decision making about the compliance status and, thus, must be grounded in the certification scheme selected for evaluation (e.g., EUCS). Thus, the results of the evaluation must primarily depend on how many and how well controls of the scheme are implemented.

Some of the methods mentioned may address some of these issues (e.g., the risk assessment method proposed by P. Santini et al., [15] is based on the CIS top 20[8] critical security controls), but none can solve them all. The MONARC[5] and SPIDERISK[6] indeed simplify the process, but their process is still rather complex for MEDINA and requires involvement of an analyst. In short, we need a lightweight, automatic risk assessment method and process based on a cyber security certification scheme.

[8] https://www.rapid7.com/solutions/compliance/critical-controls/. Currently, the list of the top CIS Critical Security Controls is reduced to 18 (see https://www.cisecurity.org/controls/cis-controls-list for the most up to date list).

(cc) BY-SA

4 Risk Assessment Model

Our risk assessment model is based on the analysis of *cyber* security risk, i.e., potential events aiming to compromise *cyber* assets, primarily[9] from a cyber *security* point of view.

A risk assessment model usually (and our model, in particular) includes the following key components: Assets, Threats, and Vulnerabilities:

- **Asset** is a valuable (for the owner) object (including digital objects, like a service or data) which can be compromised by a threat.
- **"Threat** is a potential cause of an unwanted incident, which may result in harm to a system or organization" [44].
- **"Vulnerability** is a weakness of an asset or control that can be exploited by one or more threats" [44].

Our model aims to quantitatively estimate cyber security risks. In order to conduct such a quantitative analysis, the model estimates the possible *impact* of the successful compromising of an asset, the expected *frequency* of threats to arrive, and the *probability* of the threats to successfully exploit the existing vulnerabilities in the system. The model also defines relations between the outlined concepts to form a mathematical tool for computing risk values.

Our model can be split into three layers:

- **Conceptual layer**. This layer defines the main concepts and relations.
- **Domain layer**. This layer identifies the main concepts and relations for a specific domain (cloud service, in our case).
- **Individual layer**. This layer feeds input data about a considered system (i.e., cloud service) into the model and makes it possible to analyse it.

Having these three layers allows reusing the core parts (and the knowledge) of the model in different contexts, in contrast to other risk assessment models, which include information from different layers in a holistic approach.

The conceptual layer is the most generic one and defines only a mathematical structure of the model. The domain layer sets the parameters for a concrete domain in which the model will be applied, taking into account the domain-specific knowledge. The individual layer focuses on a concrete CSP and is thus relevant in the context of this CSP only.

4.1 Assets/Resources

A list of assets is one type of input the model requires for the computation of risk values. There are various types of assets which may be considered, and every domain may have its own list of typical assets. Since concrete assets are specific to every CSP and our model is aimed to be generic, the *conceptual layer* focuses on *asset types* rather than assets themselves. Asset types specify only the kind of assets we are considering; this allows defining relations between assets and threats without the knowledge of the specific service itself.

It is typical to consider the following three aspects of security, which could be compromised:

- Confidentiality,

[9] Some of the considered cyber security events (threats) may also be attributed to cyber dependability, rather than to cyber security, but it is often difficult to clearly split these aspects to consider them separately. Therefore, for completeness of the model (and also because certification schemes like EUCS include requirements to prevent such events), these events are also included in the model.

- Integrity, and
- Availability.

Some threats are mostly focused on targeting one of these aspects (e.g., ransomware and DoS attacks compromise Availability of data and a service), while others may have more diversified impact. In order to model this dependency, every asset type is associated with three possible impacts: Compromised Confidentiality; Compromised Integrity; and Compromised Availability.

Formally[10], we may see identified asset types as a vector *AT* with dimension $n_{AT} \in \mathbb{Z}^*$. A CSP will be asked to provide a list of assets *A* with values for estimated impact in case confidentiality A_C, integrity A_I, and/or availability is violated A_A (all vectors are of the dimension $n_A \in \mathbb{Z}^*$). These three impact vectors contain real values denoting the estimated impact. Also, the CSP explicitly links inserted assets with the asset types, which can be represented with a Boolean matrix ***AAT*** *of* ($n_A \times n_{AT}$) *in which every row contains all 0 except the for the selected attack type (value =1).*

At the *domain layer*, we focus on the asset types shown in Table 3. To select the most suitable asset types, we started with an internal study of FhG (MEDINA's partner) with the aim of identifying the typical resources for cloud services[11]. Moreover, the Clouditor tool[12] (provided by FhG and based on their study) has the capability to automatically detect some of the existing resources. This synergy is particularly important for the continuous monitoring phase, during which Clouditor manages a collection of evidence for specific resources and provides this information to the risk assessment component for the re-evaluation of risk.

Thus, we started with the list of resources identified by FhG. On one hand, within the scope of work on risk assessment needs for MEDINA, we found that some of these resources do not represent direct assets for CSPs (e.g., Identity Management, Account, etc.). Moreover, the FhG ontology is more detailed than what is required for conducting risk assessment, the security controls for securing these resources are the same, as are the threats targeting these resources. On the other hand, too much detail requires more work on the CSP side in setting up the risk assessment functionality. Thus, in order to simplify the risk assessment procedure, we reduced the amount of asset types to consider with respect to the FhG ontology (and comparing to the list of asset types listed in D2.6 [1]):

Table 3. Resources to Asset types mapping

FhG Resources	Asset types
Account	---
CI CD Service	CI CD Service
Compute. Container	Container
Compute. Function	Function
Compute. Virtual Machine	Virtual Machine
ContainerOrchestration	ContainerOrchestration
ContainerRegistry	ContainerRegistry
Identifiable	---
Identity Management	---
Image. Container Image	Image. Container Image

[10] In the formal notation used in this document, capital letters represent vectors (e.g., *A*), bold capital letters represent a matrix (e.g., **A**), and lower-case letters are members of these lists: $a_i \in A$ or $\boldsymbol{a_{ij}} \in \boldsymbol{A}$. Letters i, j, k, l represent non-negative integer numbers used as counters and $n_A \in \mathbb{Z}^*$ is always the number of items in vector *A* (or rows and columns in a corresponding matrix).

[11] A brief description of the FhG cloud ontology is provided in *APPENDIX: Cloud resource ontology*.

[12] The interested reader is referred to Clouditor technical specifications in the deliverable D3.5 [47].

(cc) BY-SA

FhG Resources	Asset types
Image. VM Image	Image. VM Image
IoT. Device Provisioning Service	IoT. Device Provisioning Service
IoT. Messaging Hub	IoT. Messaging Hub
Networking. Network Interface	Network
Networking. Network Security Group	Network
Networking. Networking Service. Load Balancer	Network
Networking. Virtual Network	Network
Networking. Virtual Sub Network	Network
Storage. BlockStorage	local storage
Storage. FileStorage	local storage
Storage. ObjectStorage	local storage
Storage. DatabaseStorage	Database
PasswordPolicy	---
	CSC trust

It is important to underline the importance of one specific asset added to the list of asset types: *CSC trust*. It relates to the specific damage caused by threats, especially those that cause damage to CSCs rather than to the CSP.

At the *individual layer*, a CSP is asked to:

- Provide a list of its main assets.
- Associate these assets with the defined asset types.
- Estimate the impact in case confidentiality, integrity or/and availability of an asset is compromised.
- Specify the approximate number of these assets.

This is CSP-specific knowledge (*individual layer*) and can only be provided by the CSP. Since a CSP may have many resources of the same kind (e.g., VMs), our model provides the opportunity for the CSP to set up the approximate number of every asset, instead of entering every asset separately. The expected impact is then integrated for all similar assets to obtain the valid entries for vectors A_C, A_I, and A_A.

It is important to note that a CSP having several assets of the same kind, but with different significance (or different expected losses for different impact types), still may (and should) report these assets separately for a more correct assessment. In other words, it is possible to enter several assets of the same type.

Running example

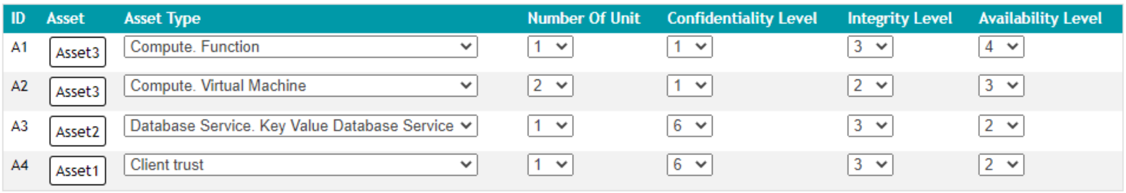
For better illustration of our risk computational model, we will use a running example. This example does not include all domain specific parameters defined in this document (and implemented by the supporting tool) because it would require a huge volume of data. Thus, the example is minimal and has the focus only on demonstration of the computations.

In our running example, we consider a simple SaaS service consisting of two VMs, one database and one Web application (function). First, the CSP is asked to provide the required information (see Figure 1).

For simplicity, we assume to consider only the following four ($n_{AT} = 4$) resource types, i.e., *AT*:

1) Compute. Virtual Machine

1) Database Service. Key Value Database Service
2) Compute. Function
3) CSC trust.



ID	Asset	Asset Type	Number Of Unit	Confidentiality Level	Integrity Level	Availability Level
A1	Asset3	Compute. Function	1	1	3	4
A2	Asset3	Compute. Virtual Machine	2	1	2	3
A3	Asset2	Database Service. Key Value Database Service	1	6	3	2
A4	Asset1	Client trust	1	6	3	2

Figure 1. Running example. Assets

Table 4 shows the relations between Assets and Asset types (**AAT**) .

Table 4. Running example. Assets to asset types mapping (AAT)

	Compute. Virtual Machine	Database Service. Key Value Database Service	Compute. Function	CSC trust
VMs	1	0	0	0
DB	0	1	0	0
Web app	0	0	1	0
Client trust	0	0	0	1

Finally, vectors A_C, A_I, and A_A are defined in Table 5.

Table 5. Running example. Impact values for CIA

	A_C	A_I	A_A
A1	1	100	1000
A2	1	10	100
A3	100000	100	10
A4	100000	100	10

Please note that our method requires quantitative values for impact, but for convenience of the user it has been decided to use a semi-quantitative scale (from 1 to 10). Thus, the values are converted to the quantitative ones with a simple formula: $a_{new} = 10^{a_{old}-1}$.

4.2 Threats

In our model, threats are considered as a predefined list of causes which may harm the specified assets. Although at the *conceptual layer* the model does not know which specific assets are present in the evaluated system, it is possible to establish a link between threats and asset types. Every threat is associated with its expected frequency.

We may see threats as a vector *T* with threats and a *TV* containing the expected frequencies (real values). Both vectors are of size $n_T \in \mathbb{Z}^*$.

Being predefined, threats should cover all the major threats for the considered domain (i.e., cloud in our case) and be specific enough to identify possible protection (during risk mitigation). Predefining the list of threats has advantages and disadvantages. The cons of such an approach are less burden for the CSP (and thus less reliance on the cyber security knowledge the CSP employees possess). This also helps to make our model more “automatic”. On the other hand, this does not allow a CSP to insert CSP-specific threats and, thus, the model loses a bit of its flexibility. This is the price we have to pay for making our model less reliable on CSP's experience in security.

At the *domain layer*, our model is populated with the threat causes which are listed below. Naturally, "external cyber attacker" cause is the biggest and most heterogeneous (from the point of view of the used tools and methods) and we need to address it in a more fine-grained way. To do that, we split all possible attacks related to this cause on the basis of the way the attacker penetrates into the system. Also, some attacks with specific impact are singled out (e.g., DoS and ransomware[13]).

Another important observation is that in the cloud environment, a system owner (CSP) also bears some responsibility for the security of its CSCs. Not only should the CSP make sure that its service is not compromised, but it should also do its best (and up to its capabilities) to prevent its CSCs to be compromised. Certification schemes, and EUCS in particular, require that a CSP implements certain security features to help its CSCs to secure themselves.

External cyber attacker

- *Account hijacking (CSCs or CSP)* – This threat relates to the attacks in which an external cyber attacker obtains the required credentials for entering the service. There are a number of ways of doing this for the attacker, including social engineering attacks (e.g., phishing), penetrating into the administrative system and installing a Trojan horse, eavesdropping the internal communication, etc. The ways for an attacker to obtain the credentials for accessing the system are beyond the scope of the assessed service, but the service may strengthen its identification and authentication policies (e.g., applying multi-factor authentication, better audit capabilities, etc.). Naturally, a CSP and its CSCs could be the targets of this threat.
- *Web-application threat: API, GUI, service vulnerabilities* – this threat includes all attacks first aiming at exploiting vulnerabilities in service GUI and APIs (e.g., SQL injection attacks).
- *Exploitation of metastructure (CSC or CSP)* – similar to the previous threats the attacker is assumed to exploit the vulnerabilities in the control plane components. Depending on whether the CSP provides a control plane to its CSCs (e.g., IaaS or PaaS provider) or consuming it (e.g., SaaS), this threat could be of a problem for the CSCs and the CSP.
- *Web-based attack* – this threat targets the users of the provided service, rather than its owners. The attacker has the goal of exploiting the service to perform some malicious functionality and attack the users. An example is Cross-site scripting.
- *CI/CD attacks* – this threat includes various attacks on the CI/CD pipeline with the goal to modify it (e.g., embed a backdoor or a malicious script).
- *Poor IAM (CSCs or CSP)* – this threat complements account hijacking but focuses on the ways to break through the Identification and Authorisation Management functionality (e.g., guessing weak passwords or exploiting a vulnerability in the IAM functionality allowing the attacker to log in into the system). Both CSCs and CSP could be a victim of such attacks.
- *Exploit Poor configuration (CSCs or CSP)* – an attacker may penetrate into the system exploiting poor configuration of the service (e.g., using default credentials, or getting access

[13] We would like to acknowledge that there is one more significant threat with a specific impact: Data breach. The problem of considering it as a separate cause is that most of the ways to penetrate into the system considered in our domain layer of the model (and also caused by causes others than "external cyber attacker") may lead to this type of impact. This does not allow to identify the security features and controls targeting to prevent exactly this type of threat.

to unsecured data storage). This could be a problem for a CSC, as well as for the CSP itself if it buys a service from a hyperscaler.

- *Ransomware* – ransomware is a popular threat nowadays. It is delivered by malware that once penetrated into the system encrypts information and demands a ransom to be paid for the ability to decrypt it. In our model, we focus only on the ransomware that hits the CSP itself (rather than targeting the CSCs and making them to substitute the data in the cloud with encrypted versions). The reason for not separating a version of such a dangerous and frequent threat for CSCs is that the certification schemes, and EUCS in particular, do not have specific requirements targeting and being very effective against such advanced threat.
- *DoS (CSCs or CSP)* – Denial of Service threat aims to bombard the selected service with a huge amount of requests that make the service unavailable for legitimate users. The attack may be launched against a specific CSC (e.g., a SaaS provider) or against the CSP itself.
- *Compromised Communication* – this threat aims to eavesdrop or tamper the communication between the service and the outer cyber world, or between services in the virtual networks. The attacker may find a way to decipher the communication (with no or weak encryption) or exploit vulnerabilities of the non-secure protocols.
- *On-site tampering/penetration* – this threat includes the attacks which start with an attacker physically tampering with the servers or administration network devices.

Other intentional threat causes:

- *Insider abuse* – this threat includes the malicious actions of an employee who uses its legitimate privileges for its own, unlawful purposes (e.g., copy private data).
- *Insider hacker* – in contrast (or in addition) to a simple abuse this threat considers a malicious employee of the CSP who further exploits the cloud service to compromise it.
- *Malicious CSC* – this threat is caused by a CSC which abuses the rights of the bought service to compromise the CSP or other CSCs of the CSP.
- *Unlawful CSC* – in contrast to a malicious CSC, the unlawful CSC does not target the CSP itself, but uses the bought cloud service for its unlawful purposes (e.g., running a spam service, distributing malware, etc.).
- *Malicious CSC employee* – this threat is similar to the malicious or unlawful CSC threats, but it is not the CSC itself that executes malicious actions, but merely some of the CSC's employees, i.e., against the CSC's will. The CSP itself can use its capability to help (or provide enough technical means) to the CSC to identify the malicious behaviour.
- *Third party problems* – this threat relates to any third party the CSP depends on, and which is willingly or unwillingly (supply chain attack) misbehaving.

Unintentional threat causes:

- *CSP's employee negligence and mistakes* – this threat relates to different ingenuous actions of employees which lead to a security breach (e.g., exposing sensitive information).
- *System glitch* – a technological problem (e.g., an integration issue or error reporting functionality) which compromises cyber security. Examples are an integration issue or error reporting which expose sensitive information (e.g., as a part of error messages or allowing public access).
- *Exhaustion of resources (CSC)* – insufficient allocation of resources for a CSC may become a security issue (especially, with respect to availability).

- *Unnecessary disclosure to law enforcement* – once the law enforcement agencies require access to the cloud service, the CSP should aim to reduce the amount of sensitive information shared with them, on the one hand, and be able to provide the required information, on the other one. Technical functionality should be available for preventing unnecessary disclosure.
- *Data location failure* – this threat relates to the data location issues. The CSP must make sure that data are physically located according to the contractual agreement and legal requirements.

Physical threats with impact on cyber security:

- *Hardware theft/loss (DC)* – physical theft of equipment, which may contain important information or be essential for provisioning of the service.
- *Environment threat (DC)* – various types of environmental threats causing physical damage to the cloud service (earthquakes, flood, fire, dust, etc.).
- *Physical threat (DC)* – physical damage of the hardware the service is running on.

As it is defined by the conceptual layer, every threat is to be associated with a real value representing the expected frequency of the attack (based on general statistics for cloud attacks). But there are differences in the threats targeting cloud market types (e.g., PaaS and IaaS providers should care more about the meta-interfaces, but they will not be affected by web-application attacks). Therefore, we need different lists of frequency values for different market types. A CSP (at *individual layer*) should provide the market type of its service, and the supporting tool will select the corresponding *TV* list with expected threat frequencies.

Running example

In the scope of our running example, we consider only two ($n_T = 2$) threats; T and TV are defined as follows:

T
Web-application threat
DoS

TV
4
0.5

4.3 Vulnerabilities/Requirements

In the scope of MEDINA, the main vulnerabilities for cloud services are the lack of implementation of the security requirements defined in the considered certification scheme (e.g., EUCS [4]). With this assumption we also assume that the certification scheme contains the main security features which can and should be installed to protect a cloud service. On the other hand, there is not a more comprehensive description of security features than a cyber security standard/certification scheme.

MEDINA assumes that every certification scheme contains a list of requirements, which can be grouped into security controls. We use this structure, in order to reduce the model. Let *R* be a list of all requirements and *RV* an associated Boolean list denoting if the corresponding requirement is fulfilled (1) or not (0). Both vectors are of size $n_R \in \mathbb{Z}^*$. Since the number of requirements can be very large, in order to make it manageable, we aggregate all the requirements up to the level of controls (using the relations established by the certification scheme itself). Let *C* be a list of all controls of size $n_C \in \mathbb{Z}^*$, and **RC** be a matrix $n_R \times n_C$, which contains real values from [0,1] interval, denoting the degree up to which a requirement *r* contributes to control *c;* the non-zero values are assigned only if requirement *r* belongs to control *c* in the selected certification scheme. Now, it is enough to multiply $(\boldsymbol{RC^T} \times RV)$ to obtain the degree of coverage for every requirement *c, i.e., CV* of size $n_C \in \mathbb{Z}^*$.

At the *domain layer* the model does not specify the requirements and controls, but retrieves them from the MEDINA Catalogue of controls and metrics[14], which is reported in D2.1 [45]. Currently, only the EUCS scheme is considered. At the *individual layer,* a CSP is asked to answer a questionnaire about the fulfilment of all requirements from a certification scheme. Currently, several possible answers are available for specifying the entity implementing the requirement, but they all the answers map to either 1 or 0. This is done for future differentiation between the entities which are responsible for implementing the requirement. In the future, different assessment tools could be used in order to monitor and confirm the initial input.

Answers rated as 1:

- Yes. CSP only
- Yes. Hyperscaler[15]
- Yes. Hyperscaler and CSP
- Not Applicable

Answers rated as 0:

- No

Running example

Consider only the following EUCS controls and requirements [4], with the provided answers, , shown in Table 6.

Table 6. Running Example. Controls and Requirements

Control	Requirements	Provided answer (RV)
OIS-01.1	The CSP shall define, implement, maintain and continually improve an information security management system	Yes. CSP only (1)
	The CSP shall document the measures for documenting, implementing, maintaining and continuously improving the ISMS.	No (0)
ISP-02.1	The CSP shall document a global information security policy	Yes. CSP only (1)
	The CSP's top management shall approve the security policies and procedures or delegate this responsibility to authorized bodies	Yes. CSP only (1)
OPS-05.1	The CSP shall deploy malware protection, if technically feasible, on all systems that support delivery of the cloud service in the production environment, according to policies and procedures	Yes. Hyperscaler (1)
	Signature-based and behaviour-based malware protection tools shall be updated at least daily	No (0)
IAM-02.1	The CSP shall document policies for managing accounts	Yes. CSP only (1)
	The CSP shall document and implement procedures for managing personal user accounts and access rights to internal and external	Yes. CSP only (1)

[14] This functionality is not implemented in the current version of the tool and is expected to be added in the future.

[15] Here and in the following, the hyperscaler is just an entity which provides the basic functionality for the CSP. In case CSP provides a service of hyperscaler it should implement all requirements itself.

	employees that comply with the role and rights concept and with the policies for managing accounts	

In short, in our running example we use only eight ($n_R = 8$) answers to eight requirements and four ($n_c = 4$) controls. $\boldsymbol{RC}$ is defined by experts and is embedded in the tool. Then, the coverage of controls *CV*= $(\boldsymbol{RC^T} \times RV)$ is computed as follows*:*

$$\overset{\boldsymbol{RC^T}}{\begin{bmatrix} 0.7 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0.6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0.7 \end{bmatrix}} \times \overset{\text{RV}}{\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}} = \overset{\text{CV}}{\begin{bmatrix} 0.7 \\ 1 \\ 0.7 \\ 1 \end{bmatrix}}$$

4.4 Relations

As it has been specified before, initially, we start with:

- Three vectors with estimated impact for assets (real values) A_C, A_I, A_A of the dimension n_A (provided by the CSP).
- A Boolean matrix ***AAT** of ($n_A \times n_{AT}$)* mapping assets to asset types (established by the CSP).
- A vector of frequencies (real values) *TV* containing the expected frequency (real values). Both vectors are of size $n_T \in \mathbb{Z}^*$ (defined at the domain layer of the model).
- A coverage of controls $CV = (RV \times \boldsymbol{RC})^T$ of size n_C (*RV* is provided by the CSP and ***RC*** is defined at the domain layer of the model).

Total loss per threat. First, we compute the total expected loss per a threat occurrence (a real value vector *TL* of size $n_T \in \mathbb{Z}^*$). In order to obtain it, the model defines 3 matrices $\boldsymbol{ATT_C}$, $\boldsymbol{ATT_I}$, $\boldsymbol{ATT_A}$ of size $n_{AT} \times n_T$, in which every cell denotes the probability that confidentiality, integrity or availability of an asset of type *at* is compromised if a threat *t* occurs*.* Note, that the matrix is defined for attack types, which makes it possible to define it at the domain layer (i.e., a CSP is not required to define these relations).

The total expected loss per threat could be computed using the following formula:

$$TL = \ (\boldsymbol{AAT} \ \times \boldsymbol{ATT_C})^{\boldsymbol{T}} \times A_C + (\boldsymbol{AAT} \ \times \boldsymbol{ATT_I})^{\boldsymbol{T}} \times A_I + (\boldsymbol{AAT} \ \times \boldsymbol{ATT_A})^{\boldsymbol{T}} \times A_A \qquad (1)$$

In short, the formula multiplies the expected loss of a single occurrence by the probability of the threat to impact the corresponding security aspect of the assets, and then sums the value for all assets and for the three security aspects to receive one value per threat. The result is a vector *TL* containing the expected losses per threat occurrence.

Survival probability of a threat. The next step is to find the probability of a threat to survive all implemented security features and reach its goal, i.e., to occur. This operation should result in a vector *TP* of size $n_T \in \mathbb{Z}^*$.

First, the model splits all security controls *C* in two lists: C' and C'' with $n_{C'} + n_{C''} = n_C$. *C'* includes all controls, which can be seen as *means* to reduce certain threats. In contrast, *C''* includes *management* controls, which aim to organise the right usage of the means by defining generic policies, assigning roles for employees responsible for certain cybersecurity tasks, establishing effective procedures for quick response to occurred incidents, etc. On the one hand, all controls from *C′′* are not specific for mitigating specific threats, but contribute to leveraging the capabilities of implemented means. On the other hand, these controls are not very effective if there are no concrete means from *C′* to fight a threat.

In order to model the effects of different controls, our model first computes a coefficient for management quality coef$_{MQ}$ (a real value) applying a weighted function for management controls. Let $W_{C''}$ be a vector with values from [0;1] interval of size $n_{C''}$ ($\sum_{\forall i} w_i^{c''} = 1$). Also, similar to *C, we* split the related vector of coverage value *CV* into *CV′ and CV′′*. Then,

$$coef_{MQ} = (W_{C''})^T \times CV'' \tag{2}$$

For *C′* we also put in correspondence a vector $W_{C'}$ with real values from [0;1], but in contrast to $W_{C''}$ $\sum_{\forall i} w_i^{c'}$ is not bound to be 1. Every value $w_i^{c'}$ from $W_{C'}$ denotes the guaranteed protection, i.e., a portion of protective capability of control c_i, which is guaranteed even if management is very poor (e.g., $coef_{MQ} \to 0$). Then, the model adjusts the protective capability of controls from $W_{C'}$, transforming *CV′* into *CVC′* using the following formula:

$$\forall i\ cvc'_i = cv_i * (w_i^{c'} + coef_{MQ} * \left(1 - w_i^{c'}\right)). \tag{3}$$

The model defines matrix ***RT*** of size $n_{C'} \times n_T$, in which every cell denotes the probability for a security control *c* to prevent a threat *t*. Once again, ***RT*** matrix does not depend on the values to be provided by CSP and, thus, can be defined at the domain layer.

The survival probability of a threat can be found as:

$$TP = CVC' \otimes RT, \tag{4}$$

where operation ⊗ is defined as follows (probabilistically):

$$\forall j\ tp_j = \prod_{\forall i}(1 - cvc'_i * rt_{i,j}). \tag{5}$$

Risk computation. Now we are able to compute the risk per threat *R* (a real-value vector of size $n_T \in \mathbb{Z}^*$) and the overall risk for the service (Risk). A risk per threat can be computed by multiplying the corresponding frequency, survival probability and expected total loss:

$$R = TV \odot TP \odot TL, \tag{6}$$

where ⊙ is a Hadamard multiplication, defined as:

$$\forall j\ r_j = tv_j * tp_j * tl_j. \tag{7}$$

The total risk (a scalar real value) is just a summation of risks per threat:

$$Risk = \sum_{\forall i} r_i. \tag{8}$$

The result of the computation represents the annual expected amount of losses for the CSP. The CSP should evaluate the received amount and decide if the estimated risk can be accepted, or a treatment option is to be applied.

In the context of the MEDINA project, the risk result is used as a parameter for evaluating the degree of non-conformity with the selected certification scheme. Table 7 lists the variables used in the model.

Table 7. Summary of the key variables used by the model

Variable	Domain	Dimension	Source/formula	Meaning
A_C, A_I, A_A	real	n_A	User input	Confidentiality, integrity, availability impact per asset
$\boldsymbol{AAT}$	{0;1}	$n_A \times n_{AT}$	User input	Asset to asset type mapping
TV	Real	n_T	Predefined values	Expected frequency per threat
RV	{0;1}	n_R	User input	Satisfaction (1) or failure (0) per requirement
RC	real	$n_R \times n_C$	Predefined values	Degree of contribution of a requirement to a control
CV	real	n_C	$(\boldsymbol{RC}^{\boldsymbol{T}} \times RV)$	Coverage per control
$\boldsymbol{ATT_C}$, $\boldsymbol{ATT_I}$, $\boldsymbol{ATT_A}$	real	$n_{AT} \times n_T$	Predefined values	The probability of a threat occurrence to compromise confidentiality, integrity, availability.
TL	real	n_T	$(\boldsymbol{AAT} \times \boldsymbol{ATT_C})^{\boldsymbol{T}} \times A_C +$ $(\boldsymbol{AAT} \times \boldsymbol{ATT_I})^{\boldsymbol{T}} \times A_I +$ $(\boldsymbol{AAT} \times \boldsymbol{ATT_A})^{\boldsymbol{T}} \times A_A$	Total expected loss per threat
CV’	Real	$n_{C'}$	Predefined part of *CV*: $CV' \cup CV'' = CV$	Coverage of means controls
CV’’	Real	$n_{C''}$	Predefined part of *CV*: $CV' \cup CV'' = CV$	Coverage of management controls
$coef_{MQ}$	Real	scalar	$(W_{C''})^T \times CV''$	Management quality Coefficient
CVC’	real	$n_{C'}$	$\forall i \; cvc'_i = cv_i * (w_i^{c'} + coef_{MQ} * \left(1 - w_i^{c'}\right))$.	Adjusted portability of means controls
RT	real	$n_{C'} \times n_T$	Predefined values	Probability for a means control to stop a threat
TP	real	n_T	$\forall j \; tp_j = \prod_{\forall i} (1 - cvc'_i * rt_{i,j})$	Threat survival probability
R	real	n_T	$\forall j \; r_j = tv_j * tp_j * tl_j$	Risk per threat
Risk	real	Scalar	$Risk = \sum_{\forall j} r_j$	Overall risk value

Running example

Table 8 shows how the mapping of asset types to threats **(ATT)** is defined by experts.

*Table 8. Running example. Asset types to threats mapping (**ATT**)*

	Web-app. Threat	DoS
Compute. Virtual Machine	0	0.9
Database Service. Key Value Database Service	0.6	0.2
Compute. Function	1	0.4
CSC trust	0.7	0.4

Thus, the expected loss per threat (**TL**) is computed as follows (following Equation 1):

$$\begin{array}{ccccccc} \textbf{AAT} & & \boldsymbol{ATT_C} & & A_C & & TL_C \\ \begin{array}{|c|c|c|c|}\hline 1&0&0&0\\\hline 0&1&0&0\\\hline 0&0&1&0\\\hline 0&0&0&1\\\hline\end{array} & \mathrm{X} & \begin{array}{|c|c|}\hline 0&0\\\hline 0.7&0\\\hline 0&0\\\hline 0.4&0\\\hline\end{array} & \mathrm{X} & \begin{array}{|l|}\hline 1\\\hline 1\\\hline 100000\\\hline 100000\\\hline\end{array} & = & \begin{array}{|l|}\hline 40000.7\\\hline 0\\\hline\end{array} \end{array}$$

$$\begin{array}{ccccccc} \textbf{AAT} & & \boldsymbol{ATT_I} & & A_I & & TL_I \\ \begin{array}{|c|c|c|c|}\hline 1&0&0&0\\\hline 0&1&0&0\\\hline 0&0&1&0\\\hline 0&0&0&1\\\hline\end{array} & \mathrm{X} & \begin{array}{|c|c|}\hline 0&0\\\hline 0.2&0\\\hline 0.4&0\\\hline 0.3&0\\\hline\end{array} & \mathrm{X} & \begin{array}{|l|}\hline 100\\\hline 10\\\hline 100\\\hline 100\\\hline\end{array} & = & \begin{array}{|l|}\hline 72\\\hline 0\\\hline\end{array} \end{array}$$

$$\begin{array}{ccccccc} \textbf{AAT} & & \boldsymbol{ATT_A} & & A_A & & TL_A \\ \begin{array}{|c|c|c|c|}\hline 1&0&0&0\\\hline 0&1&0&0\\\hline 0&0&1&0\\\hline 0&0&0&1\\\hline\end{array} & \mathrm{X} & \begin{array}{|c|c|}\hline 0&0.9\\\hline 0.6&0.2\\\hline 1&0.4\\\hline 0.2&0.4\\\hline\end{array} & \mathrm{X} & \begin{array}{|l|}\hline 1000\\\hline 100\\\hline 10\\\hline 10\\\hline\end{array} & = & \begin{array}{|l|}\hline 72\\\hline 928\\\hline\end{array} \end{array}$$

$$TL = TL_C + TL_I + TL_A = \{40144.7; 928\}.$$

Consider the controls C={OIS-01.1, ISP-02.1, OPS-05.1, IAM-02.1}. This set of controls is split in two:

- means C'={OPS-05.1, IAM-02.1} (with $CV' = \{0.7; 1\}$) and
- management controls C''={OIS-01.1, ISP-02.1} (with $CV'' = \{0.7; 1\}$).

First, following Equation 2, we compute the coefficient for management quality coef$_{MQ}$, assuming that the weights associated with these controls are $W_{C''} = \{0.6; 0.4\}$:

$$\begin{array}{ccccc} (W_{C''})^T & & CV'' & & \text{coef}_{\text{MQ}} \\ \begin{array}{|c|c|}\hline 0.6 & 0.4\\\hline\end{array} & \mathrm{X} & \begin{array}{|c|}\hline 0.7\\\hline 1\\\hline\end{array} & = & 0.82 \end{array}$$

Next, we compute the adjusted portability of means controls (with Equation 3), i.e., *CVC'*, assuming that the following weights are assigned to these controls $W_{C'} = \{0.9; 0.4\}$:

$$cvc'_1 = 0.7 * (0.9 + 0.82 * (1 - 0.9)) = 0.712;$$
$$cvc'_2 = 1 * (0.4 + 0.82 * (1 - 0.4)) = 0.892.$$

Next, we compute the survival probabilities (with Equation 5) for the two considered threats, assuming that the following control strength values ***RT*** have been defined:

*Table 9. Running example. Probabilities for a means control to stop threats (**RT**)*

	Web-App. attacks	DoS
OPS-05.1	0.2	0.3
IAM-02.1	0.3	0.1

$$tp_1 = (1 - 0.712 * 0.2) * (1 - 0.5 * 0.892) = 0.5697 * 0.554 \approx 0.685;$$
$$tp_2 = (1 - 0.712 * 0.3) * (1 - 0.1 * 0.892) = 0.7864 * 0.9108 \approx 0.722.$$

Finally, assuming that risk is computed using Equation 7 as:

$$risk_1 = 4 * 0.685 * 40144{,}7 \approx 109996;$$
$$risk_2 = 0.5 * 0.722 * 928 \approx 335.$$

The total risk is (by Equation 8) is as follows:

$$Risk = 109996 + 335 = 110331.$$

4.5 Non-conformity analysis

Our model performs the analysis quantitatively, but its further analysis may be simplified for users by converting it into a value from [0;100] interval. This is done with a *10*log₁₀(Risk)* operation. Naturally, this maps a value only from (0;10 000 000 000] to the determined interval. This limit was selected on the basis of analysis of real-world losses to accommodate all of them, but it can be extended or shortened if required.

Let the result of risk computation using the input from the CSP be $risk_{real}$. Once a risk level is calculated, it is possible to perform another risk assessment for the full coverage of the selected certification scheme ($risk_{ideal}$). The full coverage may depend on the assurance level if the selected certification scheme has several assurance levels (e.g., as EUCS does). The difference ($risk_{real}$- $risk_{ideal}$) estimates the degree of the non-conformity. If this degree is higher than a defined threshold the non-conformity is considered as major, and this may lead to the revocation of a certificate. It is important to note that the mapping of a real value result to [0;100] is performed using a logarithm operation, and thus allows to evaluate the ratio between risks, i.e., in how many times $risk_{ideal}$ is better than $risk_{real}$. For example, the difference $risk_{real}$ - $risk_{ideal}$ =10 means that $risk_{ideal}$ is 10 times lower than $risk_{real}$.

This is the first rough strategy applied for a non-conformity analysis using risk assessment used by our model. We will evaluate it, set up the required threshold and correct it (if required) in the next year of the project. At this point, the threshold is planned to be the same for all CSPs. Nevertheless, we would like to note that this threshold will evaluate the *ratio* between the ideal and real scenarios, rather than absolute difference. Thus, a CSP with expensive assets and another one with less sensitive ones could be compared using the same threshold. At the same time, it is worth noting that the threshold does not simply represent the targeted reduction in probability level (thus, leaving aside the cost of assets), but it focuses on the risk (i.e., a product of probability and impact) ratio.

Running example

The real value of risk mapped to [0;100] interval is as follows:

$$Risk_{real} = 10 * \log(110331) \approx 50.43.$$

The ideal risk for our running example can be computed as follows. First, we need to re-compute TP with complete coverage, i.e., *CVC'={1;1}*:

$tp_1^{ideal} = (1 - 1 * 0.2) * (1 - 1 * 0.5) = 0.4;$
$tp_2^{ideal} = (1 - 1 * 0.3) * (1 - 1 * 0.1) = 0.63.$

Other values do not change, and we can compute $Risk_{ideal}$ as:

$$Risk_{ideal} = 10 * \log(4 * 0.4 * 40144{,}7 + 0.5 * 0.63 * 928) \approx 48.01.$$

If the defined threshold is 0.1, then $Risk_{real}$-$Risk_{ideal} = 2.42 < 0.1$ and the detected non-conformity is considered to be too high for certification (i.e., the non-conformity is *Major*).

4.6 TOMs Optimisation

During the preparation phase, if a non-conformity is detected, the CSP must decide what to do next. Naturally, if no non-conformity is detected, the CSP should apply for a certificate. If the non-conformity is minor, it is possible for the CSP to apply for a certificate anyway and use the result of the analysis to prove insignificance of the deviation. Various options are possible at this point, but they are out of scope of this deliverable. In this section, we will focus on a situation when the detected non-conformity is major, i.e., the current security configuration is not robust enough to be certified[16].

4.6.1 Optimisation problem definition

In short, we consider a situation in which a CSP detects a non-conformity with several requirements and needs to determine which of them must be implemented. The most obvious answer is: *all failed requirements* must be implemented. On the other hand, there could be some constraints on implementation of the missing requirements. For example, the available budget does not allow for implementing all failed requirements. Alternatively, implementing all missing requirements could be more costly than accepting the risk.

Let R be a set of all requirements and $\bar{R}$ ($n_r = |\bar{R}|$) be a set of failed requirements. Let also X be a Boolean set ($n_r = |X|$) which sets $x_i = 1$ if $\bar{r}_i$ is satisfied. Naturally, there could be different variations of X, and they are denoted with X_j. Initially, we start with X_0: $\forall\, x_i \in X_0.\, x_i = 0$.

Let also every r_i require c_i investments for its implementation ($n_r = |C|$). Finally, let B be the budget limit and $Risk(X_j)$ be the overall risk value computed using the computational method above with all initially satisfied requirements ($R \backslash \bar{R}$) and those initially failed requirements satisfied according to X_j.

In this case, the CSP faces an optimisation problem, which can be formalised either as:

find X_j $which$ $\begin{cases} \min Risk(X_j) \\ \mathrm{B} > \sum_{i=0}^{n_t} x_i c_i \end{cases}$

or

find X_j $with$ $\min\left(Risk\left(X_j\right) + \sum_{i=0}^{n_t} x_i c_i\right)$

The first optimisation problem aims to look for such subset of $\bar{R}$ (defined by X_j) which minimises the risk value, but keeps the overall cost of additional controls below the budget limit B. The second optimisation problem does not require the budget limit and simply looks for the most cost-balanced configuration.

[16] Also, in case of a minor non-conformity, the CSP may think about improving its security configuration by using the optimization approach proposed in this section.

4.6.2 Optimisation solution

The optimization problems stated above can be solved with a Genetic Algorithm approach [46] [47]. This approach finds the nearest optimal solution in a very short time. Our solution is mainly based on the Genetic Algorithm, but makes little adjustments to set it up for our problem.

The algorithm starts by randomly generating an initial population of chromosomes (our X_j), i.e., $X_{j1}, X_{j2}, X_{j3}, \dots$ are generated by randomly setting some bits to 1 or 0. This population will be further used for the generation of new population. Every time new population is generated the most fitting (according to the defined criteria) chromosomes are to be used further, while others are rejected. In this way chromosomes evolve with each iteration.

The generation of the new population is performed with two operations: crossover and mutation. First, crossover takes two chromosomes from the current population (called parents) and generates a new chromosome by taking pieces of the two initial chromosomes. The technique may use several points denoting the chromosome to be used (see Figure 2).

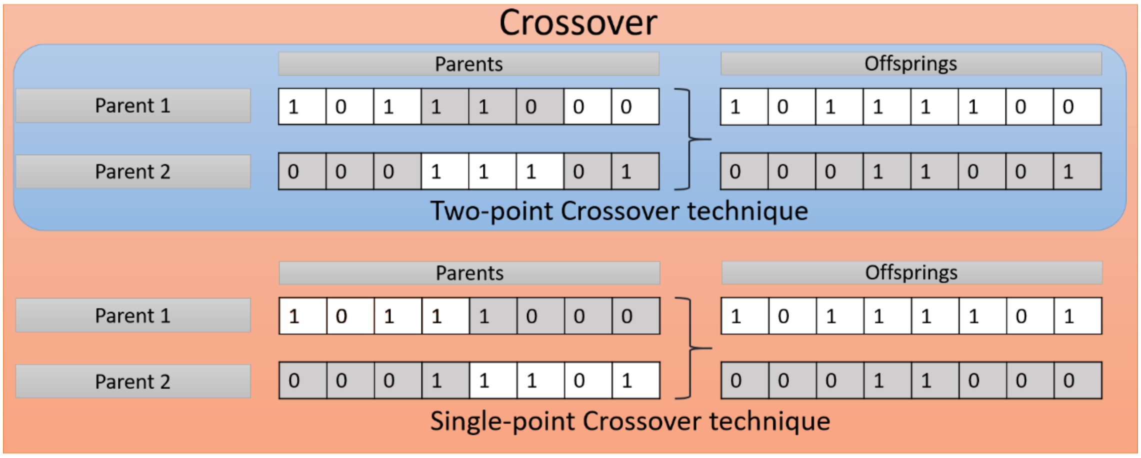


Figure 2. Single- and two-point crossover technique [48]

Next, the chromosomes are compared (according to the criteria) and the best ones are mutated. The mutation switches a pre-defined number of several random bits (see Figure 3). This technique allows avoiding local minimums.

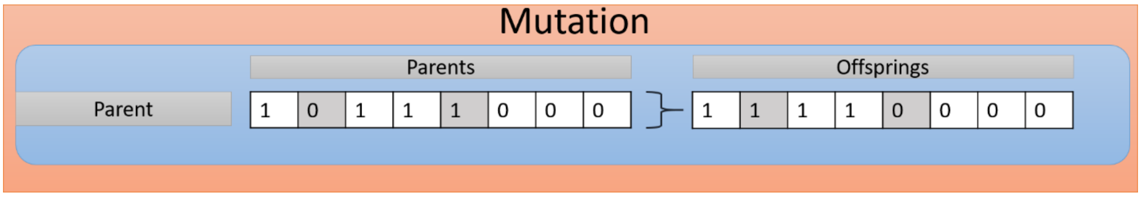


Figure 3. Mutation technique [48]

Then, the fitness criteria are applied again.

Once all chromosomes from the old population have been processed, a new population is generated and the process repeats.

After a pre-defined number of evolution cycles the process stops and the best (according to the criteria) chromosome is selected as output.

5 Implementation

The risk assessment model described in Section 4 is supported by a Risk Assessment and Optimisation Framework (RAOF) which implements the defined functionality. This deliverable reports the second version of the tool implementing the described model.

It is important to underline once again that the main goal of the RAOF in MEDINA is to evaluate the degree of non-conformity of the service with the selected certification scheme. This analysis should be performed using the assessed risk as a core functionality. That is why this deliverable is focused on defining and implementing the risk assessment (model).

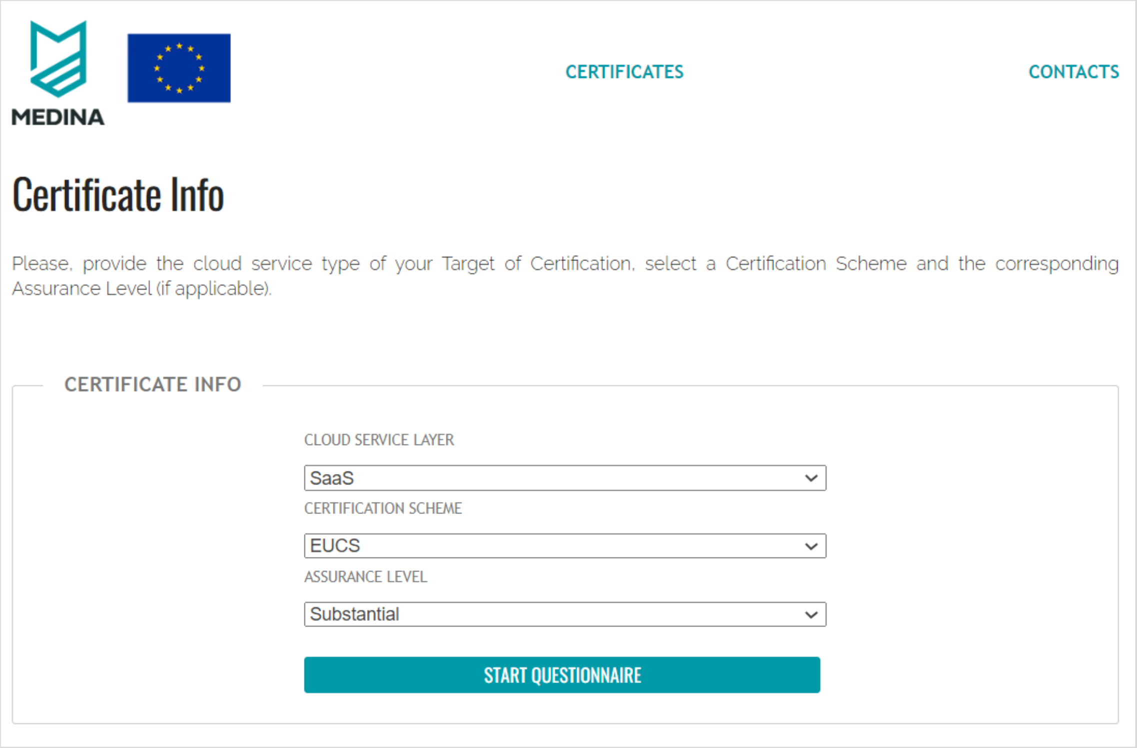
5.1 Functional description

The RAOF is implemented as a service which is able to quickly perform risk assessment and use this information to analyse the degree of non-conformity with the selected certification scheme.

5.1.1 Risk assessment

The tool provides both GUI and API for interaction. The GUI is created for direct interaction with the tool by a human operator (e.g., compliance manager). The operator is asked to provide the required information:

- General information, like the service market type, the selected certification scheme, and the assurance level (see Figure 4).
- A list of assets, lined with the defined asset types, approximate number of similar assets, and expected loss if Confidentiality, Integrity or Availability of these assets is compromised (see Figure 5).
- The information about implemented requirements of the selected certification scheme, e.g., EUCS (see Figure 6).



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

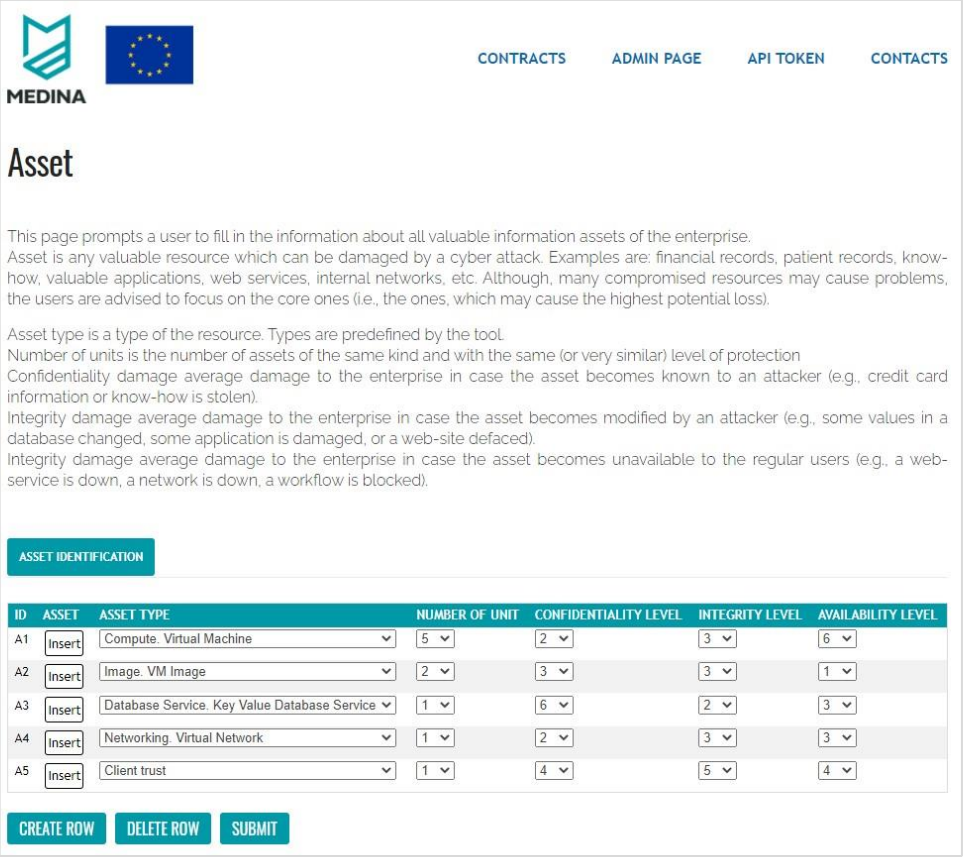
ASSURANCE LEVEL: Substantial

Figure 4. ToE setting page

The current version of the tool uses all the information inserted on this web page. The cloud service layer is used to define which threats are more/less relevant for the service. At this point, we consider three layers: SaaS, PaaS, and IaaS. In the future, mixed layers could also be considered.

The tool has also the possibility to select another certification scheme, but since MEDINA is focused on EUCS only, this possibility is not used and only EUCS scheme [4] can be selected. In the future, more schemes could be supported.

The assurance level is linked to the target assurance level of EUCS, i.e., Basic, Substantial, or High. After this selection, the tool will automatically filter the questions related to the requirements of the selected assurance level and conduct the non-conformity assessment targeting this level, i.e., considering only the requirements of the selected level.



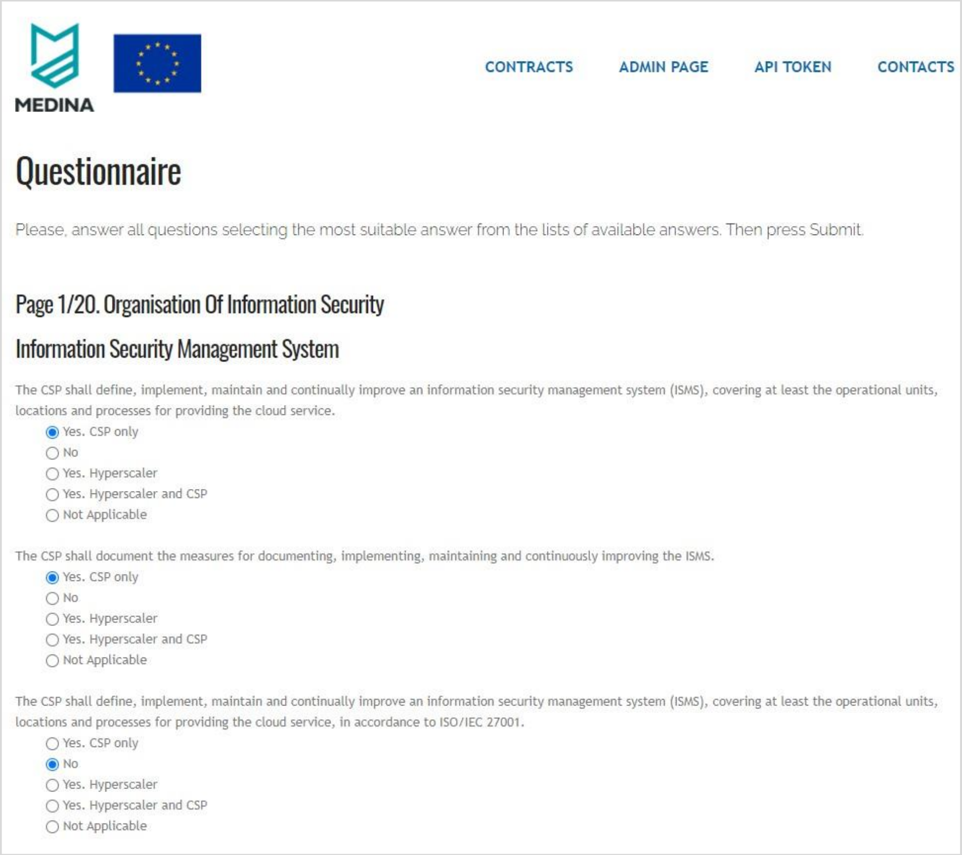
Asset

This page prompts a user to fill in the information about all valuable information assets of the enterprise.
Asset is any valuable resource which can be damaged by a cyber attack. Examples are: financial records, patient records, know-how, valuable applications, web services, internal networks, etc. Although, many compromised resources may cause problems, the users are advised to focus on the core ones (i.e., the ones, which may cause the highest potential loss).

Asset type is a type of the resource. Types are predefined by the tool.
Number of units is the number of assets of the same kind and with the same (or very similar) level of protection
Confidentiality damage average damage to the enterprise in case the asset becomes known to an attacker (e.g., credit card information or know-how is stolen).
Integrity damage average damage to the enterprise in case the asset becomes modified by an attacker (e.g., some values in a database changed, some application is damaged, or a web-site defaced).
Integrity damage average damage to the enterprise in case the asset becomes unavailable to the regular users (e.g., a web-service is down, a network is down, a workflow is blocked).

ID	ASSET	ASSET TYPE	NUMBER OF UNIT	CONFIDENTIALITY LEVEL	INTEGRITY LEVEL	AVAILABILITY LEVEL
A1	Insert	Compute. Virtual Machine	5	2	3	6
A2	Insert	Image. VM Image	2	3	3	1
A3	Insert	Database Service. Key Value Database Service	1	6	2	3
A4	Insert	Networking. Virtual Network	1	2	3	3
A5	Insert	Client trust	1	4	5	4

Figure 5. Asset table



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

The CSP shall define, implement, maintain and continually improve an information security management system (ISMS), covering at least the operational units, locations and processes for providing the cloud service.

● Yes. CSP only
○ No
○ Yes. Hyperscaler
○ Yes. Hyperscaler and CSP
○ Not Applicable

The CSP shall document the measures for documenting, implementing, maintaining and continuously improving the ISMS.

● Yes. CSP only
○ No
○ Yes. Hyperscaler
○ Yes. Hyperscaler and CSP
○ Not Applicable

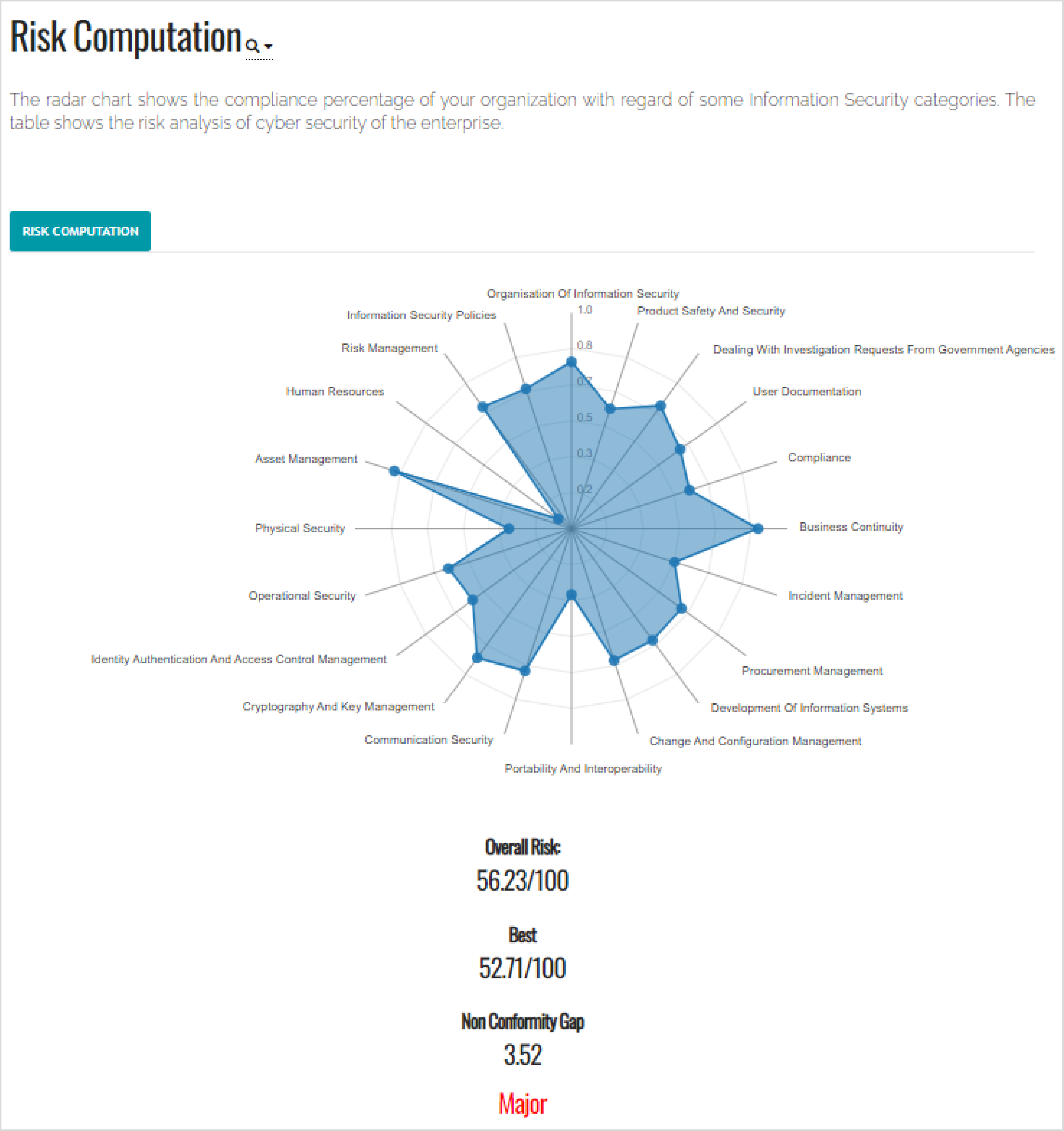
The CSP shall define, implement, maintain and continually improve an information security management system (ISMS), covering at least the operational units, locations and processes for providing the cloud service, in accordance to ISO/IEC 27001.

○ Yes. CSP only
● No
○ Yes. Hyperscaler
○ Yes. Hyperscaler and CSP
○ Not Applicable

Figure 6. Questionnaire

Once the inputs are provided by a CSP, the tool will calculate the risk level according to the procedure defined in Section 4. The result is displayed to the CSP (see Figure 7). The CSP may see the computed risk level and non-conformity evaluation result (minor/major non-conformity).

As it was described in Section 2, the CSP may perform several rounds of the analysis to determine the less risky configuration of its security if full conformity with the selected certification scheme is impossible or not required.



Risk Computation

The radar chart shows the compliance percentage of your organization with regard of some Information Security categories. The table shows the risk analysis of cyber security of the enterprise.

RISK COMPUTATION

Organisation Of Information Security
Product Safety And Security
Dealing With Investigation Requests From Government Agencies
User Documentation
Compliance
Business Continuity
Incident Management
Procurement Management
Development Of Information Systems
Change And Configuration Management
Portability And Interoperability
Communication Security
Cryptography And Key Management
Identity Authentication And Access Control Management
Operational Security
Physical Security
Asset Management
Human Resources
Risk Management
Information Security Policies
1.0
0.8
0.7
0.5
0.3
0.2

Overall Risk:
56.23/100

Best
52.71/100

Non Conformity Gap
3.52

Major

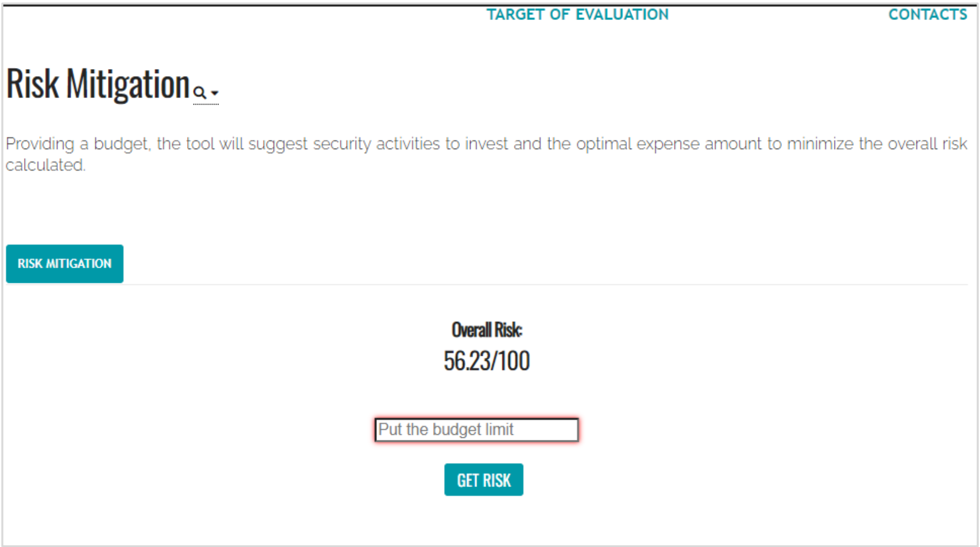
Figure 7. Risk Assessment result page

The tool also implements APIs for the integration with CSP's dashboards. The input information is to be provided through these APIs exploiting ways to collect the information more suitable for a CSP. This also allows re-usage of the information already contained in the CSP's system. Also, the APIs are required for performing automatic risk assessment during the continuous monitoring phase, but this functionality is discussed in dedicated deliverables D4.4 [2] and D4.5 [3] and as a part of integration of the tool to the overall MEDINA framework (see D5.2 [49]).

In short, our tool proposes a simple and fast way to assess risk for a cloud service, without reliance on the CSP's deep knowledge of cyber security. The user is only assumed to know well its own service. The risk assessment model and tool are tailored for the use in the cloud service domain, considering cloud specific threats, market types, and specific (vertical) relations between a CSP, hyperscaler, and CSCs. Last, but not least, the model and the tool are defined for supporting compliance checking and perform the risk assessment using the selected certification scheme, thus evaluating risk using a scheme-specific point of view.

5.1.2 Risk optimisation

Once the risk assessment phase is over, the user can use the implemented risk optimisation facility by clicking the "Optimisation page" button. The tool executes two types of optimisation analysis, as it is specified in Section 4.6.1. The first optimisation problem requires entering the available budget limit and searches for the requirements which minimise the risk for the CSP (see Figure 8). It is also possible to change the expected cost of implementing every requirement, if the CSP expects different cost than the one pre-set.



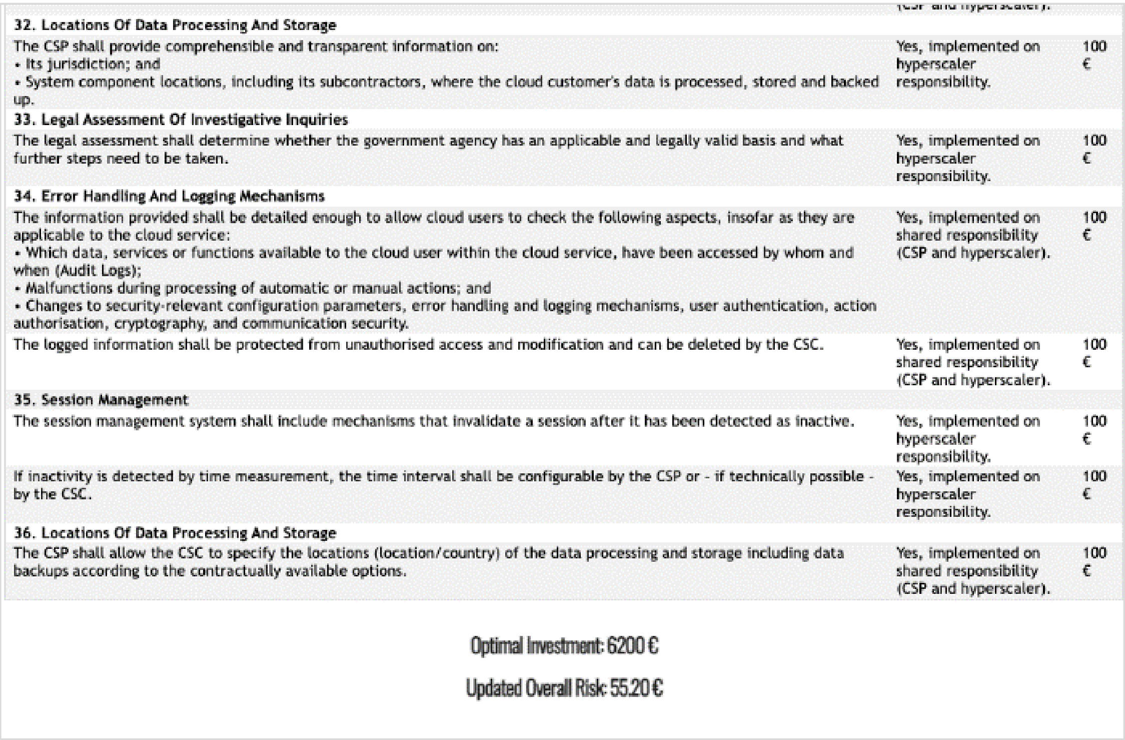
Risk Mitigation

Providing a budget, the tool will suggest security activities to invest and the optimal expense amount to minimize the overall risk calculated.

Overall Risk
56.23/100

Figure 8. Optimisation setting page

As a result, the tool returns a set of requirements to implement, the updated overall risk level, and the optimal investment cost (see Figure 9).



	(CSP and hyperscaler).	
32. Locations Of Data Processing And Storage		
The CSP shall provide comprehensible and transparent information on: • Its jurisdiction; and • System component locations, including its subcontractors, where the cloud customer's data is processed, stored and backed up.	Yes, implemented on hyperscaler responsibility.	100 €
33. Legal Assessment Of Investigative Inquiries		
The legal assessment shall determine whether the government agency has an applicable and legally valid basis and what further steps need to be taken.	Yes, implemented on hyperscaler responsibility.	100 €
34. Error Handling And Logging Mechanisms		
The information provided shall be detailed enough to allow cloud users to check the following aspects, insofar as they are applicable to the cloud service: • Which data, services or functions available to the cloud user within the cloud service, have been accessed by whom and when (Audit Logs); • Malfunctions during processing of automatic or manual actions; and • Changes to security-relevant configuration parameters, error handling and logging mechanisms, user authentication, action authorisation, cryptography, and communication security.	Yes, implemented on shared responsibility (CSP and hyperscaler).	100 €
The logged information shall be protected from unauthorised access and modification and can be deleted by the CSC.	Yes, implemented on shared responsibility (CSP and hyperscaler).	100 €
35. Session Management		
The session management system shall include mechanisms that invalidate a session after it has been detected as inactive.	Yes, implemented on hyperscaler responsibility.	100 €
If inactivity is detected by time measurement, the time interval shall be configurable by the CSP or - if technically possible - by the CSC.	Yes, implemented on hyperscaler responsibility.	100 €
36. Locations Of Data Processing And Storage		
The CSP shall allow the CSC to specify the locations (location/country) of the data processing and storage including data backups according to the contractually available options.	Yes, implemented on shared responsibility (CSP and hyperscaler).	100 €

Optimal Investment: 6200 €

Updated Overall Risk: 55.20 €

Figure 9. Optimisation result page

The second optimisation problem looks for the most optimal selection of requirements by optimising the overall expenditure, i.e., the sum of risk and additional cost. In order to run this analysis, it is enough to enter 0 as the target budget.

5.1.3 Requirements

The requirements from the deliverable D5.2 [49] that are relevant for this tool are listed below and their status is evaluated.

Requirement id	RBSCF.01
Short title	Risk assessment tool
Description	The tool shall be based on a risk-assessment methodology and in order to help CSP, as well as an auditor, to identify the key assets, threats and existing weaknesses of the cloud system.
Implementation status	Partially implemented

The tool is mostly implemented and is based on the identification and assessment of assets, threats and vulnerabilities of a cloud service. Some final changes, related to the integration of the tool into MEDINA platform could be added before finalising this requirement.

Requirement id	RBSCF.02
Short title	Risk assessment tool and TOMs
Description	Identification of key assets, threats and existing weaknesses should support stakeholders in reflecting their chosen TOMs in accordance to their risk strategy, along with risk treatment options.
Implementation status	Fully implemented

The tool performs risk assessment using the chosen TOMs and allows selecting the most appropriate ones according to the risk strategy of the CSP.

Requirement id	RBSCF.03
Short title	Implementation selection functionality
Description	MEDINA proposes a tool-supported methodology for the selection of controls and associated TOMs, which address the concrete needs of a CSP taking into consideration both its risk appetite and requested certification's assurance level.
Implementation status	Partially implemented

The first version of the optimisation functionality is added to this version of RAOF. The tool provides a support for optimisation of the TOMs selection (with/without a limited budget).

Requirement id	RBSCF.04
Short title	Interface to the auditor
Description	Auditor follows a risk-based approach which provides flexibility to the certification process: since an ever-changing threat landscape often requires timely reaction from the security team provoking changes in the security configurations. These could be efficient from the risk treatment point of view, but will affect the previously obtained certificate, in the worst case, invalidating it.
Implementation status	Discarded

This requirement is obsolete, since no specific access for an auditor is required. It has been decided that an auditor will use the same access facility which are provided to a compliance manager.

5.1.4 Fitting into overall MEDINA architecture

The RAOF service is involved in the preparation phase or risk assessment, in order to help the CSP to prepare its system for certification, as well as in the continuous monitoring phase, in which the CSP system is continuously monitored to verify its conformity with the selected certification scheme.

Figure 10 shows the initial plan to integrate RAOF into the MEDINA's architecture. During the preparation phase a *compliance manager* directly (via GUI) or through the *CSP's Compliance Dashboard* (via API) connects to the RAOF and provides the information about the service to be assessed, main assets, and satisfied requirements for the selected certification scheme.

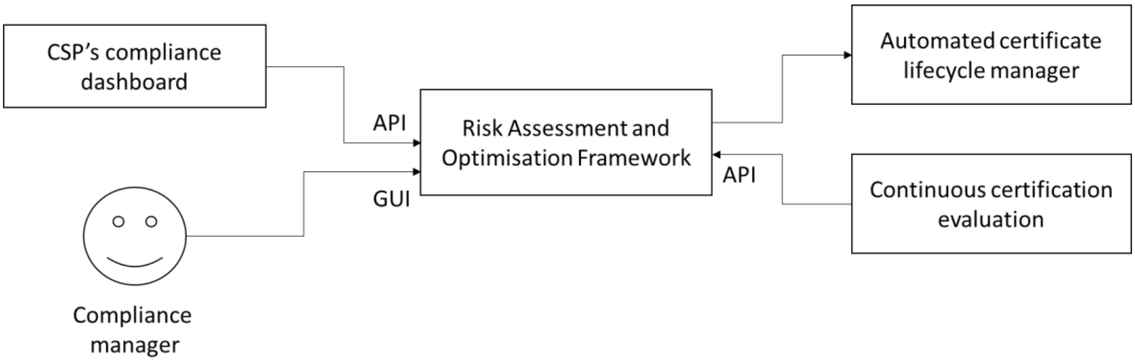

CSP's compliance dashboard
API
GUI
Compliance manager
Risk Assessment and Optimisation Framework
API
Automated certificate lifecycle manager
Continuous certification evaluation

Figure 10. A part of the MEDINA's workflow

During the continuous monitoring phase, once a non-conformity is detected, the *Continuous Certification Evaluation* module invokes RAOF in order to evaluate non-conformity[17]. RAOF performs the analysis and returns the results of its assessment to the *Automated Certificate Lifecycle Manager* for further decisions on the certification status.

5.2 Technical description

This section provides technical details about the internal structure of the RAOF.

5.2.1 Prototype architecture

The RAOF consists of the following three components (see Figure 11):

- A *Risk storage* database.
- *Main engine* with
  - GUI
  - Risk assessment module
  - Risk-based decision support
- *APIs*
- *Risk Optimiser*

Once *API* or *GUI* is contacted and the required information is provided (see Section 5.1.1), the *Risk assessment module* is invoked. Using the information from the *Risk Storage* database it executes the procedure defined by the risk assessment model (see Section 4). The *Risk-based*

[17] The interested reader is referred to the deliverable D4.4 [2].

decision support module implements the functionality of evaluating the non-conformity degree using the real and ideal results of risk assessment. The *Risk Optimiser* module supports the selection of the optimal requirements.

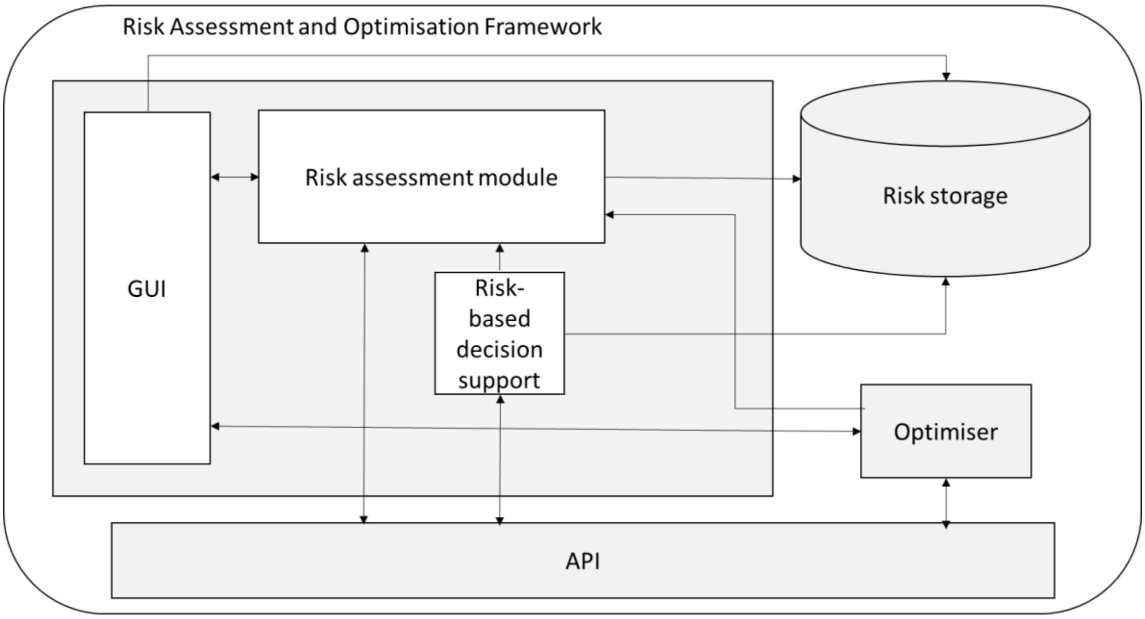

Risk Assessment and Optimisation Framework
GUI
Risk assessment module
Risk-based decision support
Risk storage
Optimiser
API

Figure 11. Internal architecture of the Risk Assessment and Optimisation Framework

5.2.2 Description of components

The ***Risk storage*** database keeps the user data and the information required for the correct operation of the tool. First, it contains the access information about the user, its risk assessment practices (for its services), and input values for every such practices (i.e., the information about the service to be assessed, selected certification scheme, the status of requirements, and assets with supporting information). Second, the database contains the predefined mapping tables and vectors required by the model (see Section 4.4). Finally, it also stores the information required for the correct representation of the information by the GUI (e.g., order of elements, structure of the questionnaire, type of elements for gathering inputs from users, etc.).

The ***GUI*** provides a user-friendly way for providing input to the tool and displaying its output. It guides the user through all the steps, collecting the information about the service to be assessed, and shows the final result. The GUI is dynamic and is governed by the information stored in the database (e.g., requirements). In the next steps of the project we are going to make it even more dynamic, by making it to be formed using the information retrieved from the Catalogue of controls and metrics.

The ***Risk assessment*** module is the main computation engine, which implements the computations according to the model described in Section 4. It uses the information provided by the user and the pre-defined knowledge stored in the database. The result of the execution of this module is the risk values (one per threat and the overall one).

The ***Risk-based decision support*** component is aimed to further process the results of the risk assessment produced by the risk assessment module. In particular, in the scope of MEDINA it will compute and analyse the degree of non-conformity according to the ideas described in Section 4.5.

The ***Risk optimiser*** component has been added to the second version of the tool and provides the functionality for optimisation of investments in order to obtain the most efficient coverage of requirements for a scheme (in case the complete coverage is not possible).

Finally, the ***API*** component defines the interfaces for the interaction of other modules with RAOF. In particular, a compliance manager may send commands to RAOF through a proprietary dashboard. Also, APIs will be used during the continuous monitoring phase, during which the *Continuous Certification Evaluation* component will invoke the RAOF and provide the results of monitoring for specific assets. The RAOF will conduct its non-conformity analysis automatically and send the results to the *Automated Certificate Lifecycle manager* (this functionality is reported in deliverable D4.4 [2]).

5.2.3 Technical specifications

Currently, the latest version of RAOF (SATRA) is reachable via the common MEDINA's testing facility[18] using the following url: https://integrated-ui-test.k8s.medina.esilab.org/satra [internal use only - authentication required]. The APIs can be found using the following url: https://risk-assessment-app-test.k8s.medina.esilab.org/api/v1/.

The project is deployed using three docker containers, each one running its own service and implementing separate functionality. The main service implements the core computational engine and the GUI. It is run over a Tomcat 8 and is running on Apache2 Web Service. The backend of this service is developed in Java, using the Springboot 5 framework. The front end uses JSP, HTML, Javascript and CSS.

The main service requires a database to store the basic domain layer settings of the model and user input values. The MySQL DBMS runs in a separate docker container.

The third service consists of Python REST APIs realised with swagger documentation that communicate with the main service to perform computations according to the defined model and retrieve user data by automatic means (e.g., CSP's dashboard) or from a monitoring component.

[18] Authorization is required for access.

6 Delivery and Usage

6.1 Package information

Table 10 shows the structure of the "Risk-Assessment-tool" project, which is divided into three folders called "-engine" that contain the code of the GUI and the computational logic (risk assessment module, risk-based decision support, and risk optimiser) developed in Java; the API interfaces folder called "app" developed in python with swagger documentation; and the databases backup folder called "db". Table 11 provides an overview of the RAOF component packages.

Table 10. 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).
-dmm	Source code used to implement the risk organisation

Table 11. Overview and description of package

Package	Description
API	
api/	Contain the source code for the API interfaces.
api.endpoints/	Contain all endpoint versions for the API interfaces.
api.endpoints.v1/	Contain 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.
iit.cnr.it.wentool/	Contains the source code to perform the risk analysis and manage input and output of this operation.
iit.cnr.it.wentool.computation/	Contains the code to compute the risk analysis.

BY-SA

Package	Description
iit.cnr.it.wentool.computation.riskanalysis/	Contains the code to execute the 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.
Dmm	
dmm/	Contains the code to optimise risk.

6.2 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[19]. 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
sudo docker run -dp 5000:5000 risk-assessement-app
```

- For Windows:

```
cd -app /
```

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

```
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
```

6.3 User manual

At this moment there is no detailed user manual for the implemented tool. The use of the tool with the GUI interface is intuitive and self-explanatory and is supported by descriptions explaining the steps in the process. The manual for API usage is to be implemented in the future when the APIs are finalised.

6.4 Licensing information

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

6.5 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

7 Conclusions

This deliverable reports the main achievements of the Task 2.6. First, we describe how the risk assessment may contribute to the compliance management process and ensure that it focuses on the real need of the CSP instead of mere fulfilment of the requirements from the chosen certification scheme. This strategy will be implemented in the MEDINA framework.

Second, we present in detail our model for risk assessment for cloud services rooted in the selected certification scheme. The model could be split in three layers: conceptual (raw mathematical structure), domain (pre-filled with domain-specific cyber-security dependent knowledge) and individual (knowledge about a concrete system). This deliverable explains in detail the conceptual layer, and provides some details about the cloud-specific settings. The knowledge for the individual layer is to be provided by a concrete CSP. This deliverable also adds the optimisation functionality to the tool, supporting the CSP in the process of improving their cybersecurity configuration and preparing the system for certification.

Finally, we provide the second version of the prototype for risk assessment and analysis (RAOF), which is set up for supporting cyber security compliance management for cloud service. The supporting tool is based on the defined model and is integrated in the overall MEDINA framework.

Our risk assessment approach (and tool) is yet not without limitations. First, as it was discussed at the beginning of this document, our risk assessment does not have the goal to substitute the usual risk assessment, which should be performed with more details and care. Second, an approach which tries to automate a process and aims to be applied in the same way for all cases almost always sacrifices depth for simplification. Yet, the same logic is used by the certification process itself, so, this sacrifice should not affect much the overall process of MEDINA. Third, we acknowledge that the estimation of the expected damage in case of compromising confidentiality, integrity, and availability could be a difficult task for a (especially, unexperienced) CSP, but these values depend very much on the CSP and cannot be pre-set for all CSPs. Our approach also tries to simplify this task by limiting possible values up to 10 levels. Finally, the optimisation task requires knowledge of the costs of implementation of all failed requirements. We acknowledge that this could be a hard task. On the other hand, we expect that only a limited amount of requirements will fail (otherwise, the major non-conformity would be clear without the risk assessment) and the CSP should have in mind anapproximate cost of applying the required corrections.

This is only the second version of the risk assessment model and tool. At the model level, we will continue working on its tuning, focusing, and (if possible) simplifying it for the cloud environment. Indeed, the so far approach for setting up the model (especially, at the domain specific level) was based on the information which is easy to get (e.g., resources, which can be retrieved by another MEDINA tool, i.e., Clouditor) and getting the knowledge from external resources about possible events (e.g., threat list). In the next phase, the model will be improved to ensure that the initially identified values are optimally selected, comprehensively describe cybersecurity events, and focused enough for measuring cyber risk. This activity will be performed in close collaboration with use case providers, exploiting their on-field domain-specific knowledge of the cloud environment, certification process and cyber security practices.

Also, we will continue improving our tool focussing more on the issues related with a tighter integration of the tool with other MEDINA components. Moreover, the settings of the RAOF will be aligned with the latest available version of EUCS released in 2022.

The new results in updating the model and the tool will be finalised by D2.8 at M30. Moreover, the continuous aspects of risk assessment with our model (and implemented by the supporting tool) will be reported in D4.5.

8 References

[1] MEDINA Consortium, "D2.6 Risk-based techniques and tools for Cloud Security Certification-v1," 2022.

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

[3] MEDINA Consortium, "D4.5 Methodology and tools for risk-based assessment and security control reconfiguration-v2," 2023.

[4] ENISA, "EUCS – Cloud Services Scheme," 2020. [Online]. Available: https://www.enisa.europa.eu/publications/eucs-cloud-service-scheme. [Accessed October 2022].

[5] PwC, "2022 Global Risk Survey Report," 2022. [Online]. Available: https://www.pwc.com/us/en/services/consulting/cybersecurity-risk-regulatory/library/global-risk-survey.html. [Accessed October 2022].

[6] ISO/IEC, "27001:2013 Information technology — Security techniques — Information security management systems — Requirements," 2013.

[7] NIST, "Cybersecurity Framework Version 1.1," 2018.

[8] D. J. Landoll, The security risk assessment handbook. A Complete Guide for performing Security Risk Assessment, Boca Raton: Taylor & Francis Group, 2011.

[9] ISO/IEC, "27005:2018 Information technology — Security techniques — Information security risk management," 2018.

[10] R. S. Ross, "Guide for Conducting Risk Assessments," NIST SP 800-30 Rev. 1, 2012.

[11] R. A. Caralli, J. F. Stevens, L. R. Young and W. R. Wilson, "Introducing OCTAVE Allegro: Improving the Information Security Risks," Software Engineering Institute, Carnegie Mellon University,, 2007.

[12] M. A. Amutio and J. Candau, "MAGERIT- Methodology for Information Systems Risk Analysis and Management. Book I - The Method, edition," Ministerio de Hacienda Y Administraciones Publicas, 2014.

[13] ISACA, "The RISK IT Framework," ISACA, 2009.

[14] E. Carlsson and M. Mattsson, "The MaRiQ model: A quantitative approach to risk management in cybersecurity," Uppsala University. Num: UPTEC STS 19017, Uppsala, 2019.

[15] P. Santini, G. Gottardi, M. Baldi and F. Chiaraluce, "A Data-Driven Approach to Cyber Risk Assessment," *Security and Communication Networks,* pp. 1-8, 2019.

(cc) BY-SA

[16] B. Karabacak and I. Sogukpinar, “ISRAM: information security risk analysis method,” *Computer & Security,* vol. 24, 2005.

[17] M. Krisper, J. Dobaj, G. Macher and C. Schmittner, “RISKEE: A Risk-Tree Based Method for Assessing Risk in Cyber Security,” in *EuroSPI 2019: Systems, Software and Services Process Improvement*, 2019.

[18] M. S. Lund, B. Solhaug and K. Stolen, Model-Driven Risk Analysis, Springer, 2011.

[19] F. Mathey, C. Bonhomme, J. Rocha, J. Lombardi and B. Joly, “Risk Assessment Optimisation with MONARC,” SMILE, 2018.

[20] G. Wangen, C. Hallstensen and E. Snek, “A framework for estimating information security risk assessment method completeness,” *International Journal of Information Security volume,* vol. 17, p. 681–699, 2018.

[21] D. W. Hybbard and R. Seiersen, How to measure anything in cybersecurity risk, 1st edn., New Jersey: John Wiley & Sons, 2016.

[22] L. A. T. Cox, “What's Wrong with Risk Matrices?,” *Risk analysis,* vol. 28, no. 2, pp. 497-512, 2008.

[23] J. Freund and J. Jones, Measuring and Managing Information Risk: A FAIR Approach, Oxford: Butterworth-Heinemann, 2014.

[24] E. Wheeler, Security risk management : building an information security risk management program from the ground up, Amsterdam: Syngress, 2011.

[25] N. A. Hashim, Z. Z. Abidin, N. A. Zakaria and R. Ahmad, “Risk Assessment Method for Insider Threats in Cyber Security: A Review,” *International Journal of Advanced Computer Science and Applications,* vol. 9, no. 11, pp. 126-130, 2018.

[26] V. Agrawal, “A Comparative Study on Information Security Risk,” *Journal of Computers,* vol. 12, no. 1, pp. 57-67, 2017.

[27] S. Schauer, “An adaptive supply chain cyber risk management methodology,” in *Hamburg International Conference of Logistics*, 2017.

[28] U. M. Aksu, H. M. Dilek, I. E. Tath, K. Bicakci, I. H. Dirik, U. M. Demirezen and T. Aykir, “A quantitatitve CVSS-based Cyber Security Risk Assessment Methodology For IT Systems,” in *2017 International Carnahan Conference on Security Technology (ICCST)*, Madrid, 2017.

[29] S. Musman and A. Turner, “A game theoretic approach to cyber security risk management,” *Journal of Defense Modeling and Simulation: Applications, Methodology, Technology,* vol. 15, no. 2, pp. 127-146, 2018.

[30] K. Jabbour and J. Poisson, “Cyber risk assessment in distributed information systems,” *The Cyber Defence Review,* vol. 1, no. 1, pp. 91-112, 2016.

(cc) BY-SA

[31] Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby and K. Stoddart, "A review of Cyber security risk assessment methods for SCADA systems," *Computers & Security,* vol. 56, pp. 1-27, 2016.

[32] G. Macher, E. Armengaud, E. Brenner and C. Kreiner, "Threat and Risk Assessment Methodologies in the Automotive Domain," *Procedia Computer Science,* vol. 83, pp. 1288-1294, 2016.

[33] S. Papastergiou, E.-M. Kalogeraki, N. Polemi and C. Douligeris, "Challenges and Issues in Risk Assessment in Modern Maritime Systems," in *Advances in Core Computer Science-Based Technologies*, Springer, 2020, p. 129–156.

[34] O. Akinrolabu, J. R. Nurse, A. Martin and S. New, "Cyber risk assessment in cloud provider environments: Current models and future needs," *Computers & Security,* vol. 87, pp. 1-18, 2019.

[35] F. Farahmand, S. B. Navathe, G. P. Sharp and P. H. Enslow, "Managing Vulnerabilities of Information Systems to Security Incidents," in *The 5th international conference on Electronic commerce*, 2003.

[36] B. Sheehan, F. Murphy, A. N. Kia and R. Kiely, "A quantitative bow-tie cyber risk classification and assessment framework," *Journal of Risk Research,* vol. 24, no. 12, pp. 1619-1638, 2021.

[37] B. Schneier, "Attack trees - modeling security threats," *Dr. Dobb's Journal,* vol. 24, no. 12, pp. 21-29, 1999.

[38] O. Akinrolabu, S. New and A. Martin, "CSCCRA: A Novel Quantitative Risk Assessment Model for Cloud Service Providers," in *European, Mediterranean, and Middle Eastern Conference on Information Systems*, Limassol, 2018.

[39] C.-A. Chih and Y.-L. Huang, "An Adjustable Risk Assessment Method for a Cloud System," in *2015 IEEE International Conference on Software Quality, Reliability and Security*, 2015.

[40] P. Saripalli and B. Walters, "QUIRC: A Quantitative Impact and Risk Assessment Framework for Cloud Security," in *The 3rd IEEE International Conference on Cloud Computing*, 2010.

[41] K. Djemame, D. Armstrong, J. Guitart and M. Macias, "A Risk Assessment Framework for Cloud Computing," *IEEE Transactions on cloud Computing,* vol. 4, no. 3, pp. 265-278, 2016.

[42] S. H. Albakri, B. Shanmugam, G. N. Samy, N. B. Idris and A. Ahmed, "Security risk assessment framework for cloud," *Security and communication network,* vol. 7, p. 2114–2124, 2014.

[43] H. A. Linstone and M. Turoff, The Delphi Method: Techniques and Applications, Addison-Wesley, 1975.

[44] ISO/IEC, "ISO/IEC 27000:2016 Information technology — Security techniques — Information security management systems — Overview and vocabulary," 2016.

(cc) BY-SA

[45] MEDINA Consortium, "D2.1 Continuously certifiable technical and organizational measures and catalogue of cloud security metrics-v1," 2021.

[46] M. Hristakeva and D. Shrestha, "Solving the 0–1 Knapsack Problem with Genetic Algorithms," in *Midwest instruction and computing symposium*, 2004.

[47] L. P. Rees, J. K. Deane, T. R. Rakes and W. H. Baker, "Decision support for Cybersecurity risk planning," *Decision Support Systems,* vol. 51, no. 3, pp. 493-505, June 2011.

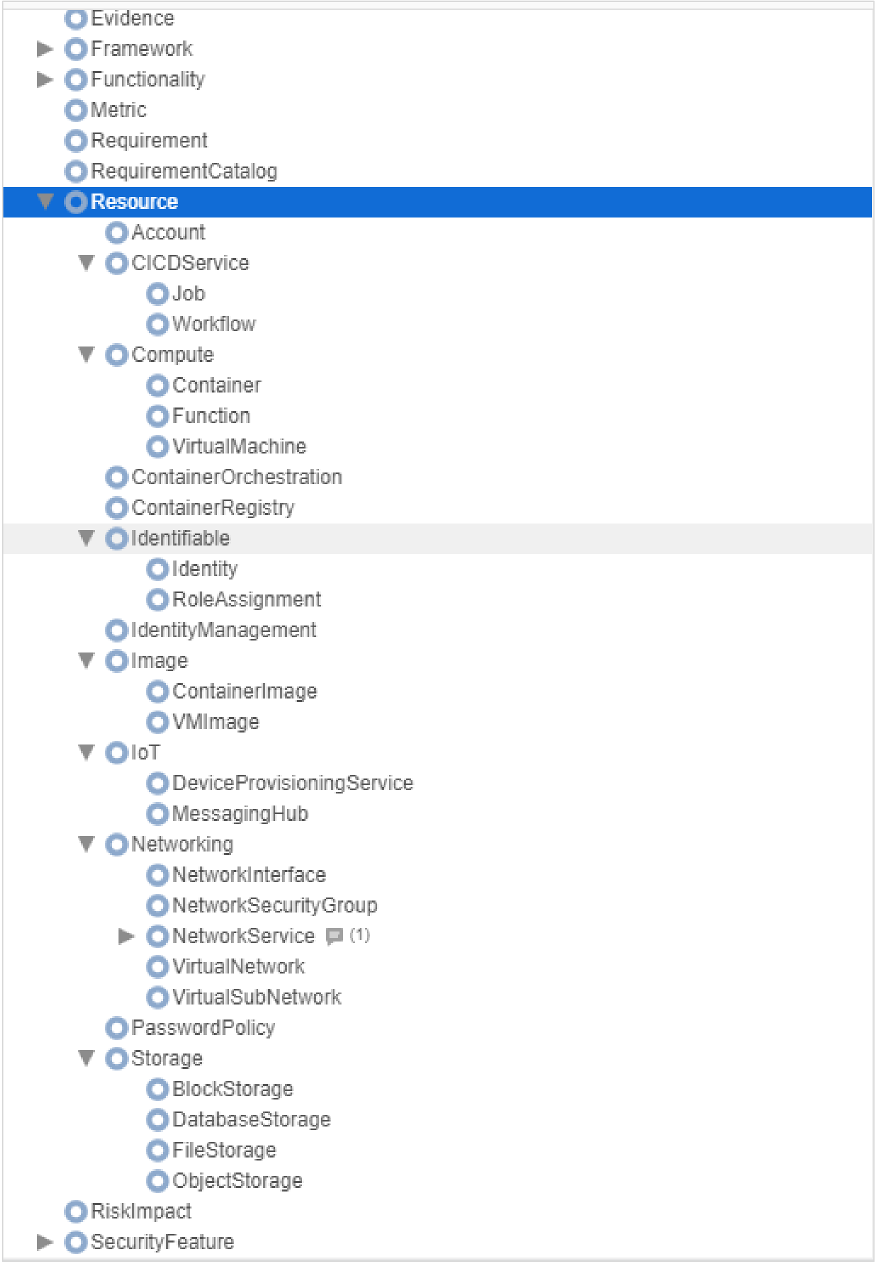
[48] G. Uuganbayar, A. Yautsiukhin, F. Martinelli y F. Massacci, «Optimisation of cyber insurance coverage with selection of cost effective security controls.,» *Computers & Security,* vol. 101, 2021.

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

[50] MEDINA Consortium, "D3.5 Tools and techniques for collecting evidence of technical and organisational measures – v2," 2022.

APPENDIX: Cloud resource ontology

The resource types used for the identification of asset types for our risk computation models are taken from the FhG cloud ontology, and they are also used by the Clouditor tool. Clouditor is able to detect a resource, categorise it and provide this information to other MEDINA components, including RAOF, during the continuous monitoring phase. Figure 12 shows the part of the ontology related to the Cloud Resources.

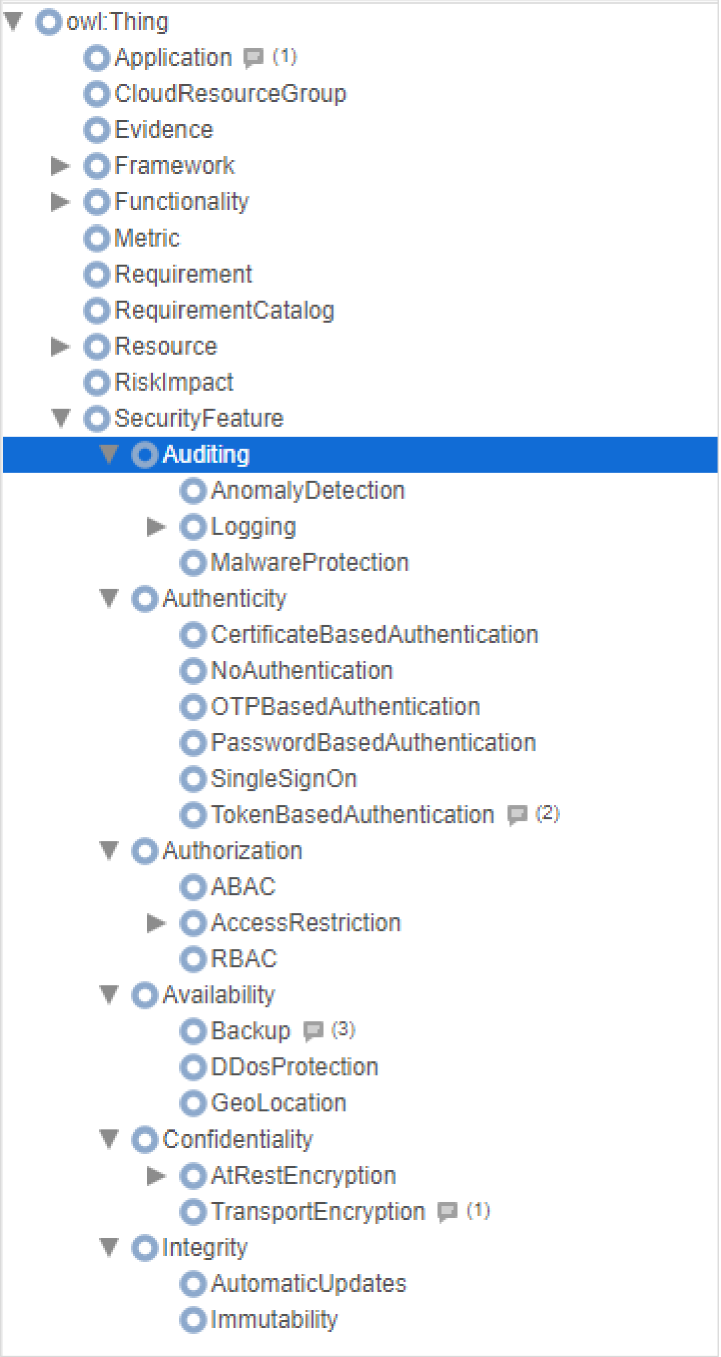


- Evidence
- ▶ Framework
- ▶ Functionality
- Metric
- Requirement
- RequirementCatalog
- ▼ Resource
  - Account
  - ▼ CICDService
    - Job
    - Workflow
  - ▼ Compute
    - Container
    - Function
    - VirtualMachine
  - ContainerOrchestration
  - ContainerRegistry
  - ▼ Identifiable
    - Identity
    - RoleAssignment
  - IdentityManagement
  - ▼ Image
    - ContainerImage
    - VMImage
  - ▼ IoT
    - DeviceProvisioningService
    - MessagingHub
  - ▼ Networking
    - NetworkInterface
    - NetworkSecurityGroup
    - ▶ NetworkService (1)
    - VirtualNetwork
    - VirtualSubNetwork
  - PasswordPolicy
  - ▼ Storage
    - BlockStorage
    - DatabaseStorage
    - FileStorage
    - ObjectStorage
- RiskImpact
- ▶ SecurityFeature

Figure 12. Cloud Resources of FhG ontology

It is worth noting that the ontology is wider than only resource types and includes other elements related to cloud security (see Figure 13), yet this information is not used by RAOF.

BY-SA



- owl:Thing
  - Application (1)
  - CloudResourceGroup
  - Evidence
  - Framework
  - Functionality
  - Metric
  - Requirement
  - RequirementCatalog
  - Resource
  - RiskImpact
  - SecurityFeature
    - Auditing
      - AnomalyDetection
      - Logging
      - MalwareProtection
    - Authenticity
      - CertificateBasedAuthentication
      - NoAuthentication
      - OTPBasedAuthentication
      - PasswordBasedAuthentication
      - SingleSignOn
      - TokenBasedAuthentication (2)
    - Authorization
      - ABAC
      - AccessRestriction
      - RBAC
    - Availability
      - Backup (3)
      - DDosProtection
      - GeoLocation
    - Confidentiality
      - AtRestEncryption
      - TransportEncryption (1)
    - Integrity
      - AutomaticUpdates
      - Immutability

Figure 13. FhG ontology. Security features

(cc) BY-SA